

$$00A0 \ 2203\exists \ 2200\forall \ 2286\subseteq \ 2713x \ 27FA\Longleftrightarrow \ 221A\sqrt{\phantom{x}} \ 221B\sqrt[3]{\phantom{x}} \ 2295\oplus \ 2297\otimes$$

---

Google □□□□□□□□

Mar 14, 2024



# CONTENTS

|      |                                   |     |
|------|-----------------------------------|-----|
| 1    | Google 开发指南——简介                   | 1   |
| 2    | C++ 开发指南 - 简介                     | 3   |
| 2.1  | 简介                                | 3   |
| 2.2  | 1. 简介                             | 6   |
| 2.3  | 2. 简介                             | 10  |
| 2.4  | 3. 简介                             | 18  |
| 2.5  | 4. 简介                             | 24  |
| 2.6  | 5. 简介 Google 开发指南                 | 27  |
| 2.7  | 6. 简介 C++ 开发指南                    | 28  |
| 2.8  | 7. 简介                             | 44  |
| 2.9  | 8. 简介                             | 48  |
| 2.10 | 9. 简介                             | 54  |
| 2.11 | 10. 简介                            | 68  |
| 2.12 | 11. 简介                            | 69  |
| 3    | Objective-C 开发指南 - 简介             | 71  |
| 3.1  | Google Objective-C Style Guide 简介 | 71  |
| 3.2  | 开发指南                              | 74  |
| 3.3  | 简介                                | 78  |
| 3.4  | 简介                                | 81  |
| 3.5  | Cocoa 开发指南 Objective-C 开发指南       | 83  |
| 3.6  | Cocoa 开发指南                        | 91  |
| 4    | Python 开发指南 - 简介                  | 93  |
| 4.1  | 简介                                | 93  |
| 4.2  | 简介                                | 93  |
| 4.3  | Python 开发指南                       | 94  |
| 4.4  | Python 开发指南                       | 107 |
| 4.5  | 开发指南                              | 130 |
| 5    | Shell 开发指南 - 简介                   | 131 |
| 5.1  | 简介                                | 131 |
| 5.2  | 简介                                | 131 |
| 5.3  | Shell 开发指南                        | 132 |
| 5.4  | 简介                                | 132 |
| 5.5  | 简介                                | 133 |
| 5.6  | 简介                                | 134 |
| 5.7  | 开发指南                              | 139 |
| 5.8  | 开发指南                              | 142 |

|      |                        |     |
|------|------------------------|-----|
| 5.9  | ????                   | 145 |
| 5.10 | ??                     | 146 |
| 6    | Javascript ???? - ???? | 147 |
| 6.1  | ??                     | 147 |
| 6.2  | Javascript????         | 147 |
| 6.3  | Javascript????         | 156 |
| 7    | TypeScript ????        | 191 |
| 7.1  | ??                     | 191 |
| 7.2  | ????                   | 192 |
| 7.3  | ????                   | 197 |
| 7.4  | ????                   | 214 |
| 7.5  | ????                   | 219 |
| 7.6  | ???                    | 227 |
| 8    | HTML/CSS ???? - ????   | 229 |
| 8.1  | ??                     | 229 |
| 8.2  | ??????                 | 229 |
| 8.3  | ??????                 | 229 |
| 8.4  | ????????               | 230 |
| 8.5  | HTML????               | 231 |
| 8.6  | HTML????               | 234 |
| 8.7  | css????                | 235 |
| 8.8  | CSS??????              | 238 |
| 8.9  | CSS????                | 241 |
| 8.10 | ??                     | 241 |
| 9    | Java ???? - ????       | 243 |
| 9.1  | 0. ??                  | 243 |
| 9.2  | 1. ??                  | 243 |
| 9.3  | 2. ?????               | 244 |
| 9.4  | 3. ?????               | 245 |
| 9.5  | 4. ??                  | 246 |
| 9.6  | 5. ??                  | 254 |
| 9.7  | 6. ?????               | 257 |
| 9.8  | 7. Javadoc             | 258 |

## GOOGLE 开发风格指南——中文版

- ReadTheDocs [开发风格指南](#)
- GitHub [zh-google-styleguide](#)
- [开发风格指南](#) release

Note: 注意

开发风格指南 Google 开发风格指南

开发风格指南 Google 开发风格指南 [Google Style Guide](#)

开发风格指南 Google 开发风格指南

“开发风格指南”“开发风格指南”camelCase“开发风格指南”“开发风格指南”“开发风格指南”

开发风格指南 Google 开发风格指南 Google 开发风格指南

开发风格指南 开发风格指南:

1. Google C++ 开发风格指南
2. Google Objective-C 开发风格指南
3. Google Python 开发风格指南
4. Google JavaScript 开发风格指南
5. Google Shell 开发风格指南
6. Google JSON 开发风格指南
7. Google TypeScript 开发风格指南

开发风格指南 reStructuredText 开发风格指南 Sphinx 开发风格指南 HTML / CHM / PDF 开发风格指南

- 开发风格指南 cplint —— 开发风格指南 google-c-style.el Google 开发风格指南 Emacs 开发风格指南
- 开发风格指南 JavaScript Style Guide 开发风格指南 XML Document Format Style Guide 开发风格指南 Yang.Y 开发风格指南



C++ `constexpr` - `constexpr`

Contents

- C++ `constexpr` - `constexpr`

2.1 `constexpr`

- `constexpr`
  - 2024/02/18
  - `constexpr`
    - Benjy Weinberger
    - Craig Silverstein
    - Gregory Eitzmann
    - Mark Mentovai
    - Tashana Landray
  - `constexpr`
    - YuleFox
    - Yang.Y
    - acgtyrant
    - lilinsanity
    - `constexpr`
  - `constexpr`
    - Google Style Guide
    - Google `constexpr` - `constexpr`





### 2.1.3 命名空间

命名空间

命名空间是 C++ 中用于组织代码的一种机制。它允许我们将具有相同名称的变量、函数或类放在不同的命名空间中，以避免冲突。命名空间通常用于组织大型项目中的代码，或者用于封装库中的代码。

命名空间的定义：

命名空间的定义

```
namespace std {
    // 定义 std 命名空间
    // ...
}

// 使用 std 命名空间
using namespace std;
```

命名空间的使用

```
std::cout << "Hello, World!" << endl;

// 使用 using 声明
using std::cout;
cout << "Hello, World!" << endl;
```

命名空间的别名

```
using namespace std;

// 使用别名
using namespace std;

// 使用别名
using namespace std;
```

命名空间 C++ 命名空间

```
namespace std {
    // 定义 std 命名空间
    // ...
}

// 使用 std 命名空间
using namespace std;
```

命名空间的别名

```
using namespace std;

// 使用别名
using namespace std;
```

命名空间 C++ 命名空间

```
namespace std {
    // 定义 std 命名空间
    // ...
}

// 使用 std 命名空间
using namespace std;
```

命名空间的别名

```
using namespace std;

// 使用别名
using namespace std;
```

命名空间的别名

```
using namespace std;

// 使用别名
using namespace std;
```

命名空间 C++ 命名空间。命名空间是 C++ 中用于组织代码的一种机制。它允许我们将具有相同名称的变量、函数或类放在不同的命名空间中，以避免冲突。命名空间通常用于组织大型项目中的代码，或者用于封装库中的代码。

## 2.1.4 C++

C++20, C++23 . C++ (2020) .  
.  
C++17 C++20 , .

## 2.2 1.

.cc .h . main() .cc .  
.  
.

### 2.2.1 1.1.

Tip: (self-contained, ), .h . (include) .inc , .

, . (header guards, 1.2. #define ) .

(inline function) (template), (instantiate) , (definition). (2 -inl.h ) ; , .cc , (explicit) , .cc (definition), .

, . . , . (pre-requisite). .inc . , .

### 2.2.2 1.2. #define

Tip: #define . : <>\_<>\_<>\_H\_ .

, . foo foo/src/bar/baz.h :

```
#ifndef FOO_BAR_BAZ_H_
#define FOO_BAR_BAZ_H_
...
#endif // FOO_BAR_BAZ_H_
```

## 2.2.3 1.3. `foo.h`

Tip: `foo.h` (symbol), `foo.h` (include) `foo.h` (declaration) `foo.h` (definition) `foo.h`. `foo.h`.

`foo.h`. `foo.h`, `foo.h` `#include` `foo.h`, `foo.h`. `foo.h`: `foo.cc` `foo.h` `foo.h`, `foo.h` `foo.h`, `foo.h` `foo.h` `foo.h`.

## 2.2.4 1.4. `foo.h`

Tip: `foo.h`. `foo.h` `foo.h`

`foo.h`

`foo.h` (forward declaration) `foo.h` (definition) `foo.h`.

```
// C++ foo.h
class B;
void FuncInB();
extern int variable_in_b;
ABSL_DECLARE_FLAG(flag_in_b);
```

`foo.h`

- `foo.h`, `foo.h` `#include` `foo.h`.
- `foo.h`, `foo.h` `#include`, `foo.h`.

`foo.h`

- `foo.h`, `foo.h`.
- `foo.h` `#include`, `foo.h`.
- `foo.h` (library) `foo.h`. `foo.h` API, `foo.h` (widening) `foo.h`, `foo.h`, `foo.h`, `foo.h`.
- `foo.h` `std::` `foo.h` (undefined behavior).
- `foo.h`, `foo.h` `#include`. `foo.h` `#include` `foo.h`, `foo.h`:

```
// b.h:
struct B {};
struct D : B {};

// good_user.cc:
#include "b.h"
void f(B*);
void f(void*);
void test(D* x) { f(x); } // f(B*)
```

`foo.h` `B` `D` `foo.h` `#include`, `test()` `foo.h` `f(void*)`.

- `foo.h` `#include` `foo.h`.
- `foo.h` (library) `foo.h`.

???

????????????????????.

## 2.2.5 1.5. ????

Tip: ?? 10 ?????????????? (inline).

??:

????????????????????, ?????????????????.

??:

????????????, ????????????? (object code) ????. ????????????? (accessors)???????? (mutators) ?????????????????????.

??:

?????????. ??????, ??????????????????. ??, ?????????????????????, ?????????????????????. ???????, ?????????????, ?????????????.

??:

???????????????? 10 ????. ??????????. ?????????????, ?????????????????????!

????????????: ????????? switch ????????????? (???????? switch ?????????).

????, ?????????????, ?????????????; ??, ?????????????????. ?????????????????. (YuleFox ?): ?????????????????, ?????????????????, ?????????????????). ????????????????? (class) ??????, ??????????????. ?????????????.

## 2.2.6 1.6. #include ???????

Tip: ?????????????: ??????, C ?????????, C++ ??????, ??????, ??????.

????????????????, ????? UNIX ????? (alias) . (????) ? .. (????). ??, ????????? google-awesome-project/src/base/logging.h:

```
#include "base/logging.h"
```

? dir/foo.cc ? dir/foo\_test.cc ????????? dir2/foo2.h ??????, ??????????:

1. dir2/foo2.h.
2. ??
3. C ?????? (????: ?????? .h ??????), ?? <unistd.h> ? <stdlib.h>.
4. ??
5. C++ ?????? (????), ?? <algorithm> ? <cstdint>.
6. ??
7. ????? .h ??.
8. ??

????????????????.

```
dir/foo.cc ? dir2/foo2.h ?????????? (? base/basictypes_unittest.cc ? base/
basictypes.h), ??????????????.
```

[illegible]

```
#include "foo/server/fooserver.h"

#include <sys/types.h>
#include <unistd.h>

#include <string>
#include <vector>

#include "base/basictypes.h"
#include "foo/server/bar.h"
#include "third_party/absl/flags/flag.h"
```

```

##### (system-specific) ##### (conditional include)#####.
#####.

```

```
#include "foo/public/fooserver.h"

#include "base/port.h"  // 22 LANG_CXX11.

#ifdef LANG_CXX11
#include <initializer_list>
#endif  // LANG_CXX11
```

1. `????????????????????;`
2. `??;`
3. `????????????????????;`
4. `-inl.h ???????? (?????:D);`
5. `???????????????????? (????????, ?????????);`
6. `???????? . ? . ?????, ?????????, ???, ?????????, ?????????, ?????? “????” (???? :D) ????, ??????, ?????????, ?????????, ?????????.`



```
// .h
namespace mynamespace {

// .....
// .....
class MyClass {
public:
    ...
    void Foo();
};

} // namespace mynamespace
```

```
// .cc
namespace mynamespace {

// .....
void MyClass::Foo() {
    ...
}

} // namespace mynamespace
```

cc, (flag) using .

```
#include "a.h"

DEFINE_FLAG(bool, someflag, false, "a");

namespace mynamespace {

using ::foo::Bar;

... // .....
} // namespace mynamespace
```

- proto, .proto package (specifier). Protocol Buffer .
- std. (forward declare) std (undefined behavior), .
- using namespace

```
// :.
using namespace foo;
```

- , (namespace alias). API.

```
// .cc,
namespace baz = ::foo::bar::baz;
```

```
// .h,
namespace librarian {
namespace impl { // , API.
```

(continues on next page)



(continued from previous page)

```

namespace sidetable = ::pipeline_diagnostics::sidetable;
} // namespace impl

inline void my_inline_function() {
    // (f)
    namespace baz = ::foo::bar::baz;
    ...
}
} // namespace librarian

```

- .
- “internal”, API.

```

// Absl
using ::absl::container_internal::ImplementationDetail;

```

- , .

```

:

```

```

namespace foo::bar {
    ...
} // namespace foo::bar

```

## 2.3.2 2.2.

Tip: .cc , (unnamed namespace) static, (internal linkage). .h .

:

```

. static .
, .

```

:

```

.cc .h .

```

```

. , :

```

```

namespace {
    ...
} // namespace

```

### 2.3.3 2.3. `constexpr`

Tip: `constexpr` (nonmember) `constexpr`; `constexpr` (completely global function). `constexpr` (static member) `constexpr` (class). `constexpr`.

Example:

```
constexpr int max = 100; // max is a constexpr variable.
```

Example:

```
constexpr int max = 100; // max is a constexpr variable.
```

Example:

```
constexpr int max = 100; // max is a constexpr variable.
constexpr int min = 0; // min is a constexpr variable.
constexpr int sum = max + min; // sum is a constexpr variable.
constexpr int diff = max - min; // diff is a constexpr variable.
constexpr int prod = max * min; // prod is a constexpr variable.
```

### 2.3.4 2.4. `constexpr`

Tip: `constexpr` (scope), `constexpr`.

Example C++ `constexpr`. `constexpr`, `constexpr`. `constexpr`, `constexpr`. `constexpr`, `constexpr`, `constexpr`:

```
int i;
i = f(); // i: constexpr.
```

```
int i = f(); // i: constexpr
```

```
int jobs = NumJobs();
// ...
f(jobs); // i: constexpr.
```

```
int jobs = NumJobs();
f(jobs); // i: constexpr (i) i.
```

```
vector<int> v;
v.push_back(1); // ...
v.push_back(2);
```

```
vector<int> v = {1, 2}; // i: constexpr v.
```

Example `if`/`while` `for` `constexpr`, `constexpr`. Example:

```
while (const char* p = strchr(str, '/')) str = p + 1;
```

Example, `constexpr`, `constexpr`, `constexpr`.

```
// 000000:
for (int i = 0; i < 1000000; ++i) {
    Foo f; // 00 1000000 000000000000.
    f.DoSomething(i);
}
```

[illegible]

```
Foo f; // 1 0000000000.
for (int i = 0; i < 1000000; ++i) {
    f.DoSomething(i);
}
```

### 2.3.5 2.5.

Tip: `static` (static storage duration) `constexpr` (trivially destructible). `constexpr` (destructor) `constexpr`, `constexpr` (base) `constexpr`. `constexpr`, `constexpr` (type) `constexpr` (virtual destructor), `constexpr`. `constexpr` (dynamic initialization) . `constexpr`, `constexpr`.

`constexpr`: `constexpr`, `constexpr` (constexpr), `constexpr`.

??:

`???` (object) `????` (linetime) `????` (storage duration). `??`,  
`????????`. `????????????????????????????` (`???`)`????????????????` static `???` (specifier) `????????????`.  
`????????????`, `????????????????????????????`; `????????????????????`. `????????????????????????????`  
(`????????` (join) `??????`).

XXXXXXXXXX (dynamic) 0, XXXXXXXXXXXXXXXX (non-trivial) 000. (00, XXXXXXXXXXXX, XXXXXXXX ID XXXXXXX.) XXXXXXXXXXXX (static) 000. XXXXXXXXXX: XXXXXXXXXXXX 00 XXXXXXX (XXXXXXXXXXXXXXXXXXXX), XXXXXXXXXXXXXXXX.

??:

```

XXXXXXXXXXXXXXXXXX:  XXXX  (named  constants)XXXXXX  (translation  unit)  XXXXXXXXXXXXXXXXXXXX
(flag)XXXXXXXXXXXXXXXXXXXX.

```

??:

00000000000000000000000000000000, 00000000, 00000000000. 00000000000000000000, 00000000 (00000000000000000000). 00000000000000000000000000000000, 00000000000000000000000000000000 (00000000). 00, 0000000000000000, 00000000000000000000000000000000.

??:

□□□□□□

[illegible]

```
const int kNum = 10; // ??

struct X { int n; };
```

(continues on next page)

```
const X kX[] = {{1}, {2}, {3}}; // 22

void foo() {
    static const char* const kMessages[] = {"hello", "world"}; // 22
}

// 22: constexpr ??????????????.
constexpr std::array<int, 3> kArray = {1, 2, 3};
```

```
// 00: 0000000.  
const std::string kFoo = "foo";  
  
// 000000000, 00 kBar 000 (00000000000000000000).  
const std::string& kBar = StrCat("a", "b", "c");  
  
void bar() {  
    // 00: 0000000.  
    static std::map<int, int> kData = {{1, 0}, {2, 0}, {3, 0}};  
}
```

□□□□□□□□

?????????, ?????????????????????, ?????????? (initializer) ????.

```
int n = 5;      // ??
int m = f();    // ? (?? f)
Foo x;          // ? (?? Foo::Foo)
Bar y = g();    // ? (?? g ? Bar::Bar)
```

????????, ?????????????????????.

`constexpr C++` `constexpr` (constant initialization), `constexpr` (constant expression), `constexpr`, `constexpr` `constexpr`:

```
struct Foo { constexpr Foo(int) {} };

int n = 5;    // 00, 5 000000.
Foo x(2);     // 00, 2 000000000000000000 constexpr.
Foo a[] = { Foo(1), Foo(2), Foo(3) }; // 00
```

```

XXXXXXXXXXXXXX.      XXX   constexpr    ?       constinit    XXXXXXXXXXXXXXXXXXXX.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX,XXXXXXXXXXXXXX.

```

□□□□, □□□□□□□□□□:

[illegible]

~~~~~. ~~, ~~~~~, ~~~~~, ~~~~~, ~~~~~.

```
int p = getpid(); // ~~~~~ p, ~~~.
```

~~~~~ (~~~~).

~~~~~

- ~~~~: ~~~~~ (named) ~~~~~, ~~~~ constexpr ~~~~ string\_view ~~~~~ (literal) ~~~~~, ~~~~~, ~~~~~, ~~~~ 140 ~~~~~.
- ~~~~~ (container): ~~~~~ (~~~~), ~~~~~, ~~~~~, ~~~~~, ~~~~~, ~~~~ int ~~~~~ (~~~ int ~~~~ int ~~~~) ~~~~~ (pair) ~~~~ (~~~~ int ~~~~ const char\* ~~~~), ~~~~~, ~~~~~, ~~~~~ (memory locality) ~~~~~; ~~~~~ `absl/algorithm/container.h` ~~~~~, ~~~~~, ~~~~~ (binary search), ~~~~~, ~~~~~.
- ~~~~~ (smart pointer, ~~~~ `std::unique_ptr` ~~~~ `std::shared_ptr`) ~~~~~, ~~~~~, ~~~~~, ~~~~~ (plain pointer) ~~~~~, ~~~~~ (~~~~).
- ~~~~~: ~~~~~, ~~~~~ constexpr ~~~~~.
- ~~~~~, ~~~~~, ~~~~~ (~~~ static const auto& impl = \*new T(args...);).

## 2.3.6 2.6. thread\_local

Tip: ~~~~~ (compile-time constant) ~~~~~ thread\_local ~~, ~~~~~ `ABSL_CONST_INIT` ~~~~~, ~~~~~ thread\_local, ~~~~~.

~~~:

~~~~~ thread\_local ~~~~~:

```
thread_local Foo foo = ...;
```

~~~~~. ~~~~~, ~~~~~, thread\_local ~~~~~ ~~~~~, ~~, ~~~~~, ~~~~~.

thread\_local ~~~~~, ~~~~ thread\_local ~~~~~, ~~~~~, ~~~~~ thread\_local ~~~~~, ~~~~~ thread\_local ~~, ~~~~~ (~~~~).

thread\_local ~~~~~: ~~~~~, thread\_local ~~~~~ (~~~ C++ ~~~~~). ~~~~ thread\_local ~~~~~ thread\_local ~~, ~~~~~ (use-after-free, ~~~~~) ~~~~.

~~~:

- ~~~~~ (race) (~~~~), ~~~~ thread\_local ~~~~~.
- ~~~~~, thread\_local ~~~~~.

~~~:

- ~~~~~ thread\_local ~~~~, ~~~~~.
- thread\_local ~~~~~, ~~~~~, ~~~~~.

- ??:

thread\_local, .

1. cc 0000000000000000, 00000, 000000 using 000000000;
2. 000000000000, 0000000000000000, 0000 public;
3. 00000000000000, 0000000000000, 0000000000;
4. 0000000000 (00000000) 0000 class 00 (0 STL 00), 0000000000 bug.
5. 000000, 000000000, 00000, 000000000, 0000/0000.

1. `using` `using-directive` `using` `using-declaration`
2. C static C++
3. locality
- 4.



~~~~~:

```
Func({42, 3.14}); // Error
```

~~~~~, ~~~~ explicit ~~~~.

~~

- ~~~~~, ~~~~~, ~~~~~.
- ~~~~~, ~~~~~ string\_view ~~~~~ std::string & const char\* ~~~~~.
- ~~~~~, ~~~~~.

~~

- ~~~~~, ~~~~~, ~~~~~.
- ~~~~~, ~~~~~, ~~~~~.
- ~~~~~.
- ~~~~~ explicit ~~~~, ~~~~~, ~~~~~ explicit ~~~~.
- ~~~~~, ~~~~~, ~~~~~, ~~~~~.
- ~~~~~, ~~~~~, ~~~~~.

~~

~~~~~, ~~~~~ explicit ~~~~~. ~~~~~, ~~~~~ explicit, ~~~~~.

~~~~~, ~~~~~, ~~~~~, ~~~~~, ~~~~~.

~~~~~ explicit. ~~~~ std::initializer\_list ~~~~~ explicit, ~~~~~ (~~ MyType m = {1, 2};).

### 2.4.3 3.3. ~~~~~

~~

~~~~~/~~~~/~~~~, ~~~~~.

~~

~~~~~.

~~~~~ (~~~~~), ~~~~~ std::unique\_ptr<int> ~~~~~ (~~~~~). int & string ~~~~~ (~~ int ~~, ~~~~~, ~~~ std::string ~~, ~~~~~).

~~~~~, ~~~~~ ~~~~~ ~~~~~, ~~~~~.

~~~~~, ~~~~~/~~~~, ~~, ~~~~~.

~~

~~~~~, ~~~ API ~~~, ~~~~~, ~~~~~, ~~~~~, ~~~~~, ~~~~~, ~~~~~, ~~~~~, ~~~~~ API ~~~~~, ~~~~~, ~~~~~.





protected 成员函数，在类成员函数前加上 protected 关键字，在类成员函数前加上 protected 关键字。

## 2.4.4 3.4. 结构体 VS. 类

结构体

结构体 struct, 类 class。

结构体

C++ 中 struct 和 class 的区别。结构体和类都是 C++ 中的数据类型，结构体和类都是 C++ 中的数据类型。

结构体 struct 和类 class 的区别。结构体和类都是 C++ 中的数据类型，结构体和类都是 C++ 中的数据类型。

结构体 struct 和类 class 的区别。结构体和类都是 C++ 中的数据类型，结构体和类都是 C++ 中的数据类型。

STL 中的 traits 和 struct 的区别。traits 是 C++ 中的一个概念，struct 是 C++ 中的一个数据类型。

结构体 struct 和类 class 的区别。结构体和类都是 C++ 中的数据类型，结构体和类都是 C++ 中的数据类型。

## 3.5. 对 vs. pair 和 tuple

对

对 struct 和 pair 和 tuple。

对

对 pair 和 tuple 的区别。pair 是 C++ 中的一个数据类型，tuple 是 C++ 中的一个数据类型。

pair 和 tuple 的区别。pair 是 C++ 中的一个数据类型，tuple 是 C++ 中的一个数据类型。

## 2.4.5 3.6. 接口

接口

接口 (YuleFox 的 GoF 的 <<Design Patterns>> 中的接口)。接口是 C++ 中的一个数据类型。

接口

接口是 C++ 中的一个数据类型。接口是 C++ 中的一个数据类型。

接口

接口是 C++ 中的一个数据类型。接口是 C++ 中的一个数据类型。

接口

接口是 C++ 中的一个数据类型。接口是 C++ 中的一个数据类型。

接口是 C++ 中的一个数据类型。接口是 C++ 中的一个数据类型。

接口

接口是 C++ 中的一个数据类型。接口是 C++ 中的一个数据类型。

接口是 C++ 中的一个数据类型。接口是 C++ 中的一个数据类型。

protected. The protected member functions of the derived class are not accessible to the base class.

override final (virtual) member functions. virtual. The final override final, virtual, virtual, virtual. The final override final, virtual, virtual, virtual.

virtual, virtual.

## 2.4.6 3.7. [C++](#)

[C++](#)

virtual. virtual.

[C++](#)

C++ operator, operator. operator operator operator " " operator, operator, operator operator bool().

[C++](#)

virtual, virtual. virtual (==, <, =, <), virtual, virtual.

virtual, virtual.

[C++](#)

- virtual, virtual, virtual, virtual, virtual Bug.
- virtual, virtual.
- virtual, virtual.
- virtual, virtual.
- virtual grep, C++.
- virtual, virtual. foo < bar, &foo < &bar.
- virtual. foo, virtual & virtual, virtual. foo &, || , virtual.
- virtual, virtual, virtual. virtual, virtual, virtual.
- virtual (UDLs) C++ virtual, "Hello World"sv std::string\_view("Hello World") virtual.
- virtual (UDLs) virtual, virtual virtual (virtual). virtual, virtual.

[C++](#)

virtual, virtual. virtual, virtual shell.

virtual. virtual, virtual. cc virtual. virtual, virtual. virtual, virtual, virtual.

virtual. virtual, virtual. virtual a + b virtual b + a virtual.

template<typename T, typename operator==> T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>. template<typename T> T t1, T t2 { return t1 == t2; } operator<=>, operator==>.

template<typename T> T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>. template<typename T> T t1, T t2 { return t1 == t2; } operator<=>, operator==>.

template<typename T> T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>.

template<typename T> T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>.

## 2.4.7 3.8. 2020

2020

template<typename T> T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>.

template<typename T> T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>.

## 2.4.8 3.9. 2020

2020

template<typename T> T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>.

2020

template<typename T> T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>.

template<typename T> T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>.

- 1. template<typename T> T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>.
- 2. (T, T struct) T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>.
- 3. T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>.
- 4. T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>.
- 5. T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>.
- 6. T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>.
- 7. T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>.
- 8. T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>.

template<typename T> T operator==(T t1, T t2) { return t1 == t2; } operator<=>, operator==>.



## 2.5.2 4.2. 字符串

在

字符串, 字符串。

在

字符串, 字符串。 字符串 40 个, 字符串。

字符串, 字符串, 字符串, 字符串 bug。 字符串, 字符串。

字符串, 字符串。 字符串: 字符串 / 字符串, 字符串, 字符串。

## 2.5.3 4.3. 字符串

在

字符串, 字符串, 字符串。 字符串。

在

字符串 const string& 字符串, 字符串 const char\* 字符串:

```
class MyClass {
public:
    void Analyze(const string &text);
    void Analyze(const char *text, size_t textlen);
};
```

在

字符串, 字符串。 字符串, 字符串。

在

字符串 (acgtyrant 字符串), 字符串 C++ 字符串, 字符串。 字符串, 字符串。

在

字符串, 字符串。 字符串, 字符串 AppendString() 字符串 AppendInt() 字符串, 字符串 Append()。 字符串 std::vector 字符串 字符串。

## 2.5.4 4.4. 字符串

在

字符串, 字符串。 字符串 字符串 字符串。 字符串, 字符串。

在

字符串, 字符串。 字符串, 字符串。 字符串, 字符串 “字符串” 字符串 “字符串”。

在

字符串, 字符串 字符串 字符串。

字符串, 字符串 字符串。

~~~~~  
~~~~~

~~~~~. ~~~~~.

??

~~~~~, ~~~~~, ~~~~~~. ~~~~~~. ~~~~~~. (??, ~~~~~ void f(int n = counter++); ~~~~~.)

~~~~~, ~~~~~~. ~~~~~~. ~~~~~~.

## 2.5.5 4.5. ~~~~~

??

~~~~~ (~~~~~) ~~~~~~.

??

C++ ~~~~~~. ~~~~~~. ??:

```
int foo(int x);
```

C++11 ~~~~~~. ~~~~~~ auto ~~~, ~~~~~~. ??:

```
auto foo(int x) -> int;
```

~~~~~. ~~~ int ~~~~~, ~~~~~~. ~~~~~~. ~~~~~~.

??

~~~~~ Lambda ~~~ ~~~~~~. ~~~~~, ~~~~~~ Lambda ~~~~~~. ~~~~~~.

~~~~~. ~~~~~~. ~~~~~, ~~~~~~. ??:

```
template <class T, class U> auto add(T t, U u) -> decltype(t + u);
```

~~~~~:

```
template <class T, class U> decltype(declval<T&>() + declval<U&>()) add(T t, U u);
```

??

~~~~~. ~~~ C ~ Java ~~~~~~.

~~~~~. ~~~~~~. ~~~~~~. ~~~~~~.

??

~~~~~, ~~~~~~. ~~~~~~. ~~~~~~ (?? Lambda ~~~) ~~~~~~. ~~~~~~. ~~~~~~. ~~~~~~.





```
std::shared_ptr<const Foo> p1, p2.
std::auto_ptr, std::unique_ptr
```

## 2.6.2 5.2. Cpplint

```
>
cpplint.py
>
cpplint.py, // NO-
LINT, // NOLINTNEXTLINE,
cpplint.py. cpplint.py.
```

## 2.6.3 acgyrant

1. ,
2. Rust Ownership C++
3. scoped\_ptr auto\_ptr, shared\_ptr unique\_ptr
4. , , ,
5. Arch Linux , AUR cpplint

## 2.7 6. C++

### 2.7.1 6.1.

Tip: .

:

```
void f(string& s);
```

```
&& " ".
```

:

```
vector<string>,
auto v2(std::move(v1))
std::unique_ptr, std::move
```

:

```
( C++11 ),
```

??:

```

std::forward<T>(std::move(
    std::move(

```

## 2.7.2 6.2. ???? ?

Tip: call site

??:

```

const string& text, const char* text:

```

```

class MyClass {
public:
    void Analyze(const string &text);
    void Analyze(const char *text, size_t textlen);
};

```

??:

```

, , , .

```

??:

```

acgtyrant C++

```

??:

```

AppendString() AppendInt()
Append().

```

## 2.7.3 6.3. ???? ?

Tip:

??:

```


```

??:

```

function signature
site acgtyrant

```

??:

```

acgtyrant
.cc

```

```
// AlphaNum
string StrCat(const AlphaNum &a,
              const AlphaNum &b = gEmptyAlphaNum,
              const AlphaNum &c = gEmptyAlphaNum,
              const AlphaNum &d = gEmptyAlphaNum);
```

## 2.7.4 6.4. `alloca()`

Tip: `alloca()`.

??:

`alloca()`. `alloca()` `alloca()`.

??:

`alloca()` `C++`. `alloca()`, `alloca()`, `alloca()` bugs: “`alloca()`, `alloca()`”.

??:

`allocator` `std::vector` `std::unique_ptr<T[]>`.

## 2.7.5 6.5. `??`

Tip: `??`.

`FooBuilder` `Foo` `Foo`, `FooBuilder` `Foo` `Foo`, `FooBuilder`. `Foo`, `FooBuilder`.

`Foo` (`Foo`) `Foo`. `Foo`, `Foo` `public`, `Foo`, `Foo`. `Foo`, `Foo`.

## 2.7.6 6.6. `??`

Tip: `C++`.

??:

- `failures` `acgtyrant` `error code`, `int`
- `C++` `Python`, `Java` `C++`
- `C++`
- `acgtyrant` `factory function`, `C++` `Init()` `Init()`, `“??”`
- `??`

??:

- ??:

Windows 10, 11.

??:

??:

RTTI. ??:

??:





2.7.10 6.10. `constexpr`


---

Tip: `constexpr` (`++i`) `foo`, `bar`.

---

??:

```
constexpr (++i) foo (--i) bar, bar (foo).
```

??:

```
constexpr, foo (++i) bar (i++) foo. bar (foo) bar i bar. i i
bar, bar. bar, bar?
```

??:

```
in C, bar, bar, bar for foo. bar, bar, i
(i) bar (++). 
```

??:

```
bar (foo), bar. bar, bar (foo).
```

2.7.11 6.11. `const` ??

---

Tip: `constexpr` `const`. `constexpr` C++11 `constexpr` ??

---

??:

```
constexpr const constexpr (const int foo). constexpr const
constexpr (class Foo { int Bar(char c) const; };
```

??:

```
constexpr. constexpr, foo, constexpr. constexpr,
constexpr. constexpr, constexpr.
```

??:

```
const foo: constexpr const foo, constexpr const foo (constexpr const_cast
foo), bar.
```

??:

```
const foo, bar, bar; bar. foo, bar const:


- constexpr, bar const.
- constexpr const. constexpr const. constexpr, bar const foo, bar const
const constexpr const.
- constexpr, bar const.


foo, constexpr const. const int * const * const x; bar, bar x.
constexpr: constexpr const int** x.
```

```
foo mutable bar, bar, bar.
```

const foo:

```
int const *foo, const int* foo, : const
const ., “” . const
, (const) (int) .
```

, const . ! (Yang.Y : const , , , )

## 2.7.12 6.12. constexpr

Tip: C++11 的 constexpr

:

```
constexpr constexpr,
constexpr
```

:

```
constexpr
```

:

```
constexprconstexprconstexpr
```

:

```
constexpr C++ constexpr
constexprconstexprconstexprconstexprconstexprconstexpr
constexpr
```

## 2.7.13 6.13.

Tip: C++ 的, int. , <stdint.h> , int16\_t. 2^31 (2GiB), 64 int64\_t. int

:

C++ short 16 , int 32 , long 32 , long long 64 .

:

:

C++ .

:

```
<stdint.h> int16_t, uint32_t, int64_t , short, un-
signed long long . C , int. , size_t ptrdiff_t.
```

```
int, . int. int 32 , 32 . 64 , int64_t uint64_t.
```

```
, int64_t.
```



```

uint32_t 00000000, 00000000000000000000, 0000000000000000. 000000000000000000,
00000000. 00, 0000000000000000.

```

size

```

XXXXXXXXXXXXXXXXXXXXagctyrant XXinteger promotions, XX int X unsigned int XXXXXXXXXXXXXX un-
signed int XXXXXXXXXXXXXXXXXXXXXXX

```

□□□□□□□:

222, 2222222222, 222222222222222. 222222222222222. 22, 2 C 222, 222222222222 bug 222.  
22222222:

```
for (unsigned int i = foo.Length()-1; i >= 0; --i) ...
```

XXXXXXXXXX! 00 gcc 0000 bug 000, XXXXXXXXXXXX. 000 bug XXXXXXXXXXXXXXXXXXXX. 000 C  
XXXXXXXXXXXXXXXXXXXXXXXXXXXX.

??, ??????????, ?????????!

2.7.14 6.14. 64

Tip: 64 32. , , :

- `00000000, printf() 000000 32 00 64 0000000000000000. C99 00000000000000000000. 00000, MSVC 7.1 00000000, 000000000000, 000000000000000000000000 (000 inttypes.h 000000):`

```
// printf macros for size_t, in the style of inttypes.h
#ifdef __LP64
#define __PRIS_PREFIX "z"
#else
#define __PRIS_PREFIX
#endif

// Use these macros after a % in a printf format string
// to get correct 32/64 bit behavior, like this:
// size_t size = records.size();
// printf("%"PRIuS"\n", size);
#define PRIdS __PRIS_PREFIX "d"
#define PRIxS __PRIS_PREFIX "x"
#define PRIuS __PRIS_PREFIX "u"
#define PRIxS __PRIS_PREFIX "x"
#define PRIoS __PRIS_PREFIX "o"
```

| ??               | ????            | ??                   | ??         |
|------------------|-----------------|----------------------|------------|
| void * (???????) | %lx             | %p                   |            |
| int64_t          | %qd, %lld       | %"PRId64"            |            |
| uint64_t         | %qu, %llu, %llx | %"PRIu64", %"PRIx64" |            |
| size_t           | %u              | %"PRIuS", %"PRIxS"   | C99 ?? %zu |
| ptrdiff_t        | %d              | %"PRIdS"             | C99 ?? %zd |

```
00 PRI* 00000000000000. 0000000000000000, 0000000000000000. 00 PRI*
000000 % 00000000. 00.printf("x = %30PRIuS"\n", x) 0 32 0 Linux 000000
```

```
printf("x = %30" "u" "\n", x), printf("x = %30u\n", x) (Yang.Y  
?: 64 MSVC 6.0 64, VC 6 64)
```

- sizeof(void \*) != sizeof(int). intptr\_t.
- 64, (Yang.Y : 64 - 64). 64 64, int64\_t/uint64\_t 64/64, 64 8 64. 32 64, 64 64. gcc \_\_attribute\_\_((packed)). MSVC #pragma pack() \_\_declspec(align()) (YuleFox, 64).
- 64 LL ULL, :

```
int64_t my_value = 0x123456789LL;
uint64_t my_mask = 3ULL << 48;
```

- 32 64, #ifdef \_LP64 32/64. (64, 64, 64)

## 2.7.15 6.15. 6.15

Tip: 6.15, 6.15, 6.15.

6.15, 6.15, 6.15.

6.15, C++ 6.15, C 6.15. 6.15, 6.15. 6.15 const 6.15. 6.15 “6.15” 6.15. 6.15... 6.15, 6.15, 6.15 (#define 6.15).

6.15, 6.15 (6.15) 6.15 (6.15 # 6.15, 6.15 ## 6.15). 6.15, 6.15.

6.15; 6.15, 6.15:

- 6.15 .h 6.15.
- 6.15 #define, 6.15 #undef.
- 6.15 #undef 6.15
- 6.15 C++ 6.15, 6.15.
- 6.15 ## 6.15

## 2.7.16 6.16. 0, nullptr 6.16 NULL

Tip: 6.16 nullptr 6.16 '\0' (6.16 0 6.16) 6.16

6.16 (6.16) 6.16 nullptr 6.16

6.16 '\0' 6.16

## 2.7.17 6.17. sizeof

Tip: `sizeof(varname)` `sizeof(type)`.

```
sizeof(varname) // sizeof(type)
// sizeof(type)
```

```
Struct data;
Struct data; memset(&data, 0, sizeof(data));
```

Warning:

```
memset(&data, 0, sizeof(Struct));
```

```
if (raw_size < sizeof(int)) {
    LOG(ERROR) << "compressed record not big enough for count: " << raw_size;
    return false;
}
```

## 2.7.18 6.18. auto

Tip: `auto` `auto`

`auto`

C++11 `auto`, `auto`

```
vector<string> v;
...
auto s1 = v[0]; // v[0]
const auto& s2 = v[0]; // s2 v[0]
```

`auto`

C++ `auto`

```
sparse_hash_map<string, int>::iterator iter = m.find(val);
```

`auto`

```
auto iter = m.find(val);
```

`auto`

`auto`

```
diagnostics::ErrorStatus* status = new diagnostics::ErrorStatus("xyz");
```

`auto`, `auto`

`auto`



```
std::initializer_list<T>
```

```
class MyType {
public:
    // std::initializer_list init
    //
    MyType(std::initializer_list<int> init_list) {
        for (int i : init_list) append(i);
    }
    MyType& operator=(std::initializer_list<int> init_list) {
        clear();
        for (int i : init_list) append(i);
    }
};
MyType m{2, 3, 5, 7};
```

```
std::initializer_list<T>
```

```
double d{1.23};
// MyOtherType std::initializer_list
//
class MyOtherType {
public:
    explicit MyOtherType(string);
    MyOtherType(int, string);
};
MyOtherType m = {1, "b"};
// explicit`= {}`
MyOtherType m{"b"};
```

```
auto
```

#### Warning:

```
auto d = {1.23}; // d std::initializer_list<double>
```

```
auto d = double{1.23}; // -- d double, std::initializer_list.
```

9.7.

## 2.7.20 6.20. Lambda

Tip: lambda lambda

Lambda

```
std::sort(v.begin(), v.end(), [](int x, int y) {
    return Weight(x) < Weight(y);
});
```

C++11 Lambdas, polymorphic wrapper std::function.



## 2.7.22 6.22. Boost

---

Tip: Boost.

---

:

Boost, , C++.

:

Boost, , C++ , , ,

:

Boost , “”.

:

Boost .:

- **Call Traits**: `boost/call_traits.hpp`
- **Compressed Pair**: `boost/compressed_pair.hpp`
- **<The Boost Graph Library (BGL)**: `boost/graph`, except `serialization` (`adj_list_serialize.hpp`) and `parallel/distributed` algorithms and data structures (`boost/graph/parallel/*` and `boost/graph/distributed/*`)
- **Property Map**: `boost/property_map.hpp`
- The part of **Iterator** that deals with defining iterators: `boost/iterator/iterator_adaptor.hpp`, `boost/iterator/iterator_facade.hpp`, and `boost/function_output_iterator.hpp`
- The part of **Polygon** that deals with Voronoi diagram construction and doesn't depend on the rest of **Polygon**: `boost/polygon/voronoi_builder.hpp`, `boost/polygon/voronoi_diagram.hpp`, and `boost/polygon/voronoi_geometry_type.hpp`
- **Bimap**: `boost/bimap`
- **Statistical Distributions and Functions**: `boost/math/distributions`
- **Multi-index**: `boost/multi_index`
- **Heap**: `boost/heap`
- The flat containers from **Container**: `boost/container/flat_map`, and `boost/container/flat_set`

Boost ,.

C++ 11

- **Pointer Container**: `boost/ptr_container`, `std::unique_ptr`
- **Array**: `boost/array.hpp`, `std::array`

## 2.7.23 6.23. C++11

Tip: C++11 C++0x C++11

C++11

C++ C++ C++

C++11 1300 vs 800

6.22. Boost C++11

C++11 C++11

- auto foo() -> int foo().
- <ratio>,
- <cfenv> <fenv.h>
- lambda

## 2.7.24 acgtyrant

1. void a() void a(int b = 0), int
- 2.
- 3.
4. friend friend
- 5.
- 6.
7. C++
8. const
- 9.
10. auto
11. Should the trailing return type syntax style become the default for new C++11 programs? auto



## 2.8 7. 命名空间

命名空间（namespace）是 C++ 中用于组织代码的一种机制。它允许你将相关的函数、变量、类等组织在一起，并避免命名冲突。命名空间的名称通常由点（.）分隔，表示层次结构。例如，std 命名空间包含 C++ 标准库中的许多函数和变量。

命名空间的名称通常由点（.）分隔，表示层次结构。例如，std 命名空间包含 C++ 标准库中的许多函数和变量。

### 2.8.1 7.1. 命名空间的声明和定义

命名空间的声明和定义通常如下所示：

命名空间的名称通常由点（.）分隔，表示层次结构。例如，std 命名空间包含 C++ 标准库中的许多函数和变量。

命名空间的名称通常由点（.）分隔，表示层次结构。例如，std 命名空间包含 C++ 标准库中的许多函数和变量。

命名空间的名称通常由点（.）分隔，表示层次结构。例如，std 命名空间包含 C++ 标准库中的许多函数和变量。

```
int price_count_reader;    // 价格计数器
int num_errors;            // "num" 错误次数
int num_dns_connections;   // 命名空间 "DNS" 中的连接数
```

```
int n;                     // 命名空间。
int nerr;                  // 命名空间。
int n_comp_conns;         // 命名空间。
int wgc_connections;      // 命名空间。
int pc_reader;            // "pc" 命名空间。
int cstmr_id;             // 命名空间。
```

命名空间的名称通常由点（.）分隔，表示层次结构。例如，std 命名空间包含 C++ 标准库中的许多函数和变量。

命名空间的名称通常由点（.）分隔，表示层次结构。例如，std 命名空间包含 C++ 标准库中的许多函数和变量。

### 2.8.2 7.2. 命名空间的别名

命名空间的名称通常由点（.）分隔，表示层次结构。例如，std 命名空间包含 C++ 标准库中的许多函数和变量。

命名空间的名称通常由点（.）分隔，表示层次结构。例如，std 命名空间包含 C++ 标准库中的许多函数和变量。

命名空间的名称通常由点（.）分隔，表示层次结构。例如，std 命名空间包含 C++ 标准库中的许多函数和变量。

命名空间的名称通常由点（.）分隔，表示层次结构。例如，std 命名空间包含 C++ 标准库中的许多函数和变量。

- my\_useful\_class.cc
- my-useful-class.cc
- myusefulclass.cc
- myusefulclass\_test.cc // \_unittest 和 \_regtest 测试。

C++ 命名空间 .cc 文件，命名空间 .h 文件。命名空间 .inc 文件，命名空间 .h 文件。

命名空间 /usr/include 命名空间 (Yang.Y 命名空间: 命名空间), 命名空间 db.h。

命名空间 http\_server\_logs.h 命名空间 logs.h 命名空间。命名空间, 命名空间 foo\_bar.h 命名空间 foo\_bar.cc, 命名空间 FooBar。

命名空间 .h 文件。命名空间, 命名空间 .h 文件。

### 2.8.3 7.3. `MyExcitingClass`

`MyExcitingClass`

`MyExcitingClass`, `MyExcitingEnum`.

`MyExcitingEnum`

`MyExcitingEnum` — `MyExcitingEnum` (typedef), `MyExcitingEnum` — `MyExcitingEnum`, `MyExcitingEnum`, `MyExcitingEnum`, `MyExcitingEnum`. `MyExcitingEnum`:

```
// MyExcitingClass
class UrlTable { ...
class UrlTableTester { ...
struct UrlTableProperties { ...

// MyExcitingEnum
typedef hash_map<UrlTableProperties *, string> PropertiesMap;

// using MyExcitingEnum
using PropertiesMap = hash_map<UrlTableProperties *, string>;

// MyExcitingEnum
enum UrlTableErrors { ...
```

### 2.8.4 7.4. `MyExcitingClass`

`MyExcitingClass`

`MyExcitingClass` (`MyExcitingClass`) `MyExcitingClass`, `MyExcitingClass`. `MyExcitingClass`, `MyExcitingClass`, `MyExcitingClass`: `a_local_variable`, `a_struct_data_member`, `a_class_data_member`.

`MyExcitingClass`

`MyExcitingClass`

`MyExcitingClass`:

```
string table_name; // MyExcitingClass - MyExcitingClass.
string tablename; // MyExcitingClass - MyExcitingClass.

string tableName; // MyExcitingClass - MyExcitingClass
```

`MyExcitingClass`

`MyExcitingClass`, `MyExcitingClass`, `MyExcitingClass`.

```
class TableInfo {
...
private:
string table_name_; // MyExcitingClass - MyExcitingClass.
string tablename_; // MyExcitingClass.
static Pool<TableInfo>* pool_; // MyExcitingClass.
};
```

□□□□□

```
struct UrlTableProperties {
    string name;
    int num_entries;
    static Pool<UrlTableProperties>* pool;
};
```

????? , ?? ??? vs. ?.

2.8.5 7.5.

??

??? constexpr ? const ???, ??????????????????, ??? “k” ?, ??. ??:

```
const int kDaysInAWeek = 7;
```

??

XXXXXXXXXXXX (XXXXXXXXXX, 00 XXXX) XXXXXXXX. XXXXXXXXXX, XXXXXX, XXXXXXXX.  
XXXXXXXXXX, XXXXXXXXXX.

2.8.6 7.6.

??

```

#####, #####: MyExcitingFunction(), MyExcitingMethod(),
my_exciting_member_variable(), set_my_exciting_member_variable().

```

??

0000, 00000000000000 (0 “000000” 0 “0000000”), 000000. 000000000000, 0000000000000000000000 (00, 00 StartRpc () 00 StartRPC ()).

```
AddTableEntry()  
DeleteUrl()  
OpenFileOrDie()
```

(XXXXXXXXXXXXXXXXXXXXXXXXXXXX, XXXXXX API XXXXXXXXXX, XXXXXXXXXXXXXXXXXXXX, XXXXX, XXX.)

```

XXXXXXXXXXXXXXXXXX. XXXXXXXXXXXXXXXXXXXXXXXX, XXXXXXXX. int  count() void  set_count(int
count).

```



2.8.10 7.10.

??

???????? C/C++ ????, ?????????.

```
bigopen(): 100 open() 100
```

```
uint: typedef
```

```
bigpos: struct { class, pos }
```

sparse\_hash\_map: STL [?][?][?]; [??] STL [?][?][?][?]

```
LONGLONG_MAX: ??, ?? INT_MAX
```

2.8.11 `acgtyrant`

1. `Google` `QueryResult`, `query_result`, `TextQuery::TextQuery(std::string word)` : `word_(word) {}`, `word_`

2.9 8.

XXXXXXXXXX, XXXXXXXXXXXXX. XXXXXXXXXXXXXXXXXXXX. XXXXXX: XXXXXX, XXXXXXXXXXXXXXXX.  
XXXXXXXXXX, XXXXXXXXXXXXXXXXXXXX.

????????????, ?????????????????. ?????, ?????????!!

2.9.1 8.1.

??

?? // ? / \* \* /, ????.

??

// ? / \* \* / ????; ? // ? ??, ?????????????????.

2.9.2 8.2.

??

????????????????.

[illegible]

??

2.9.3 8.3. 2.9.3

2.9.3 8.3. 2.9.3, Apache 2.0, BSD, LGPL, GPL)

2.9.3 8.3. 2.9.3, 2.9.3.

2.9.3

2.9.3 .h 2.9.3, 2.9.3, 2.9.3. 2.9.3, 2.9.3, 2.9.3.

2.9.3 .h 2.9.3 .cc 2.9.3, 2.9.3.

## 2.9.3 8.3. 2.9.3

2.9.3

2.9.3 8.3. 2.9.3, 2.9.3.

```
// Iterates over the contents of a GargantuanTable.
// Example:
//     GargantuanTableIterator* iter = table->NewIterator();
//     for (iter->Seek("foo"); !iter->done(); iter->Next()) {
//         process(iter->key(), iter->value());
//     }
//     delete iter;
class GargantuanTableIterator {
    ...
};
```

2.9.3

2.9.3 8.3. 2.9.3, 2.9.3. 2.9.3, 2.9.3. 2.9.3, 2.9.3.

2.9.3 8.3. 2.9.3, 2.9.3.

2.9.3 8.3. 2.9.3 .h 2.9.3 .cc 2.9.3), 2.9.3, 2.9.3, 2.9.3.

## 2.9.4 8.4. 2.9.4

2.9.4

2.9.4 8.4. 2.9.4; 2.9.4.

2.9.4

## File

File, File. File(File, File). File  
 (“Opens the file”) File (“Open the file”); File, File. File, File. File.

File:

- File.
- File: File, File.
- File.
- File.
- File.
- File, File?

File:

```
// Returns an iterator for this table. It is the client's
// responsibility to delete the iterator when it is done with it,
// and it must not use the iterator once the GargantuanTable object
// on which the iterator was created has been deleted.
//
// The iterator is initially positioned at the beginning of the table.
//
// This method is equivalent to:
//   Iterator* iter = table->NewIterator();
//   iter->Seek("");
//   return iter;
// If you are going to immediately seek to another place in the
// returned iterator, it will be faster to use NewIterator()
// and avoid the extra seek.
Iterator* GetIterator() const;
```

File, File. File “File false”, File:

```
// Returns true if the table cannot hold any more entries.
bool IsTableFull();
```

File, File, File. File, File, File.

File/File, File/File, File “File” File. File (File, File) File. File, File. File.

## File

File, File. File, File, File. File, File.

File.h File. File, File.

## 2.9.5 8.5. 2222

22

2222222222222222. 22222, 2222222222.

22

222222

22222222 (222222222222) 2222222222. 2222222222(22222, 2222222222, 22222)22222222222222, 22222222. 22, 2222222222222222222222, 222222222222.

222, 2222222222 NULL 2 -1 22222, 222222. 22:

```
private:
// Used to bounds-check table accesses. -1 means
// that we don't yet know how many entries the table has.
int num_total_entries_;
```

22222

22222222, 22222222222222222222, 222222222222. 22:

```
// The total number of tests cases that we run through in this regression test.
const int kNumTestCases = 6;
```

## 2.9.6 8.6. 2222

22

222222222, 222, 222, 2222222222.

22

2222222

2222222222222222. 22:

```
// Divide result by two, taking into account that x
// contains the carry from the add.
for (int i = 0; i < result->size(); i++) {
    x = (x << 8) + (*result)[i];
    (*result)[i] = x >> 1;
    x &= 1;
}
```



???

?????????????????. ??????????. ??:

```
// If we have enough memory, mmap the data portion too.
mmap_budget = max<int64>(0, mmap_budget - index_->length());
if (mmap_budget >= data_size_ && !MmapData(mmap_chunk_bytes, mlock))
    return; // Error already logged.
```

??, ?????????????????????, ?????????????????????.

????????????????, ?????????????????:

```
DoSomething(); // Comment here so the comments line up.
DoSomethingElseThatIsLonger(); // Two spaces between the code and the comment.
{ // One space before comment when opening a new scope is allowed,
  // thus the comment lines up with the following comments and code.
  DoSomethingElse(); // Two spaces before line comments normally.
}
std::vector<string> list{
    // Comments in braced lists describe the next element...
    "First item",
    // .. and should be aligned appropriately.
    "Second item"};
DoSomething(); /* For trailing block comments, one space is fine. */
```

???????

????????????, ?????????????:

- ?????????, ?????????????????, ?????????, ?????????????????, ?????????.
- ?????????, ??? bool ????????? enum ??, ?????????.
- ?????????, ?????????????????, ?????????, ?????????, ?????????????????, ?????????????, ?????????????, ?????????????, ?????????, ?????????, ?????????.
- ?????????.
- ?????, ?????????.

????????:

```
// What are these arguments?
const DecimalNumber product = CalculateProduct(values, 7, false, nullptr);
```

?

```
ProductOptions options;
options.set_precision_decimals(7);
options.set_use_cache(ProductOptions::kDontUseCache);
const DecimalNumber product =
    CalculateProduct(values, options, /*completion_callback=*/nullptr);
```

????????.

## 2.9.7 8.7. `std::find`

2.9.7 8.7. `std::find`, 2.9.7 8.7. `std::find`, 2.9.7 8.7. `std::find` C++ 2.9.7 8.7. `std::find`, 2.9.7 8.7. `std::find`:

2.9.7 8.7. `std::find`, 2.9.7 8.7. `std::find`.

2.9.7 8.7. `std::find`:

```
// Find the element in the vector. <-- 2: 2.9.7 8.7. std::find!
auto iter = std::find(v.begin(), v.end(), element);
if (iter != v.end()) {
    Process(element);
}
```

2.9.7 8.7. `std::find`:

```
// Process "element" unless it was already processed.
auto iter = std::find(v.begin(), v.end(), element);
if (iter != v.end()) {
    Process(element);
}
```

2.9.7 8.7. `std::find`, 2.9.7 8.7. `std::find`:

```
if (!IsAlreadyProcessed(element)) {
    Process(element);
}
```

## 2.9.7 8.7. `std::find`, 2.9.7 8.7. `std::find`

2.9.7 8.7. `std::find`

2.9.7 8.7. `std::find`, 2.9.7 8.7. `std::find`.

2.9.7 8.7. `std::find`

2.9.7 8.7. `std::find`, 2.9.7 8.7. `std::find`. 2.9.7 8.7. `std::find`, 2.9.7 8.7. `std::find`, 2.9.7 8.7. `std::find`.

2.9.7 8.7. `std::find`, 2.9.7 8.7. `std::find`. 2.9.7 8.7. `std::find`, 2.9.7 8.7. `std::find`.

## 2.9.8 8.8. `std::find`

2.9.8 8.8. `std::find`

2.9.8 8.8. `std::find`, 2.9.8 8.8. `std::find` TODO 2.9.8 8.8. `std::find`.

TODO 2.9.8 8.8. `std::find` TODO, 2.9.8 8.8. `std::find`, 2.9.8 8.8. `std::find`, bug ID, 2.9.8 8.8. `std::find` TODO 2.9.8 8.8. `std::find` issue. 2.9.8 8.8. `std::find` (2.9.8 8.8. `std::find`) 2.9.8 8.8. `std::find` TODO 2.9.8 8.8. `std::find`. 2.9.8 8.8. `std::find` TODO 2.9.8 8.8. `std::find`, 2.9.8 8.8. `std::find` TODO 2.9.8 8.8. `std::find`.

```
// TODO(kl@gmail.com): Use a "*" here for concatenation operator.
// TODO(Zeke) change this to use relations.
// TODO(bug 12345): remove the "Last visitors" feature
```

2.9.8 8.8. `std::find` TODO 2.9.8 8.8. `std::find` “2.9.8 8.8. `std::find`”, 2.9.8 8.8. `std::find` “Fix by November 2005”, 2.9.8 8.8. `std::find` (“Remove this code when all clients can handle XML responses.”).

## 2.9.9 8.9. 2.9.9

2.9.9

DEPRECATED comments

DEPRECATED 2.9.9, 2.9.9. 2.9.9, 2.9.9.

2.9.9 DEPRECATED 2.9.9, 2.9.9, 2.9.9.

2.9.9, 2.9.9. 2.9.9, 2.9.9, 2.9.9.

DEPRECATED 2.9.9, 2.9.9, 2.9.9.

2.9.9, 2.9.9. 2.9.9, 2.9.9.

## 2.9.10 2.9.10 (YuleFox) 2.9.10

1. 2.9.9, 2.9.9 C++ 2.9.9, C coders 2.9.9, 2.9.9;
2. 2.9.9, 2.9.9;
3. 2.9.9, 2.9.9, 2.9.9;
4. 2.9.9 Chinese coders 2.9.9, 2.9.9, it is a problem, 2.9.9, 2.9.9, 2.9.9;
5. 2.9.9, 2.9.9. 2.9.9 (2.9.9), UNIX/LINUX 2.9.9 tab 2.9.9 space, 2.9.9 space;
6. TODO 2.9.9, 2.9.9, 2.9.9, 2.9.9, 2.9.9, 2.9.9.

## 2.10 9. 2.10

2.10, 2.10, 2.10. 2.10, 2.10, 2.10, 2.10, 2.10, 2.10.

2.10, 2.10 emacs 2.10.

### 2.10.1 9.1. 2.10.1

2.10.1

2.10.1 80.

2.10.1, 2.10.1, 2.10.1.

2.10.1

2.10.1, 2.10.1, 2.10.1. 2.10.1, 2.10.1. 2.10.1, 2.10.1, 2.10.1.

2.10.1

2.10.1. 80 2.10.1 60 2.10.1; 2.10.1, 2.10.1.

2.10.1

80 2.10.1.

2.10.1, 2.10.1 80 2.10.1, 2.10.1. 2.10.1, 2.10.1 URL 2.10.1 80 2.10.1.

```
#include <stdint.h>
```

```
uint8_t array[10];
```

## 2.10.2 9.2. ASCII

ASCII

ASCII, UTF-8.

ASCII

ASCII, UTF-8, ASCII. ASCII, UTF-8, UTF-8 ASCII; (UTF-8) ASCII. ASCII, UTF-8, UTF-8 UTF-8.

UTF-8, UTF-8 — "\xEF\xBB\xBF", u8"\uFEFF", Unicode UTF-8 UTF-8 UTF-8 UTF-8 UTF-8 UTF-8.

(Yang.Y: "\xEF\xBB\xBF" UTF-8 with BOM)

u8 uXXXX UTF-8. UTF-8 UTF-8 UTF-8, UTF-8, UTF-8.

C++11 char16\_t char32\_t, UTF-8 wchar\_t, Windows API, wchar\_t.

## 2.10.3 9.3. UTF-8

UTF-8

UTF-8, 2.

UTF-8

UTF-8. UTF-8. UTF-8.

## 2.10.4 9.4. UTF-16

UTF-16

UTF-16, UTF-16, UTF-16, UTF-16.

UTF-16

UTF-16:

```
ReturnType ClassName::FunctionName(Type par_name1, Type par_name2) {
    DoSomething();
    ...
}
```

UTF-16, UTF-16:

```
ReturnType ClassName::ReallyLongFunctionName(Type par_name1, Type par_name2,
                                                Type par_name3) {
    DoSomething();
    ...
}
```

~~~~~:

```
ReturnType LongClassName::ReallyReallyReallyLongFunctionName(
    Type par_name1,    // 4 space indent
    Type par_name2,
    Type par_name3) {
    DoSomething();    // 2 space indent
    ...
}
```

~~~~~:

- ~~~~~.
- ~~~~~, ~~~~~.
- ~~~~~, ~.
- ~~~~~, ~~~.
- ~~~~~.
- ~~~~~.
- ~~~~~.
- ~~~~~, ~~~~~.
- ~~~~~, ~~~~~.
- ~~~~~.
- ~~~~~.
- ~~~~~ 2 ~~~.
- ~~~~~ 4 ~~~~~.

~~~~~, ~~~~~, ~~~~~:

```
class Foo {
public:
    Foo(Foo&&);
    Foo(const Foo&);
    Foo& operator=(Foo&&);
    Foo& operator=(const Foo&);
};
```

~~~~~, ~~~~~:

```
class Shape {
public:
    virtual void Rotate(double radians) = 0;
};

class Circle : public Shape {
public:
    void Rotate(double radians) override;
};

void Circle::Rotate(double /*radians*/) {}
```

```
// 2 - 000000000, 000000000.
void Circle::Rotate(double) {}
```

??, 00000000, 000000000000, 00000000:

```
MUST_USE_RESULT bool IsOK();
```

## 2.10.5 9.5. Lambda ???

??

Lambda 00000000000000000000, 000000, 00000000.

??

000000, 00000 & 000000.

```
int x = 0;
auto add_to_x = [&x](int n) { x += n; };
```

1 lambda 0000000000.

```
std::set<int> blacklist = {7, 8, 9};
std::vector<int> digits = {3, 9, 1, 8, 4, 7, 1};
digits.erase(std::remove_if(digits.begin(), digits.end(), [&blacklist](int i) {
    return blacklist.find(i) != blacklist.end();
}),
    digits.end());
```

## 2.10.6 9.6. ????

??

0000000000, 000000000000, 0000000000000. 0000000000, 00000000, 000000000000000000.

??

000000000000

```
bool retval = DoSomething(argument1, argument2, argument3);
```

00000000, 00000, 00000000000000, 000000000000000000

```
bool retval = DoSomething(averyveryveryverylongargument1,
    argument2, argument3);
```

0000000000, 000000

```
if (...) {
    ...
    ...
    if (...) {
        DoSomething(
            argument1, argument2, // 4 0000
            argument3, argument4);
    }
}
```

, , . , , . , , . , , .  
 ,

, , ,

```
int my_heuristic = scores[x] * y + bases[x];
bool retval = DoSomething(my_heuristic, x, y, z);
```

,

```
bool retval = DoSomething(scores[x] * y + bases[x], // Score heuristic.
                          x, y, z);
```

, , .

,

```
// 3x3 widget.
my_widget.Transform(x1, x2, x3,
                    y1, y2, y3,
                    z1, z2, z3);
```

## 2.10.7 9.7.

.

.

.

, , , {} , .

```
// .
return {foo, bar};
functioncall({foo, bar});
pair<int, int> p{foo, bar};

// .
SomeFunction(
    {"assume a zero-length name before {}"}, // { .
    some_other_function_parameter);
SomeType variable{
    some, other, values,
    {"assume a zero-length name before {}"}, // { .
    SomeOtherType{
        "Very long string requiring the surrounding breaks.", // , .
        some, other values},
    SomeOtherType{"Slightly shorter string", // .
        some, other, values}};
SomeType variable{
    "This is too long to fit all in one line"; // , .
MyType m = { // , { .
    superlongvariablename1,
    superlongvariablename2,
    {short, interior, list},
    {interiorwrappinglist,
    interiorwrappinglist2}};
```





```
// C++ - IF THEN ELSE
if (condition) {
    foo;
} else
    bar;

// C++ - ELSE THEN IF
if (condition)
    foo;
else {
    bar;
}
```

```
// C++, C++
if (condition) {
    foo;
} else {
    bar;
}
```

## 2.10.9 9.9. C++

C++

switch (expression), cases (expression). expression, expression. expression {} continue.

C++

switch (expression) case (expression), expression. expression, expression.

expression case (expression), switch (expression) default (expression case (expression), expression warning). default (expression), expression assert:

```
switch (var) {
    case 0: { // 2
        ... // 4
        break;
    }
    case 1: {
        ...
        break;
    }
    default: {
        assert(false);
    }
}
```

expression, expression

```
for (int i = 0; i < kSomeNumber; ++i)
    printf("I love you\n");

for (int i = 0; i < kSomeNumber; ++i) {
    printf("I take it back\n");
}
```

expression {} continue, expression.

```
while (condition) {
    // 
}
for (int i = 0; i < kSomeNumber; ++i) {} // - 
while (condition) continue; // - continue
```

```
while (condition); // - while/loop
```

## 2.10.10 9.10.

(\*, &)

:

```
x = *p;
p = &x;
x = r.y;
x = r->y;
```

:

- ,
- \* &

, :

```
// , 
char *c;
const string &str;

// , 
char* c;
const string& str;
```

```
int x, *y; // - & *
char * c; // - * 
const string & str; // - &
```

, , ,

## 2.10.11 9.11.

,

, (&) :

```
if (this_one_thing > this_other_thing &&
    a_third_thing == a_fourth_thing &&
    yet_another && last_one) {
    ...
}
```

2.10.14 9.14. `constexpr``??``constexpr int, constexpr.``??``constexpr int, constexpr.`

```
// 2 - constexpr
if (lopsided_score) {
#ifdef DISASTER_PENDING // 22 - constexpr
    DropEverything();
# if NOTIFY // 222 - # 2222
    NotifyClient();
# endif
#endif
    BackToNormal();
}
```

```
// 2 - 2222
if (lopsided_score) {
#ifdef DISASTER_PENDING // 2 - "#if" 2222222
    DropEverything();
#endif // 2 - "#endif" 2222
    BackToNormal();
}
```

2.10.15 9.15. `public`, `protected`, `private``??``public::, protected::, private::, 22222 1 222.``??``222 (2222222222, 22 222) 22222222:`

```
class MyClass : public OtherClass {
public: // 2222222222
    MyClass(); // 22222222
    explicit MyClass(int var);
    ~MyClass() {}

    void SomeFunction();
    void SomeFunctionThatDoesNothing() {
    }

    void set_some_var(int var) { some_var_ = var; }
    int some_var() const { return some_var_; }

private:
    bool SomeInternalFunction();

    int some_var_;
    int some_other_var_;
};
```

~~~~:

- ~~~~~ 80 ~~~~~.
- ~~~ public:,protected:,private: ~~~ 1 ~~~.
- ~~~~~ (~~~ public) ~, ~~~~~ . ~~~~~.
- ~~~~~.
- public ~~~~~, ~~~ protected, ~~~ private.
- ~~~~~ ~~~ ~.

## 2.10.16 9.16. ~~~~~

~~

~~~~~.

~~

~~~~~:

```
// ~~~~~:
MyClass::MyClass(int var) : some_var_(var) {
    DoSomething();
}

// ~~~~~,
// ~~~~~, ~~~ 4 ~~~
MyClass::MyClass(int var)
    : some_var_(var), some_other_var_(var + 1) {
    DoSomething();
}

// ~~~~~, ~~~~~
// ~~~~~
MyClass::MyClass(int var)
    : some_var_(var),           // 4 space indent
      some_other_var_(var + 1) { // lined up
    DoSomething();
}

// ~~~~ } ~~~~~ { ~~~~~
// ~~~~~
MyClass::MyClass(int var)
    : some_var_(var) {}
```

2.10.17 9.17. `namespace``foo``namespace foo {``foo``namespace foo { namespace bar {`

```
namespace {
void foo() { // foo. namespace foo.
    ...
}
} // namespace
```

`namespace foo {`

```
namespace {
    // foo, namespace.
    void foo() {
        ...
    }
} // namespace
```

`namespace foo { namespace bar {`

```
namespace foo {
namespace bar {
```

2.10.18 9.18. `using``foo``namespace foo { namespace bar { namespace baz {``foo``foo`

```
void f(bool b) { // namespace foo.
    ...
int i = 0; // namespace.
// namespace.
// namespace, namespace.
int x[] = { 0 };
int x[] = {0};

// namespace.
class Foo : public Bar {
public:
    // namespace, namespace
```

(continues on next page)



????

```
// 比较(< and >) 字符串, < 字符串, > 0 ( 字符串).
vector<string> x;
y = static_cast<char*>(x);

// 字符串字符串字符串字符串, 字符串.
vector<char *> x;
```

## 2.10.19 9.19. ????

??

????????.

??

????????????????: ?????, ??????. ???: ????????????????? 2 ?, ?????????, ?????????????.

?????: ?????????????, ??????????. ??, ?????????????????????????, ??????????. ?????????????.

????????????????:

- ?????????????????.
- ??? if-else ?????????????.

## 2.10.20 ?? (YuleFox) ??

1. ?????, ??, ?????????, ?????????????????;
2. ????????? 80 ?, ? 22 ?????????, ??????;
3. ????? ASCII ??, ?????, ?? UTF-8 ?? (??? UNIX/Linux ?, Windows ?????), ?????????, ?????????, ?????????;
4. UNIX/Linux ?????????, MSVC ??? Tab ?????;
5. ???, ???, ????: ?????????????????, ?????????;
6. ?????????????????, ???/2/??/???, ?????????????, ?????????;
7. ./-> ?????????, \*& ?????, ???, ?????????;
8. ???/????????????, ?/??/??/??/????;
9. ??? = ?? () ?????, ???;
10. return ??? ();
11. ??/????????, ??????.
12. ?? UNIX/Linux ????????????????? (.cc ?????, ?????), ?????????????, ?????????????????????, ?????????????????; Windows ?????????????????????.





- ??, ? Windows ??????????????????:

- 2.12 11.

??, ?????????; ??????. ?????!



## OBJECTIVE-C [ ] [ ] [ ] [ ] - [ ] [ ] [ ] [ ]

### 3.1 Google Objective-C Style Guide [?][?][?]

??

2.36

???

# Mike Pinkerton

Greg Miller

Dave MacLachlan

??

ewangke

Yang.Y

□□□□

- Google Style Guide
- Google [Developer Style Guide](#) - [FAQ](#)

### 3.1.1

ewanke

```

##### style guide #####7#####vim#####HTML#####
"ewangke at gmail.com" 2011.03.27

```

Yang.Y

Objective-C C/C++

- 2.36
-

### 3.1.2

Objective-C C Mac OS X iPhone

Cocoa Mac OS X Objective-C Mac OS X

Objective-C Google C++ Objective-C Google

- Apple's Cocoa Coding Guidelines
- Google's Open Source C++ Style Guide

Note: Google C++ Objective-C++

Mac OS X Google

Google Google Toolbox for Mac project GTM GTM

Objective-C Objective-C Objective-C The Objective-C Programming Language

### 3.1.3

```

@interface

```

```

// Foo.h
// AwesomeProject
//
// Created by Greg Miller on 6/13/08.
// Copyright 2008 Google, Inc. All rights reserved.
//

#import <Foundation/Foundation.h>

// A sample class demonstrating good Objective-C style. All interfaces,
// categories, and protocols (read: all top-level declarations in a header)
// MUST be commented. Comments must also be adjacent to the object they're
// documenting.
//
// (no blank line between this comment and the interface)
@interface Foo : NSObject {
    @private
    NSString *bar_;
    NSString *bam_;
}

// Returns an autoreleased instance of Foo. See -initWithBar: for details
// about |bar|.
+ (id)fooWithBar:(NSString *)bar;

// Designated initializer. |bar| is a thing that represents a thing that

```

(continues on next page)

(continued from previous page)

```
// does a thing.
- (id)initWithBar:(NSString *)bar;

// Gets and sets |bar_|.
- (NSString *)bar;
- (void)setBar:(NSString *)bar;

// Does some work with |blah| and returns YES if the work was completed
// successfully, and NO otherwise.
- (BOOL)doWorkWithBlah:(NSString *)blah;

@end
```

Implementation getters & setters & init & dealloc

```
//
// Foo.m
// AwesomeProject
//
// Created by Greg Miller on 6/13/08.
// Copyright 2008 Google, Inc. All rights reserved.
//

#import "Foo.h"

@implementation Foo

+ (id)fooWithBar:(NSString *)bar {
    return [[[self alloc] initWithBar:bar] autorelease];
}

// Must always override super's designated initializer.
- (id)init {
    return [self initWithBar:nil];
}

- (id)initWithBar:(NSString *)bar {
    if ((self = [super init])) {
        bar_ = [bar copy];
        bam_ = [[NSString alloc] initWithFormat:@"hi %d", 3];
    }
    return self;
}

- (void)dealloc {
    [bar_ release];
    [bam_ release];
    [super dealloc];
}

- (NSString *)bar {
    return bar_;
}

- (void)setBar:(NSString *)bar {
```

(continues on next page)

(continued from previous page)

```

    [bar_ autorelease];
    bar_ = [bar copy];
}

- (BOOL)doWorkWithBlah:(NSString *)blah {
    // ...
    return NO;
}

@end

```

```

@interface @implementation @end
@interface { }

```

## 3.2

### 3.2.1 vs.

Tip:

```


```

### 3.2.2

80

Objective-C 80

80

Xcode > Preferences > Text Editing > Show page guide

### 3.2.3

Tip:

- / +

```


```

```

- (void)doSomethingWithString:(NSString *)theString {
    ...
}

```

```


```

```


```

```

- (void)doSomethingWith: (GTMFoo *)theFoo
    rect: (CGRect)theRect
    interval: (float)theInterval {
    ...
}

```

4

```

- (void)short: (GTMFoo *)theFoo
    longKeyword: (CGRect)theRect
    evenLongerKeyword: (float)theInterval {
    ...
}

```

### 3.2.4

Tip:

```
[myObject doFooWith:arg1 name:arg2 error:arg3];
```

```
[myObject doFooWith:arg1
    name:arg2
    error:arg3];
```

```
[myObject doFooWith:arg1 name:arg2 // some lines with >1 arg
    error:arg3];

[myObject doFooWith:arg1
    name:arg2 error:arg3];

[myObject doFooWith:arg1
    name:arg2 // aligning keywords instead of colons
    error:arg3];
```

```
[myObj short:arg1
    longKeyword:arg2
    evenLongerKeyword:arg3];
```



### 3.2.5 @public @private

Tip: @public @private

C++ public, private protected

```
@interface MyClass : NSObject {
    @public
    ...
    @private
    ...
}
@end
```

### 3.2.6

Tip: @ @ { } @catch

Objective-C

```
@try {
    foo();
}
@catch (NSException *ex) {
    bar(ex);
}
@finally {
    baz();
}
```

### 3.2.7

Tip:

```
@interface MyProtooledClass : NSObject<NSWindowDelegate> {
    @private
    id<MyFancyDelegate> delegate_;
}
- (void) setDelegate: (id<MyFancyDelegate>) aDelegate;
@end
```

### 3.2.8 Block Syntax

Tip: A block is a target/selector that is a single expression. 4 spaces

Block Syntax

- Block Syntax
- Block Syntax
- Block Syntax 4 spaces
- Block Syntax 20 spaces
- Block Syntax { } { }
- Block Syntax

```
// The entire block fits on one line.
[operation setCompletionBlock:^( [self onOperationDone]; ]];

// The block can be put on a new line, indented four spaces, with the
// closing brace aligned with the first character of the line on which
// block was declared.
[operation setCompletionBlock:^(
    [self.delegate newDataAvailable];
)];

// Using a block with a C API follows the same alignment and spacing
// rules as with Objective-C.
dispatch_async(fileIOQueue_, ^{
    NSString* path = [self sessionFilePath];
    if (path) {
        // ...
    }
});

// An example where the parameter wraps and the block declaration fits
// on the same line. Note the spacing of |(SessionWindow *window) {|
// compared to |^{| above.
[[SessionService sharedService]
    loadWindowWithCompletionBlock:^(SessionWindow *window) {
        if (window) {
            [self windowDidLoad:window];
        } else {
            [self errorLoadingWindow];
        }
    }];

// An example where the parameter wraps and the block declaration does
// not fit on the same line as the name.
[[SessionService sharedService]
    loadWindowWithCompletionBlock:
        ^(SessionWindow *window) {
            if (window) {
                [self windowDidLoad:window];
            } else {
                [self errorLoadingWindow];
            }
        }];
```

(continues on next page)

(continued from previous page)

```

    }
    }];

// Large blocks can be declared out-of-line.
void (^largeBlock)(void) = ^{
    // ...
};
[operationQueue_ addOperationWithBlock:largeBlock];

```

### 3.3

Objective-C

Objective-C [Objective-C naming rules](#) C++ [Google C++](#) [Objective-C](#)

URL TIFF EXIF

Objective-C++ C++ API Objective-C Cocoa C++ Cocoa

@implementation Objective-C C++ C++

#### 3.3.1

Tip: –

|     |                   |
|-----|-------------------|
| .h  | C/C++/Objective-C |
| .m  | Objective-C       |
| .mm | Objective-C++     |
| .cc | C++               |
| .c  | C                 |

GTMNSString+Utils.h GTMNSTextView+Autocomplete.h

#### 3.3.2 Objective-C++

Tip: Objective-C++

Cocoa/Objective-C C++ @implementation Objective-C C++ C++

```

// file: cross_platform_header.h

class CrossPlatformAPI {
public:
    ...
    int DoSomethingPlatformSpecific(); // impl on each platform
private:
    int an_instance_var_;
};

// file: mac_implementation.mm
#include "cross_platform_header.h"

// A typical Objective-C class, using Objective-C naming.
@interface MyDelegate : NSObject {
    @private
    int instanceVar_;
    CrossPlatformAPI* backEndObject_;
}
- (void)respondToSomething:(id) something;
@end

@implementation MyDelegate
- (void)respondToSomething:(id) something {
    // bridge from Cocoa through our C++ backend
    instanceVar_ = backEndObject->DoSomethingPlatformSpecific();
    NSString* tempString = [NSString stringWithInt:instanceVar_];
    NSLog(@"%@", tempString);
}
@end

// The platform-specific implementation of the C++ class, using
// C++ naming.
int CrossPlatformAPI::DoSomethingPlatformSpecific() {
    NSString* temp_string = [NSString stringWithInt:an_instance_var_];
    NSLog(@"%@", temp_string);
    return [temp_string intValue];
}

```

### 3.3.3

Tip:

GTMSendMessage

### 3.3.4

Tip:

NSString GTMNSString+Parsing.h GTMStringParsingAdditions  
 gtm\_myCategoryMethodOnAString: Objective-C

### 3.3.5 Objective-C

Tip:

convertPoint:fromRect: re-  
 placeCharactersInRange:withString: Apple's Guide to Naming Methods  
 get

```
- (id)getDelegate; // AVOID
- (id)delegate;    // GOOD
```

Objective-C C++ C++

### 3.3.6

Tip: myLocalVariable myInstanceVariable\_  
 Objective-C 2.0 @property KVO/KVC

int

- 

```
int w;
int nerr;
int nCompConns;
tix = [[NSMutableArray alloc] init];
obj = [someObject object];
p = [network port];
```

- 

```
int numErrors;
int numCompletedConnections;
tickets = [[NSMutableArray alloc] init];
userInfo = [someObject object];
port = [network port];
```

????

```

usernameTextField_
Objective-C
2.0 KVO/KVC KVO=Key Value Observing KVC=Key
Value Coding
Objective-C
2.0@property @synthesize

```

??

```

k kInvalidHandle kWritePerm

```

### 3.4 ??

```


```

```


```

```

C++

```

#### 3.4.1 ????

---

Tip:

---

??????

```


```

- 
- 
- Copyright 2008 Google Inc.
- Apache 2.0, BSD, LGPL, GPL

```


```

#### 3.4.2 ???????

---

Tip:

---

```

// A delegate for NSApplication to handle notifications about app
// launch and shutdown. Owned by the main app controller.
@interface MyAppDelegate : NSObject {
    ...
}
@end

```

“”

### 3.4.3

Tip: |

count

```
// Sometimes we need |count| to be less than zero.
```

```
// Remember to call |StringWithoutSpaces("foo bar baz")|
```

### 3.4.4

Tip: Objective-C

NSObject retained weak \_\_weak  
retained @property Mac IBOutlet-  
lets retained

CoreFoundation C++ Objective-C re-  
tained \_\_strong \_\_weak CoreFoundation Objective-C  
\_\_weak clang C++

Objective-C C++

```
@interface MyDelegate : NSObject {
    @private
    IBOutlet NSButton *okButton_; // normal NSControl; implicitly weak on Mac only

    AnObjcObject* doohickey_; // my doohickey
    __weak MyObjcParent *parent_; // so we can send msgs back (owns me)

    // non-NSObject pointers...
    __strong CWackyCppClass *wacky_; // some cross-platform object
    __strong CFDictionaryRef *dict_;
}
@property(strong, nonatomic) NSString *doohickey;
@property(weak, nonatomic) NSString *parent;
@end
```

- retained - retained

## 3.5 Cocoa Objective-C

### 3.5.1 @private

Tip: @private

```
@interface MyClass : NSObject {
    @private
    id myInstanceVariable_;
}
// public accessors, setter takes ownership
- (id)myInstanceVariable;
- (void)setMyInstanceVariable:(id)theVar;
@end
```

### 3.5.2

Tip:

### 3.5.3

Tip: init... bug

bug

### 3.5.4 NSObject

Tip: NSObject @implementation

```
init...copyWithZone: dealloc init...copyWithZone:
dealloc
```





### 3.5.8 #import and #include

Tip: #import Objective-C/Objective-C++ `filename` #include C/C++ `filename`

```
filename #import filename #include filename
```

- `filename` Objective-C/Objective-C++ `filename` #import `filename`
- `filename` C/C++ `filename` #include `filename` #define `filename`

```
filename Objective-C filename #define filename #import filename Objective-C filename Objective-C
filename #import filename
```

```
filename Objective-C filename C/C++ filename C/C++ filename C/C++ filename #import filename
#include filename Objective-C filename #include filename
```

```
filename Mac filename C filename C++ filename #define filename Mac
filename #import filename #include filename #include filename
```

```
#import <Cocoa/Cocoa.h>
#include <CoreFoundation/CoreFoundation.h>
#import "GTMFoo.h"
#include "base/basictypes.h"
```

### 3.5.9 filename

Tip: #import filename

```
filename Cocoa filename Foundation filename
#import filename #include filename Objective-C filename
```

```
#import <Foundation/Foundation.h> // good
#import <Foundation/NSArray.h> // avoid
#import <Foundation/NSString.h>
...
```

### 3.5.10 filename autorelease

Tip: filename autorelease filename release

```
filename release filename return filename
```

```
// AVOID (unless you have a compelling performance reason)
MyController* controller = [[MyController alloc] init];
// ... code here that might return ...
[controller release];

// BETTER
MyController* controller = [[MyController alloc] init] autorelease];
```

### 3.5.11 autorelease retain

Tip: autorelease `` `` retain

“” “autorelease retain”  
autorelease

```
- (void)setFoo:(GMFoo *)aFoo {
    [foo_ autorelease]; // Won't dealloc if |foo_| == |aFoo|
    foo_ = [aFoo retain];
}
```

### 3.5.12 init dealloc

Tip: init dealloc

init dealloc ival

```
- (id)init {
    self = [super init];
    if (self) {
        bar_ = [[NSMutableString alloc] init]; // good
    }
    return self;
}

- (void)dealloc {
    [bar_ release]; // good
    [super dealloc];
}
```

```
- (id)init {
    self = [super init];
    if (self) {
        self.bar = [NSMutableString string]; // avoid
    }
    return self;
}

- (void)dealloc {
    self.bar = nil; // avoid
    [super dealloc];
}
```

### 3.5.13 dealloc

Tip: dealloc @interface

dealloc retained

dealloc retained @interface dealloc

### 3.5.14 setter

Tip: NSString setter copy

retain NSString NSMutableString

```
- (void)setFoo:(NSString *)aFoo {
    [foo_ autorelease];
    foo_ = [aFoo copy];
}
```

### 3.5.15

Tip: @throw Objective-C OS

-fobjc-exceptions @synchronized @throw @try @catch @finally

NS\_DURING NS\_HANDLER NS\_ENDHANDLER NS\_VALUEReturn NS\_VOIDRETURN Mac OS X 10.2

Objective-C Objective-C++

```
class exceptiontest {
public:
    exceptiontest() { NSLog(@"Created"); }
    ~exceptiontest() { NSLog(@"Destroyed"); }
};

void foo() {
    exceptiontest a;
    NSException *exception = [NSException exceptionWithName:@"foo"
                                                             reason:@"bar"
                                                             userInfo:nil];
    @throw exception;
}

int main(int argc, char *argv[]) {
    GMACoreAutoreleasePool pool;
    @try {
```

(continues on next page)

(continued from previous page)

```

    foo();
}
@catch(NSException *ex) {
    NSLog(@"exception raised");
}
return 0;
}

```

~~~~~

~~~~~ smartptr~~~~~ shared\_ptr~~~~~linked\_ptr~~~~~ STL  
 ~~~~~ Objective-C++ ~~~~~ C++ ~~~~~ Objective-C ~~~~~  
 @try~~~~~@catch ~~~~~@finally ~~~~~ C++ ~~~~~

### 3.5.16 nil

Tip: nil

nil Objective-C nil OS X Apple's documentation

C/C++ "NULL" C/C++ C/C++

### 3.5.17 BOOL

Tip: BOOL YES

Objective-C BOOL YES``(1)``NO``(0)``  
 ``BOOL`` BOOL NO  
 BOOL YES NO 256 256...12

BOOL Bool bool C++ Std 4.7.4, 4.12 C99 Std 6.3.1.2 BOOL Boolean  
 Boolean Objective-C BOOL

BOOL&&| ! BOOL

~~~~~

```

- (BOOL)isBold {
    return [self fontTraits] & NSFontBoldTrait;
}
- (BOOL)isValid {
    return [self stringValue];
}

```

~~~~~

```

- (BOOL)isBold {
    return ([self fontTraits] & NSFontBoldTrait) ? YES : NO;
}

```

(continues on next page)

(continued from previous page)

```

- (BOOL)isValid {
    return [self stringValue] != nil;
}
- (BOOL)isEnabled {
    return [self isValid] && [self isBold];
}

```

YES/NO BOOL

```

BOOL great = [foo isGreat];
if (great == YES)
    // ...be great!

```

```

BOOL great = [foo isGreat];
if (great)
    // ...be great!

```

### 3.5.18 Property

Tip: Property Objective-C 2.0 iPhone Mac OS X 10.5 (Leopard) @property

@synthesize

```

@interface MyClass : NSObject {
    @private
    NSString *name_;
}
@property(copy, nonatomic) NSString *name;
@end

@implementation MyClass
@synthesize name = name_;
@end

```

??

```

@implementation
@interface
@implementation

```

```

@interface MyClass : NSObject {
    @private
    NSString *name_;
}
@property(copy, nonatomic) NSString *name;
@end

@implementation MyClass
@synthesize name = name_;
- (id)init {
    ...
}
@end

```

copy @Attribute

```

copy @attribute NSString @property
NSString @ setter @ copy @ retain @

```

???

```

@property@synthesize @ setter @ getter @ get @ set
@property nonatomic

```

???

Objective-C 2.0 set/get

???

```

NSString *oldName = myObject.name;
myObject.name = @"Alice";

```

???

```

NSArray *array = [[NSArray arrayWithObject:@"hello"] retain];

NSUInteger numberOfItems = array.count; // not a property
array.release;                          // not a property

```

### 3.5.19 `__interface`

Tip: `__interface` is deprecated.

`__interface`

```
@interface MyClass : NSObject // Does a lot of stuff - (void)fooBarBam; @end
```

`__interface`

```
@interface MyClass : NSObject { } // Does a lot of stuff - (void)fooBarBam; @end
```

### 3.5.20 `__synthesize`

Tip: `__synthesize` is deprecated on iOS.

```
__synthesize __synthesize var = var_; __synthesize self.var = blah; __synthesize var = blah;
```

```
__synthesize CType __synthesize CType __synthesize @dynamic CType __synthesize CType retain  
retain release __synthesize getter setter __synthesize  
__synthesize @dynamic __synthesize
```

```
// Header file
@interface Foo : NSObject
// A guy walks into a bar.
@property(nonatomic, copy) NSString *bar;
@end

// Implementation file
@interface Foo ()
@property(nonatomic, retain) NSArray *baz;
@end

@implementation Foo
__synthesize bar = bar_;
__synthesize baz = baz_;
@end
```

## 3.6 Cocoa

### 3.6.1

Tip: `retain`

`retain`

1. `delegate_`
2. `delegate` `setDelegate:`



3. delegate\_ retain

### 3.6.2 3/3/333MVC

---

Tip: 使用 API 的 @protocol

---

- 使用 API 的 @protocol
- 使用 API 的 “” 使用 API 的 “”
- 使用 @protocol API 的 @optional `` Objective-C 1.0 使用 @optional “”

## PYTHON 开发指南 - 第四版

## 4.1 致谢

感谢

Amit Patel  
 Antoine Picard  
 Eugene Jhong  
 Jeremy Hylton  
 Matt Smart  
 Mike Shields

感谢所有在 GitHub 上

的

guoqiao v2.19  
 xuxinkun v2.59  
 captainfffsama v2.6  
 2023 年 4 月 16 日

感谢

- Google Style Guide (2015)
- Google 开发指南 - 2015

20

Python 2.0 许可证, 2003 CC-BY 3.0 20.

## 4.2 附录

Python 开发指南 Python 开发指南  
 开发指南 Vim Emacs  
 Black Pyink 开发指南, 2023

## 4.3 Python

### 4.3.1 Lint

Tip: `pylintrc` `pylint`, `pylint`.

##:

`pylint` `Python` `bug` `pylint`. `pylint` `C` `C++` `pylint`(`pylint`: `pylint`less `dynamic`)`pylint`. `Python``pylint`, `pylint`. `py`, `pylint`.

##:

`pylint`, `pylint`, `pylint`.

##:

`pylint` `pylint`. `pylint`, `pylint`: a) `pylint` b) `pylint` `py` c) `pylint`.

##:

`pylint``pylint`.

`pylint`, `pylint``pylint`. `py`:

```
def do_PUT(self): # WSGI pylint: disable=invalid-name
    ...
```

`pylint`(`empty-docstring`)`pylint`. `pylint` `g-` `pylint`.

`pylint`

`pylint`, `pylint`.

`pylint` `pylint --list-msgs` `pylint` `pylint`. `pylint` `pylint --help-msg=invalid-name` `pylint`.

`pylint` `pylint: disable-msg`, `pylint` `pylint: disable`.

`pylint`"`pylint`"`pylint`. `pylint`. `pylint`"`pylint`". `py`:

```
def viking_cafe_order(spam: str, beans: str, eggs: str | None = None) -> str:
    del beans, eggs # pylint.
    return spam + spam + spam
```

(`Viking` `pylint`.)

`pylint`: `pylint`; `pylint` `unused_`; `pylint`. `pylint`.

### 4.3.2

Tip: `import` `pylint`, `pylint`, `pylint`

##:

`pylint`.

##:

`pylint`. `pylint`. `x.Obj` `Obj` `pylint` `x`.

¶:

¶¶¶¶¶¶¶¶¶¶. ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶.

¶:

1. ¶ import x ¶¶¶¶¶¶¶¶.
2. ¶ from x import y, ¶x¶¶¶¶, y¶¶¶¶¶¶¶¶¶¶.
3. ¶¶¶¶¶¶¶¶ from x import y as z: ¶¶¶¶¶¶¶¶¶¶ y; ¶¶ y ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶; ¶¶ y ¶¶¶¶¶¶¶¶¶.
4. ¶¶¶¶ z ¶¶¶¶¶¶¶¶¶¶¶¶ import y as z.(¶¶ np ¶¶ numpy.)

¶¶, ¶¶¶¶¶¶¶¶¶¶¶¶ sound.effects.echo:

```
from sound.effects import echo
...
echo.EchoFilter(input, output, delay=0.7, atten=4)
```

¶¶¶¶¶¶¶¶¶¶¶¶. ¶¶¶¶¶¶¶¶¶¶¶¶, ¶¶¶¶¶¶¶¶. ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶.

¶:

¶¶¶¶¶¶¶¶¶¶

1. ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶:
  1. typing ¶¶
  2. collections.abc ¶¶
  3. typing\_extensions ¶¶
2. six.moves ¶¶¶¶¶¶¶¶.

### 4.3.3 ¶

Tip: ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶.

¶:

¶¶¶¶¶¶¶¶, ¶¶¶. ¶¶¶¶¶¶¶¶¶¶.

¶:

¶¶¶¶¶¶¶¶, ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶. ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶.

¶:

¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶.

¶¶¶¶¶¶¶¶¶¶:

¶:

```
# ¶¶¶¶¶¶¶¶¶¶¶¶ absl.flags (¶¶¶¶).
import absl.flags
from doctor.who import jodie

_FOO = absl.flags.DEFINE_string(...)
```

```
# [REDACTED] flags ([REDACTED]).
from absl import flags
from doctor.who import jodie

_FOO = flags.DEFINE_string(...)
```

11: (11111111 jodie.py 1111 doctor/who/ 1)

```
# ??????????????????????????????
# ????????????????????? sys.path.
# ???? jodie ???? , ???????????
import jodie
```

```

#### sys.path ####, ####.  ##, ##### import jodie ##### jodie
##### jodie.py.

```

#### 4.3.4

Tip: ☐ ☐ ☐ ☐ ☐ ☐.

22: 22222222222222, 2222222222222222.

22: 222222222222222222222222. 222222, 2222222222222222. 22, 2222N22222222, 222222222222.

22: 2222222222. 22222222222222.

$\frac{1}{2}$ :  $\frac{1}{2}$ :

1. `?????????????????. ??, ? ValueError ?????????? (????????????????????????). ???? assert  
?????????API?????. ??? assert ??????????, ???? assert ??????????????????. ?????????????????,  
??? raise. ??:`

??:

```
def connect_to_next_port(self, minimum: int) -> int:
    """??????????????.

    ??:
        minimum: ??????? 1024 ??????.

    ??:
        ???????.

    ??:
        ConnectionError: ???????.

    """
    if minimum < 1024:
        # ??????? ValueError ?????????????????? API ?
        # ???????????????.
        raise ValueError(f'?????????? 1024?????? {minimum}.')
    port = self._find_next_open_port(minimum)
```

(continues on next page)

(continued from previous page)

```

if port is None:
    raise ConnectionError(
        f'port {minimum} is too small.')
assert port >= minimum, (
    f'port {port}, {minimum} is too small.')
return port

```

def:

```

def connect_to_next_port(self, minimum: int) -> int:
    """Connect to the next port.

    Args:
        minimum: The minimum port number.

    Returns:
        The next port number.

    """
    assert minimum >= 1024, 'minimum must be at least 1024.'
    port = self._find_next_open_port(minimum)
    assert port is not None
    return port

```

1. The `raise` statement is used to raise an exception. The syntax is `raise ErrorType('message')`. For example, `raise ValueError('foo')`.
2. The `assert` statement is used to assert that a condition is true. The syntax is `assert condition, 'message'`. For example, `assert x > 0, 'x must be positive'`.
3. The `except` statement is used to catch exceptions. The syntax is `except ExceptionType as e:`. For example, `except ValueError as e:`.
4. The `try/except` statement is used to handle exceptions. The syntax is `try: ... except ExceptionType: ...`. For example, `try: ... except ValueError: ...`.
5. The `finally` statement is used to execute code regardless of whether an exception was raised. The syntax is `finally: ...`. For example, `finally: ...`.

### 4.3.5

Tip: ...

def:

... (class attribute).

def:

...

def:

1. def: ... The ... (class attribute).
2. ...

??:

```
#####
```

```
#####, #####, ##### _ _____. #####,
#####. ?? ??? ?. #####.
```

```
#####,_MAX_HOLY_HANDGRENADE_COUNT = 3 #####,
SIR_LANCELOTS_FAVORITE_COLOR = "blue" ____API___. _____, _____.
?? ??? ?.
```

#### 4.3.6 ??/??/??????

Tip: \_\_\_\_\_.

??:

```
#####. #####. _____.
(____:#####,_____,_____ nonlocal)
```

??:

```
#####. _____. _____.
```

??:

```
#####. #####, _____.
```

??:

```
_____. _____, _____ self ? cls _____. _____.
_____, _____ _ __, _____.
```

#### 4.3.7 ??? (comprehension) ??? (generator expression)

Tip: \_\_\_\_\_.

??:

```
#####, _____ map() filter(), ____ lambda . (____:
_____, () _____)
```

??:

```
_____, _____. _____, _____.
```

??:

```
_____.
```

??:

```
_____. _____: _____for_____. _____for_____. _____, _____.
```

??:

```
result = [mapping_expr for value in iterable if filter_expr]

result = [{'key': value} for value in iterable
           if a_long_filter_expression(value)]

result = [complicated_transform(x)
```

(continues on next page)

(continued from previous page)

```

        for x in iterable if predicate(x)]

descriptive_name = [
    transform({'key': key, 'value': value}, color='black')
    for key, value in generate_iterable(some_input)
    if complicated_condition_is_met(key, value)
]

result = []
for x in range(10):
    for y in range(5):
        if x * y > 10:
            result.append((x, y))

return {x: complicated_transform(x)
        for x in long_generator_function(parameter)
        if x is not None}

squares_generator = (x**2 for x in range(10))

unique_names = {user.name for user in users if user is not None}

eat(jelly_bean for jelly_bean in jelly_beans
    if jelly_bean.color == 'black')

```

??:

```

result = [complicated_transform(
    x, some_argument=x+1)
    for x in iterable if predicate(x)]

result = [(x, y) for x in range(10) for y in range(5) if x * y > 10]

return ((x, y, z)
        for x in xrange(5)
        for y in xrange(5)
        if x != y
        for z in xrange(5)
        if y != z)

```

#### 4.3.8

Tip: , , .

?:

( in not in ).

?:

. , . , .

?:

( ). .

?:



adict.items() returns a list of (key, value) tuples. This is useful if you want to iterate over the items of a dictionary. For example, you can use it to iterate over the items of a dictionary and print the key and value for each item.

Tip:

```
for key in adict: ...
if obj in alist: ...
for line in afile: ...
for k, v in adict.items(): ...
```

Tip:

```
for key in adict.keys(): ...
for line in afile.readlines(): ...
```

### 4.3.9

Tip: `yield`.

Tip:

`yield` is used to return a value from a function. It is used to create a generator function. A generator function is a function that returns an iterator object. An iterator object is an object that represents a stream of values. You can iterate over an iterator object to get the next value in the stream.

Tip:

`yield` is used to return a value from a function. It is used to create a generator function. A generator function is a function that returns an iterator object. An iterator object is an object that represents a stream of values. You can iterate over an iterator object to get the next value in the stream.

Tip:

`yield` is used to return a value from a function. It is used to create a generator function. A generator function is a function that returns an iterator object. An iterator object is an object that represents a stream of values. You can iterate over an iterator object to get the next value in the stream.

Tip:

`yield` is used to return a value from a function. It is used to create a generator function. A generator function is a function that returns an iterator object. An iterator object is an object that represents a stream of values. You can iterate over an iterator object to get the next value in the stream.

(`yield` is used to return a value from a function. It is used to create a generator function. A generator function is a function that returns an iterator object. An iterator object is an object that represents a stream of values. You can iterate over an iterator object to get the next value in the stream.

`yield` is used to return a value from a function. It is used to create a generator function. A generator function is a function that returns an iterator object. An iterator object is an object that represents a stream of values. You can iterate over an iterator object to get the next value in the stream.

`yield` is used to return a value from a function. It is used to create a generator function. A generator function is a function that returns an iterator object. An iterator object is an object that represents a stream of values. You can iterate over an iterator object to get the next value in the stream.

### 4.3.10 Lambda

Tip: `map()`, `filter()`, `lambda`.

Tip:

`lambda` is used to create a lambda function. A lambda function is a small, anonymous function that can be used in a single line of code. It is often used for short, simple functions that are only needed once.

Tip:

`lambda` is used to create a lambda function. A lambda function is a small, anonymous function that can be used in a single line of code. It is often used for short, simple functions that are only needed once.

Tip:

`lambda` is used to create a lambda function. A lambda function is a small, anonymous function that can be used in a single line of code. It is often used for short, simple functions that are only needed once.

Tip:

`lambda` is used to create a lambda function. A lambda function is a small, anonymous function that can be used in a single line of code. It is often used for short, simple functions that are only needed once.

`lambda` is used to create a lambda function. A lambda function is a small, anonymous function that can be used in a single line of code. It is often used for short, simple functions that are only needed once.

### 4.3.11 Ternary

Tip: Ternary.

```

xx:
    xxxxx(yyyyyyyy)ifzzzzzz. xx: x = 1 if cond else 2.

xx:
    ifzzzzzz, xxx.

xx:
    xxxifzzzzzz. xxxxxxxxxxxxxxxxxxxxxxxx.

xx:
    xxxxxxxx. xxxxxxxxxxxxxxxx: xxx, if xxx else xx. xxxxxxxxxxxxxxxxifxx.

xx:

```

```

one_line = 'yes' if predicate(value) else 'no'
slightly_split = ('yes' if predicate(value)
                  else 'no, nein, nyet')
the_longest_ternary_style_that_can_be_done = (
    'yes, true, affirmative, confirmed, correct'
    if predicate(value)
    else 'no, false, negative, nay')

```

```

xx:

bad_line_breaking = ('yes' if predicate(value) else
                    'no') # xxxxxx
portion_too_long = ('yes'
                    if some_long_module.some_long_predicate_function(
                        really_long_variable_name)
                    else 'no, false, negative, nay') # xx

```

### 4.3.12 Lambda

Tip: Lambda.

```

xx:
    xxxxxxxxxxxxxxxxxxxxxxxx, xx, def foo(a, b = 0): . xxxfooxxxxxxx, xb0. xxxxxxxxxxxx,
    xbxxxxxxxxxx.

xx:
    xxx, xxxxxxxxxxxxxxxx, xxxxxxxxxxxxxxxx. xxxxxxxxxxxxxxxx, xxxxxxxxxxxxxxxx.
    xx, Pythonxxxxxxxxxx, xxxxxxxxxxxxxxxx"xx"xxxxx.

xx:
    xxxxxxxxxxxxxxxxxxxxxxxx. xxxxxxxxxxxxxxxx, xxxxxxxx. xxxxxxxxxxxxx(yyyyyyyy), xxxxxxxx.

xx:
    xxx, xxxxxxxxxxxxx:
    xxxxxxxxxxxxxxxx (mutable) xx.

xx:

```



### 4.3.14 True/False

Tip: `0, 0, None, [], {}, ""` are all False.

Python:

Python `False`. `0, 0, None, [], {}, ""` are all False.

Python:

Python `False`. `0, 0, None, [], {}, ""` are all False.

Python:

C/C++ `0, 0, None, [], {}, ""` are all False.

Python:

Python `False`. `0, 0, None, [], {}, ""` are all False.

1. Python `if foo is None: (foo is not None) None`. `0, 0, None, [], {}, ""` are all False.
2. Python `==` `False`. `0, 0, None, [], {}, ""` are all False. `0, 0, None, [], {}, ""` are all False.
3. Python `(foo, 0, 0) == (foo, 0, 0)`. `0, 0, None, [], {}, ""` are all False.
4. Python `False` `(0, 0, None, [], {}, "")`. `0, 0, None, [], {}, ""` are all False.

Python:

```
if not users:
    print('0')

if i % 10 == 0:
    self.handle_multiple_of_ten()

def f(x=None):
    if x is None:
        x = []
```

Python:

```
if len(users) == 0:
    print('0')

if not i % 10:
    self.handle_multiple_of_ten()

def f(x=None):
    x = x or []
```

5. Python `'0'` `(0, 0, None, [], {}, "")` are all True.
6. Python `0` Numpy `0` `(0, 0, None, [], {}, "")` are all True.

#### 4.3.15 Lexical Scoping, Closures

Tip: `scope()`.

¶:

Python's lexical scoping is implemented using closures. A closure is a function object that has references to variables in its enclosing scope. For example, a function defined inside another function has access to the variables in the scope of the outer function.

Example:

```
def get_adder(summand1: float) -> Callable[[float], float]:
    """Returns a function that adds summand1 to its argument."""
    def adder(summand2: float) -> float:
        return summand1 + summand2
    return adder
```

(Example: `fn = get_adder(1.2); sum = fn(3.4); sum == 4.6`.)

¶:

Lexical scoping is implemented in many languages, including Lisp, Scheme, Haskell, ML, and JavaScript.

¶:

Python's implementation of lexical scoping is described in PEP-0227.

```
i = 4
def foo(x: Iterable[int]):
    def bar():
        print(i, end='')
    # ...
    # Example
    # ...
    for i in x: # i is Foo's i, not bar's i
        print(i, end='')
    bar()
```

`foo([1, 2, 3])` prints `1 2 3 3`, `1 2 3 4`.

(Example: `x = range(4); for i in x: print(i); foo(i); bar()`.)

¶:

Example.

#### 4.3.16 Decorators

Tip: `staticmethod`, `classmethod`.

¶:

Decorators are functions that take a function as an argument and return a function. They are used to modify the behavior of a function. Example: `@property`, `@staticmethod`, `@classmethod`.

```
class C(object):
    @my_decorator
    def method(self):
        # ...
```

```
class C(object):
    def method(self):
        # ...
    method = my_decorator(method)
```

??:

????????????; ?????????????????, ????????? (invariant).

??:

????????????????????????????, ??????????????????. ??, ??????????????.  
 ?????(???????)????????????????. ?????????, ??????????.

??:

????????, ??????????. ??????????????????????. ???pydoc?????????????????. ??????????.

????????????????(????????, ???, ??????), ?????????(?????). pydoc ?????????????????  
 ??????????????. ?????????????, ????? (?????) ??????.

?????????????"????". ?????Python?????"???"???

???? staticmethod, ????????????? API ??????. ?????????????????.

???????????? classmethod: ?????????(named constructor); ????????????? (?????????????)?

#### 4.3.17 ??

Tip: ?????????????.

??Python????????????, ?????????????????(??Python?? \_\_hash\_\_ ? \_\_eq\_\_ ???). ??????????.

????????????, ????? queue ??? Queue ????. ????, ??? threading ?????????????(locking primitives). ???, ????????? threading.Condition ??????.

#### 4.3.18 ?????????

Tip: ?????.

??:

Python????????, ?????, ?????(metaclasses), ?????(bytecode), ???(on-the-fly compilation), ???, ?????(object reparenting), ??(import)??, ??(?? getattr()), ?????, \_\_del\_\_ ?????????.

??:

????????????.

[illegible]

```
???:
    ??????.
    ??????????????????????, ?? abc.ABCMeta, dataclasses ? enum.
```

#### 4.3.19 python: from \_\_future\_\_ imports

Tip: `?????__future__?`, `???????????????`, `???????????????`.

```

22:
23: from __future__ import unicode_literals, Python 2.

```

Table 1:  $\chi^2$  test,  $\chi^2/\text{d.o.f.}$ ,  $\chi^2/\text{d.o.f.}$ ,  $\chi^2/\text{d.o.f.}$  (regression).  $\chi^2/\text{d.o.f.}$ ,  $\chi^2/\text{d.o.f.}$ .

22: 2222222222222222, 2222222222222222 future 22. 2222222222222222222222222222.

```

[ ]: from __future__ imports

[ ] from __future__ import [ ]. [ ], [ ] Python [ ]. [ ],
[ ].

```

3.5 分, 3.7 分, 3.7 分:

```
from __future__ import generator_stop
```

Python future `__future__`.

????????????????, ?????? future ??. ?????? future ??, ??????, ?????????????????.

????????, ????????? from \_\_\_\_future\_\_\_\_???

4.3.20

Tip: `python3 -m pip install --user pytype`

□□□□□□□□□□□□,□□□□□ **pyi** □. □□□□□□□□□□. □□□□□□□□□□□□, □□□ **pyi** □□.

??:

????????????????:

```
def func(a: int) -> List[int]:
```

PEP-526 [PEP-526](#):

```
a: SomeType = some_func()
```

??:

?????????????????. ?????????????????????, ?????????????????.

??:

?????????????. ??????????????. ?????????????.

??:

???????????????? python ????. ?????????API?, ???????, ??????(build system)??? pytype.  
 ??python????????, ?????????(????????)?????????????????. ??????, ????? BUILD  
 ????????????? TODO ??????, ?????????????????.

(??? : ?????????IDE??vim????????????)

## 4.4 Python?????

### 4.4.1 ??

Tip: ???????, ?????????????????.

### 4.4.2 ??

Tip: ????? 80 ???.

- ??:
1. ???? (import) ??.
  2. ???? URL???????????? (flag).
  3. ?????????????????????????????, ?? URL ?????.
  4. Pylint ????.(?? : # pylint: disable=invalid-name)

?????????? ???? (explicit line continuation).

???? Python ? ???, ????????????? (implicit line joining) . ???? , ?????????????????.

??:

```
foo_bar(self, width, height, color='?', design=None, x='foo',
        emphasis=None, highlight=0)

if (width == 0 and height == 0 and
    color == '?' and emphasis == '??'):

(bridge_questions.clarification_on
 .averageairspeedof.unladen_swallow) = '?????????'

with (
    very_long_first_expression_function() as spam,
    very_long_second_expression_function() as beans,
    third_thing() as eggs,
```

(continues on next page)



(continued from previous page)

```
) :
    place_order(eggs, beans, spam, beans)
```

```
██:
```

```
if width == 0 and height == 0 and \
    color == '█' and emphasis == '██':

bridge_questions.clarification_on \
    .averageairspeed_of.unladen_swallow = '██████████?'

with very_long_first_expression_function() as spam, \
    very_long_second_expression_function() as beans, \
    third_thing() as eggs:
    place_order(eggs, beans, spam, beans)
```

```
██████████ (literal) ████, ██████████████████:
```

```
x = ('██████████████████████'
     '████████████████████')
```

```
██████████████████████. ████████████, ██████████████████.
```

```
██:
```

```
bridgekeeper.answer(
    name="██", quest=questlib.find(owner="██", perilous=True))

answer = (a_long_line().of_chained_methods()
          .that_eventually_provides().an_answer())

if (
    config is None
    or 'editor.language' not in config
    or config['editor.language'].use_spaces is False
):
    use_tabs()
```

```
██:
```

```
bridgekeeper.answer(name="██", quest=questlib.find(
    owner="██", perilous=True))

answer = a_long_line().of_chained_methods().that_eventually_provides(
    ).an_answer()

if (config is None or 'editor.language' not in config or config[
    'editor.language'].use_spaces is False):
    use_tabs()
```

```
███, █████ URL ██████.
```

```
██:
```

```
# ████
# http://www.example.com/us/developer/documentation/api/content/v2.0/csv_file_name_
↳ extension_full_specification.html
```

??:

```
# ????
# http://www.example.com/us/developer/documentation/api/content/\
# v2.0/csv_file_name_extension_full_specification.html
```

?????????????; ???? ?? ????.

?????? 80 ???, ? Black ? Pyink ??????????????, ??????? 80 ???, ?????????????????.

#### 4.4.3 ??

Tip: ?????????.

????? (tuple) ???, ????. ?????????????????, ?????????????.

??:

```
if foo:
    bar()
while x:
    x = bar()
if x and y:
    bar()
if not x:
    bar()
# ?????????, ??????.
onesie = (foo,)
return foo
return spam, beans
return (spam, beans)
for (x, y) in dict.items(): ...
```

??:

```
if (x):
    bar()
if not(x):
    bar()
return (foo)
```

#### 4.4.4 ??

Tip: ?4??????.

????????, ?????????????(?? ?? ????), ????4????????, ??? (???, ??????)  
 ?????????????????, ?????????????????????????.

??:

```

# 1.
foo = long_function_name(var_one, var_two,
                          var_three, var_four)
meal = (spam,
        beans)

# 2.
foo = {
    'long_dictionary_key': value1 +
                           value2,
    ...
}

# 3. 4. 5. 6. 7. 8. 9. 10.
foo = long_function_name(
    var_one, var_two, var_three,
    var_four)
meal = (
    spam,
    beans)

# 11. 12. 13. 14. 15. 16. 17. 18. 19. 20.
foo = long_function_name(
    var_one, var_two, var_three,
    var_four
)
meal = (
    spam,
    beans,
)

# 21. 22. 23. 24. 25. 26. 27. 28. 29. 30.
foo = {
    'long_dictionary_key':
        long_dictionary_value,
    ...
}

```

21:

```

# 1.
foo = long_function_name(var_one, var_two,
                          var_three, var_four)

# 2.
foo = long_function_name(
    var_one, var_two, var_three,
    var_four)

# 3.
foo = {
    'long_dictionary_key':
        long_dictionary_value,
    ...
}

```





```

keys: Sequence[bytes | str],
require_all_keys: bool = False,
) -> Mapping[bytes, tuple[str, ...]]:
    """Smalltable Smalltable.

    table_handle Table Smalltable. Smalltable,
    Smalltable UTF-8 Smalltable.

    Smalltable:
        table_handle: Smalltable Smalltable.
        keys: Smalltable, Smalltable. Smalltable UTF-8 Smalltable.
        require_all_keys: Smalltable True, Smalltable
            Smalltable.

    Smalltable:
        Smalltable, Smalltable. Smalltable. Smalltable:

        {b'Serak': ('Rigel VII', 'Preparer'),
          b'Zim': ('Irk', 'Invader'),
          b'Lrrr': ('Omicron Persei 8', 'Emperor')}

        Smalltable. Smalltable keys Smalltable, Smalltable
        Smalltable (Smalltable require_all_keys Smalltable false).

    Smalltable:
        IOError: Smalltable Smalltable.

    """

```

```
def fetch_smalltable_rows(
    table_handle: smalltable.Table,
    keys: Sequence[bytes | str],
    require_all_keys: bool = False,
) -> Mapping[bytes, tuple[str, ...]]:
    """Fetch rows from a smalltable.

    Parameters:
        table_handle: A smalltable.Table object.
        keys: A sequence of keys to fetch.
        require_all_keys: If True, only return rows that contain all keys.

    Returns:
        A mapping from bytes to a tuple of strings.
    """
    table_handle.fetch_rows(keys, require_all_keys=False)

    # Fetch rows from the table
    rows = table_handle.fetch_rows(keys, require_all_keys=False)

    # Convert the rows to a mapping
    mapping = {}
    for row in rows:
        key = row[0]
        value = row[1:]
        mapping[key] = value

    return mapping
```

#### 4.4. Python



```
# [0][0]: [0][0] b, [0][0] i [0], [0][0] i+1
```

Tip: [REDACTED]. [REDACTED].



```
employee_table = '<table>'
for last_name, first_name in employee_list:
    employee_table += '<tr><td>%s, %s</td></tr>' % (last_name, first_name)
employee_table += '</table>'
```

~~~~~. ~~~ ' ~~~ " ~~~~~~. ~~~~~~. ~~~~~~.

~~:

```
Python('~~~~~?')
Gollum("I'm scared of lint errors. (~~~~~)")
Narrator("'~~!' ~~~~ Python ~~~~~.")
```

(~~~~: ~~~ "I'm" ~~~~~~.)

~~:

```
Python("~~~~~?")
Gollum('~~~~~. ~~~~. ~~~~~~.')
Gollum("~~~~~. ~~~. ~~~.")
```

~~~~~ """ ~~~ ' '. ~~~~~~ ' ~~~~~~. ~~~~~~ ' '. ~~~~~, ~~~~~~  
""".

~~~~~. ~~~~~~. ~~~~~~. ~~~ textwrap.dedent() ~~~~~~.

~~:

```
long_string = """~~~~~.
~~~~~.
"""
```

~~:

```
long_string = """~~~~~.
~~~~~. """

long_string = ("~~~~~.\\n" +
               "~~~~~.")

long_string = ("~~~~~.\\n"
               "~~~~~.")
```

```
import textwrap

long_string = textwrap.dedent("""\
~~~~~, ~~~ textwrap.dedent()
~~~~~. """)
```

~~, ~~~~~~ ~~~~~~. ~~~, ~~~~~~ (literal) ~ ~~~~~~.

~~

~~~~~ (~~ % ~~~) ~~~~~: ~~~~~~ (~~ f-string!) ~~~~~~.  
~~~~~. ~~~~~~. ~~~~~~. ~~~~~~.

~~~~

```
import tensorflow as tf
logger = tf.get_logger()
logger.info('TensorFlow %s: %s', tf.__version__)
```

```
import os
from absl import logging

logging.info('%s $PAGER %: %s', os.getenv('PAGER', default=''))

homedir = os.getenv('HOME')
if homedir is None or not os.access(homedir, os.W_OK):
    logging.error('%%%%%%%%, $HOME=%r', homedir)
```

%%:

```
import os
from absl import logging

logging.info('%s $PAGER %:')
logging.info(os.getenv('PAGER', default=''))

homedir = os.getenv('HOME')
if homedir is None or not os.access(homedir, os.W_OK):
    logging.error(f'%%%%%%%%, $HOME={homedir!r}')
```

%%%%

%%%% (%%: %% ValueError %%%%%%%%%%) %%%%%%%%%%:

1. %%%%%%%%%%.
2. %%%%%%%%%%.
3. %%%%%%%%%% (%%%%%%%%, %% grepping).

%%:

```
if not 0 <= p <= 1:
    raise ValueError(f'%%%%%%%%: {p!r}')

try:
    os.rmdir(workdir)
except OSError as error:
    logging.warning('%%%%%%%% (%%: %r): %r',
                    error, workdir)
```

%%:

```
if p < 0 or p > 1: # %%: %% float('nan') %%%!
    raise ValueError(f'%%%%%%%%: {p!r}')

try:
    os.rmdir(workdir)
except OSError:
    # %%: %%%%%%%%%%
    # %%%%%%%%%%, %%%%%%%%%%.
    logging.warning('%%%%%%%%: %s', workdir)

try:
```

(continues on next page)



#### 4.4.11 TODO (??) ??

Tip: ?????????????????????? TODO (??) ??.

????? TODO (??) ??????????, ?????????????????????? (??? bug ??, ?????????). ?????? TODO (https://crbug.com/<bug??>) : ??? bug ??, ?? bug ?????????, ??????????????????????. TODO ?????????????.

?? TODO ??????????????????. TODO ??????????????????????. ??, ???????????? TODO ?, ?????????????????????.

```
# TODO(crbug.com/192795): ?? cpufreq ???.  
# TODO(?????): ?????? (issue), ? '*' ?????.
```

????? TODO ??????“?????”, ?????????????????? (“2009?11????”) ?????????? (“???????????? XML ???, ??????”), ?????????????????.

#### 4.4.12 ?? (import) ???????

Tip: ??????????????. typing ? collections.abc ??????. ??:

?:

```
from collections.abc import Mapping, Sequence  
import os  
import sys  
from typing import Any, NewType
```

?:

```
import os, sys
```

?????????????, ??. ??????????????, ??????:

1. ?? Python ? \_\_future\_\_. ??:

```
from __future__ import annotations
```

?????? \_\_future\_\_ ?????.

2. ?? Python ??????. ??:

```
import sys
```

3. ?? ??? ??????. ??:

```
import tensorflow as tf
```

4. ??????????????. ??:

```
from otherproject.ai import mind
```

5. ??????: ??????????????????????????????. ??:

```
from myproject.backend.hgwells import time_machine
```

Python , .  
.

( from path import ... path) , . .

```
import collections
import queue
import sys

from absl import app
from absl import flags
import bs4
import cryptography
import tensorflow as tf

from book.genres import scifi
from myproject.backend import huxley
from myproject.backend.hgwells import time_machine
from myproject.backend.state_machine import main_loop
from otherproject.ai import body
from otherproject.ai import mind
from otherproject.ai import soul

# :
#from myproject.backend.hgwells import time_machine
#from myproject.backend.state_machine import main_loop
```

#### 4.4.13

Tip: .

, , . try / except, try except .  
if else .

:

```
if foo: bar(foo)
```

:

```
if foo: bar(foo)
else:   baz(foo)

try:    bar(foo)
except ValueError: baz(foo)

try:
    bar(foo)
except ValueError: baz(foo)
```



5. `CapWords` (a `CapWords`), `lower_with_under` (a `lower_with_under.py`). `CapWords.py` `lower_with_under`, `lower_with_under` ("a, `lower_with_under` import StringIO a from StringIO import StringIO").
6. `lower_with_under` `PEP 8`, `lower_with_under`, `lower_with_under` `test_<lower_with_under>_<lower_with_under>`. `lower_with_under` `CapWords` `lower_with_under`, `lower_with_under`, `lower_with_under` `test lower_with_under`, `lower_with_under`. `lower_with_under` `test<lower_with_under>_<lower_with_under>`.

`lower_with_under`

`lower_with_under` Python `lower_with_under` .py `lower_with_under` (-). `lower_with_under` `lower_with_under`, `lower_with_under` (symbolic link) `lower_with_under` `exec "$0.py" "$@"` `lower_with_under` `bash` `lower_with_under`.

`lower_with_under`Python`lower_with_under`Guido`lower_with_under`

Table 1: `lower_with_under`

| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code>                                   |
|-------------------------------|-------------------------------|---|
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code>                                   |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code>                                   |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code>                                   |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code>                                   |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code>                                   |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code>                                   |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code> ( <code>lower_with_under</code> ) |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code> ( <code>lower_with_under</code> ) |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code>                                   |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code>                                   |

Table 2: `lower_with_under`

| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code> |
|-------------------------------|-------------------------------|-------------------------------|
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code> |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code> |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code> |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code> |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code> |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code> |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code> |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code> |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code> |
| <code>lower_with_under</code> | <code>lower_with_under</code> | <code>lower_with_under</code> |

`lower_with_under`

`lower_with_under`, `lower_with_under`, `lower_with_under`. `lower_with_under`, `lower_with_under`, `lower_with_under`. `lower_with_under` API `lower_with_under` `PEP8` `lower_with_under`, `lower_with_under` API `lower_with_under`.





5. `MyClass`.

1. `MyClass` API.
2. `MyClass`, `MyClass`, `MyClass`.
3. `MyClass` (`MyClass`).
4. `MyClass`.
5. `MyClass`. `MyClass`.

`def``MyClass`.`MyClass`, `MyClass` (signature) `MyClass`. `MyClass`, `MyClass`.

```
def my_method(
    self,
    first_var: int,
    second_var: Foo,
    third_var: Bar | None,
) -> int:
    ...
```

`MyClass`, `MyClass`. `MyClass`, `MyClass`.

```
def my_method(self, first_var: int) -> int:
    ...
```

`MyClass`, `MyClass`. `MyClass`, `MyClass`, `MyClass` def `MyClass`.`MyClass`:

```
def my_method(
    self,
    other_arg: MyLongType | None,
) -> tuple[MyLongType1, MyLongType1]:
    ...
```

`MyClass`.`MyClass`:

```
def my_method(
    self,
    first_var: int,
    second_var: int) -> dict[OtherLongType, MyLongType]:
    ...
```

`pylint` `MyClass`, `MyClass`, `MyClass`.`MyClass`:

```
def my_method(self,
    other_arg: MyLongType | None,
) -> dict[OtherLongType, MyLongType]:
    ...
```

`MyClass`, `MyClass`. `MyClass`. `MyClass`.

```
def my_method(
    self,
    first_var: tuple[list[MyLongType1],
                     list[MyLongType2]],
    second_var: list[dict[
        MyLongType3, MyLongType4]],
) -> None:
    ...
```

Annotations, `alias` annotation. Annotations.

Example:

```
def my_function(
    long_variable_name:
        long_module_name.LongTypeName,
) -> None:
    ...
```

Example:

```
def my_function(
    long_variable_name: long_module_name.
        LongTypeName,
) -> None:
    ...
```

Annotations (forward declaration)

Annotations (Annotations), `from __future__ import annotations`.

Example:

```
from __future__ import annotations

class MyClass:
    def __init__(self, stack: Sequence[MyClass], item: OtherClass) -> None:

class OtherClass:
    ...
```

```
class MyClass:
    def __init__(self, stack: Sequence['MyClass'], item: 'OtherClass') ->
    None:

class OtherClass:
    ...
```

Annotations

PEP-008, Annotations, Annotations.

Example:

```
def func(a: int = 0) -> int:
    ...
```

Example:

```
def func(a:int=0) -> int:
    ...
```

## NoneType

Python 3.10, NoneType 的 “??” 的。NoneType, None 的 NoneType 的。NoneType, NoneType, NoneType! 的。| 的 (union) 的 (Python 3.10+ 的) 的 Optional 的 Union 的。

的 X | None 的。的 PEP 484 的 a: str = None 的 a: str | None = None, 的。

的:

```
# 的。
def modern_or_union(a: str | int | None, b: str | None = None) -> str:
    ...
# 的 Union / Optional.
def union_optional(a: Union[str, int, None], b: Optional[str] = None) -> str:
    ...
```

的:

```
# 的 Union 的 Optional.
def nullable_union(a: Union[None, str]) -> str:
    ...
# 的 Optional.
def implicit_optional(a: str = None) -> str:
    ...
```

## 的 (alias)

的。的 (的 CapWorted)。的, 的 \_ 的 (的 \_Private)。

的 : TypeAlias 的 3.10 的。

```
from typing import TypeAlias

_LossAndGradient: TypeAlias = tuple[tf.Tensor, tf.Tensor]
ComplexTFMap: TypeAlias = Mapping[str, _LossAndGradient]
```

## 的

的 # type: ignore 的。

pytype 的 (的):

```
# pytype: disable=attribute-error
```

## 的

的

的, 的: 的, 的。

```
a: Foo = SomeUndecoratedFunction()
```

的

的 (Python 3.6 的), 的 # type: <的> 的:

```
a = SomeUndecoratedFunction() # type: Foo
```

??????

?????????????????????. ??. ??.

(??? : ?????????????????????,??python?,list?,tuple????????????????????????????????????,??,?????list?,tuple?????????????????)

```
a: list[int] = [1, 2, 3]
b: tuple[int, ...] = (1, 2, 3)
c: tuple[int, str, float] = (1, "2", 3.5)
```

????? (type variable)

Python ?????????? ?? (generics). ?????????????????????????????????, ?? TypeVar ? ParamSpec.

??:

```
from collections.abc import Callable
from typing import ParamSpec, TypeVar
_P = ParamSpec("_P")
_T = TypeVar("_T")
...
def next(l: list[_T]) -> _T:
    return l.pop()

def print_when_called(f: Callable[_P, _T]) -> Callable[_P, _T]:
    def inner(*args: _P.args, **kwargs: _P.kwargs) -> _T:
        print('?????')
        return f(*args, **kwargs)
    return inner
```

TypeVar ?????????.

```
AddableType = TypeVar("AddableType", int, float, str)
def add(a: AddableType, b: AddableType) -> AddableType:
    return a + b
```

AnyStr ? typing ??????????????????????. ?????????????????? bytes ? str ??????????????????.

```
from typing import AnyStr
def check_length(x: AnyStr) -> AnyStr:
    if len(x) <= 42:
        return x
    raise ValueError()
```

(??? : ??????, x ?????????????????? bytes ??????? str.)

????????????????????, ?????????????????:

1. ??????
2. ???????

??:

```
_T = TypeVar("_T")
_P = ParamSpec("_P")
AddableType = TypeVar("AddableType", int, float, str)
AnyFunction = TypeVar("AnyFunction", bound=Callable)
```

??:

```
T = TypeVar("T")
P = ParamSpec("P")
_T = TypeVar("_T", int, float, str)
_F = TypeVar("_F", bound=Callable)
```

?????

?????????? typing.Text. ??????????? Python 2/3 ?????.

? str ?????/?????. ? bytes ???????.

```
# ?????
def deals_with_text_data(x: str) -> str:
    ...
# ?????
def deals_with_binary_data(x: bytes) -> bytes:
    ...
```

????????????????, ?????????????????, ??? AnyStr.

????

???????????????? typing ? collections.abc ?????, ??????. ?????????, ????????? typing ? collections.abc ?????????, ??:

```
from collections.abc import Mapping, Sequence
from typing import Any, Generic
```

????, ?????????, ??? typing ? collections.abc ????????? (keyword) ????. ?????????, ?????????. ?????????, ??? import x as y ?????:

```
from typing import Any as AnyType
```

???, ??????. ?? Python 3.9 ??? PEP-585, ?????????.

```
def generate_foo_scores(foo: set[str]) -> list[float]:
    ...
```

?: Apache Beam ????????? typing ?????.

```
from typing import Set, List

# ????? Apache Beam ??? PEP 585 ???, ???
# ??? Python 3.9 ?????, ?????.
def generate_foo_scores(foo: Set[str]) -> List[float]:
    ...
```

?????

????, ?????????, ??????. ??????. ?????, ?????????.

???????????????? if TYPE\_CHECKING: ???.

1. ???, ?????????, ??? Python 3.6 ?????. ?? Python 3.6 ?????.
2. ?????????, ??????. ?????, ?????.

- 

Any ., (Any Any).  
.

?? (generics)

??:

??:

??:

??:

129

## 4.5 ?????

??????.

????, ??????????????????. ?????????????????????, ??????????. ?????????????????????, ?????????????????????.

?????????????????????. ?????????“??”, ??????“??”. ?????????????????, ??????????. ?????????????????????, ??????????. ????

## SHELL 环境变量 - 环境变量

## Contents

- Shell 环境变量 - 环境变量

## 5.1 环境变量

1.26

Paul Armstrong

1.26

Bean Zhang v1.26

- Google Style Guide
- Google 环境变量 - 环境变量

## 5.2 环境变量

## 5.2.1 环境变量Shell

Tip: Bash 环境变量shell 环境变量

```
#!/bin/bash  set  shell bash  <script_name>
```

```
shell bash shell
```

```
Solaris SVR4 Bourne shell
```



## 5.2.2 Shell

Tip: Shell

Shell

- shell
- shell
- \${PHPESTATUS} Python
- 100PythonShell

## 5.3 Shell

### 5.3.1

Tip: .sh.sh

shell.sh

### 5.3.2 SUID / SGID

Tip: SUID(Set User ID)SGID(Set Group ID)shell

shellSUID/SGIDshellbashSUID  
sudo

## 5.4

### 5.4.1 STDOUT vs STDERR

Tip: STDERR

```
err() {
    echo "[$(date +%Y-%m-%dT%H:%M:%S%z)]: $@" >&2
}

if ! do_something; then
    err "Unable to do_something"
    exit "${E_DID_NOTHING}"
fi
```

## 5.5

### 5.5.1

Tip:

```
#!/bin/bash
#
# Perform hot backups of Oracle databases.
```

```
#!/bin/bash
```

```
#!/bin/bash
#
# Perform hot backups of Oracle databases.
```

### 5.5.2

Tip:

```
#!/bin/bash
#
# Perform hot backups of Oracle databases.
```

```
#!/bin/bash
```

- 
- 
- 
- 

```
#!/bin/bash
```

```
#!/bin/bash
#
# Perform hot backups of Oracle databases.

export PATH='/usr/xpg4/bin:/usr/bin:/opt/csw/bin:/opt/goog/bin'

#####
# Cleanup files from the backup dir
# Globals:
#   BACKUP_DIR
```

(continues on next page)

(continued from previous page)

```
# ORACLE_SID
# Arguments:
#   None
# Returns:
#   None
#####
cleanup() {
    ...
}
```

### 5.5.3 ????????

Tip:

[illegible]

### 5.5.4 TODO???

Tip: `??TODO????????????????????????????`

??C++????????

[illegible]

```
# TODO(mrmonkey): Handle the unlikely edge cases (bug ####)
```

## 5.6

### 5.6.1

Tip: 

[illegible]

## 5.6.2

Tip: 80

here document

```
# DO use 'here document's
cat <<END;
I am an exceptionally long
string.
END

# Embedded newlines are ok too
long_string="I am an exceptionally
long string."
```

## 5.6.3

Tip:

```
# All fits on one line
command1 | command2

# Long commands
command1 \
| command2 \
| command3 \
| command4
```

## 5.6.4

Tip: ; do,; then while,for,if

shell ; do,; then if/for/while else

```
for dir in ${dirs_to_cleanup}; do
  if [[ -d "${dir}/${ORACLE_SID}" ]]; then
    log_date "Cleaning up old files in ${dir}/${ORACLE_SID}"
    rm "${dir}/${ORACLE_SID}/*"
    if [[ "$?" -ne 0 ]]; then
      error_message
    fi
  fi
```

(continues on next page)

(continued from previous page)

```

else
    mkdir -p "${dir}/${ORACLE_SID}"
    if [[ "$?" -ne 0 ]]; then
        error_message
    fi
fi
done

```

### 5.6.5 case

Tip:

- 
- 
- 

```

case in esac ;& ;&

```

```

case "${expression}" in
a)
    variable="..."
    some_command "${variable}" "${other_expr}" ...
    ;;
absolute)
    actions="relative"
    another_command "${actions}" "${other_expr}" ...
    ;;
*)
    error "Unexpected expression '${expression}'"
    ;;
esac

```

```

;; ;& ;&

```

```

verbose='false'
aflag=''
bflag=''
files=''
while getopts 'abf:v' flag; do
    case "${flag}" in
        a) aflag='true' ;;
        b) bflag='true' ;;
        f) files="${OPTARG}" ;;
        v) verbose='true' ;;
        *) error "Unexpected option ${flag}" ;;
    esac
done

```

## 5.6.6

Tip: `{var} $var`

- 1.
- 2.
3. shell

```
# Section of recommended cases.

# Preferred style for 'special' variables:
echo "Positional: $1" "$5" "$3"
echo "Specials: !=$, -=$, _=$_. ?=$?, #=$# *= $* @=$@ \=$$ ..."

# Braces necessary:
echo "many parameters: ${10}"

# Braces avoiding confusion:
# Output is "a0b0c0"
set -- a b c
echo "${1}0${2}0${3}0"

# Preferred style for other variables:
echo "PATH=${PATH}, PWD=${PWD}, mine=${some_var}"
while read f; do
  echo "file=${f}"
done < <(ls -l /tmp)

# Section of discouraged cases

# Unquoted vars, unbraced vars, brace-quoted single letter
# shell specials.
echo a=$avar "b=$bvar" "PID=${$}" "${1}"

# Confusing use: this is expanded as "${1}0${2}0${3}0",
# not "${10}${20}${30}"
set -- a b c
echo "$10$20$30"
```

## 5.6.7

Tip:

- shell
- 
- 
- [ [

- `set -e` `set -o errexit` `set -o errtrace` `set -o nounset`

```
# 'Single' quotes indicate that no substitution is desired.
# "Double" quotes indicate that substitution is required/tolerated.

# Simple examples
# "quote command substitutions"
flag="$(some_command and its args "$@" 'quoted separately')"
```

```
# "quote variables"
echo "${flag}"

# "never quote literal integers"
value=32
# "quote command substitutions", even when you expect integers
number="$(generate_number)"

# "prefer quoting words", not compulsory
readonly USE_INTEGER='true'

# "quote shell meta characters"
echo 'Hello stranger, and well met. Earn lots of $$$'
echo "Process $$: Done making \${$}."
```

```
# "command options or path names"
# ($1 is assumed to contain a value here)
grep -li Hugo /dev/null "$1"
```

```
# Less simple examples
# "quote variables, unless proven false": ccs might be empty
git send-email --to "${reviewers}" ${ccs:+"--cc" "${ccs}"}

# Positional parameter precautions: $1 might be unset
# Single quotes leave regex as-is.
grep -cP '([Ss]pecial|\\|?characters*)$' ${1:+"$1"}
```

```
# For passing on arguments,
# "$@" is right almost everytime, and
# $* is wrong almost everytime:
#
# * $* and $@ will split on spaces, clobbering up arguments
#   that contain spaces and dropping empty strings;
# * "$@" will retain arguments as-is, so no args
#   provided will result in no args being passed on;
#   This is in most cases what you want to use for passing
#   on arguments.
# * "$*" expands to one argument, with all args joined
#   by (usually) spaces,
#   so no args provided will result in one empty string
#   being passed on.
# (Consult 'man bash' for the nit-grits ;-)
```

```
set -- 1 "2 two" "3 three tres"; echo $# ; set -- "$*"; echo "$#, $@"
set -- 1 "2 two" "3 three tres"; echo $# ; set -- "$@"; echo "$#, $@"
```

## 5.7 `test`

### 5.7.1 `test`

---

Tip: `test $(command) ...`

---

```
test $(command) ...
test
```

```
# This is preferred:
var="$(command "$(command1) ") "

# This is not:
var="\`command \`${command1}\`"
```

### 5.7.2 `test`

---

Tip: `test [[ ... ]]` `test [,test,/usr/bin/`

---

```
test [[ ]] ... test [,test,/usr/bin/ ...
]
```

```
# This ensures the string on the left is made up of characters in the
# alnum character class followed by the string name.
# Note that the RHS should not be quoted here.
# For the gory details, see
# E14 at http://tiswww.case.edu/php/chet/bash/FAQ
if [[ "filename" =~ ^[:alnum:]+name ]]; then
    echo "Match"
fi

# This matches the exact pattern "f*" (Does not match in this case)
if [[ "filename" == "f*" ]]; then
    echo "Match"
fi

# This gives a "too many arguments" error as f* is expanded to the
# contents of the current directory
if [ "filename" == f* ]; then
    echo "Match"
fi
```



### 5.7.3

Tip:

Bash

```
# Do this:
if [[ "${my_var}" = "some_string" ]]; then
    do_something
fi

# -z (string length is zero) and -n (string length is not zero) are
# preferred over testing for an empty string
if [[ -z "${my_var}" ]]; then
    do_something
fi

# This is OK (ensure quotes on the empty side), but not preferred:
if [[ "${my_var}" = "" ]]; then
    do_something
fi

# Not this:
if [[ "${my_var}X" = "some_stringX" ]]; then
    do_something
fi
```

`-z` `-n`

```
# Use this
if [[ -n "${my_var}" ]]; then
    do_something
fi

# Instead of this as errors can occur if ${my_var} expands to a test
# flag
if [[ "${my_var}" ]]; then
    do_something
fi
```

### 5.7.4

Tip:

`rm -rf somedir`

```
# Here's the contents of the directory:
# -f -r somedir somefile

# This deletes almost everything in the directory by force
psa@bilby$ rm -v *
removed directory: `somedir'
```

(continues on next page)

(continued from previous page)

```
removed `somefile'

# As opposed to:
psa@bilby$ rm -v ./\*
removed `./-f'
removed `./-r'
rm: cannot remove `./somedir': Is a directory
removed `./somefile'
```

### 5.7.5 Eval

Tip: `eval`

`eval`

```
# What does this set?
# Did it succeed? In part or whole?
eval $(set_my_variables)

# What happens if one of the returned values has a space in it?
variable="$(eval some_function)"
```

### 5.7.6 `while`

Tip: `for` `while` `while` `shell` `shell`

`while` `shell` `bug`

```
last_line=NULL
your_command | while read line; do
    last_line="${line}"
done

# This will output 'NULL'
echo "${last_line}"
```

`for`

```
total=0
# Only do this if there are no spaces in return values.
for value in $(command); do
    total+="${value}"
done
```

`shell` `bash` `while` `shell`

```
total=0
last_file=
while read count filename; do
```

(continues on next page)

(continued from previous page)

```

total+="${count}"
last_file="${filename}"
done < <(your_command | uniq -c)

# This will output the second field of the last line of output from
# the command.
echo "Total = ${total}"
echo "Last one = ${last_file}"

```

```

shellwhile" "awkshell

```

```

# Trivial implementation of awk expression:
#   awk '$3 == "nfs" { print $2 " maps to " $1 }' /proc/mounts
cat /proc/mounts | while read src dest type opts rest; do
    if [[ ${type} == "nfs" ]]; then
        echo "NFS ${dest} maps to ${src}"
    fi
done

```

## 5.8

### 5.8.1

Tip: `function` function

`Google`

```

# Single function
my_func() {
    ...
}

# Part of a package
mypackage::my_func() {
    ...
}

```

`()` function

### 5.8.2

Tip: `for`

```


```

```

for zone in ${zones}; do
    something_with "${zone}"
done

```



### 5.8.6 ☐☐☐☐☐☐

Tip: `local [REDACTED]`

[illegible]

```
my_func2() {
    local name="$1"

    # Separate lines for declaration and assignment:
    local my_var
    my_var="$ (my_func) " || return

    # DO NOT do this: $? contains the exit code of 'local', not my_func
    local my_var="$ (my_func) "
    [[ $? -eq 0 ]] || return

    ...
}
```

### 5.8.7 ???? ?

Tip:

```

#####includes#####set#####

```

### 5.8.8 `main`

Tip: `main`

```

main
main

```

```
main "$@"
```

```

main

```

## 5.9 `if`

### 5.9.1 `if`

Tip: `if` is a shell builtin

```
if [ $? = 0 ]; then
    # do something
fi
```

```
if ! mv "${file_list}" "${dest_dir}/" ; then
    echo "Unable to move ${file_list} to ${dest_dir}" >&2
    exit "${E_BAD_MOVE}"
fi

# Or
mv "${file_list}" "${dest_dir}/"
if [[ "$?" -ne 0 ]]; then
    echo "Unable to move ${file_list} to ${dest_dir}" >&2
    exit "${E_BAD_MOVE}"
fi
```

Bash `PIPESTATUS` variable

```
tar -cf - ./ * | ( cd "${dir}" && tar -xf - )
if [[ "${PIPESTATUS[0]}" -ne 0 || "${PIPESTATUS[1]}" -ne 0 ]]; then
    echo "Unable to tar files to ${dir}" >&2
fi
```

`PIPESTATUS` variable

```
tar -cf - ./ * | ( cd "${DIR}" && tar -xf - )
return_codes=("${PIPESTATUS[*]}")
if [[ "${return_codes[0]}" -ne 0 ]]; then
    do_something
fi
if [[ "${return_codes[1]}" -ne 0 ]]; then
    do_something_else
fi
```

### 5.9.2 `if`

Tip: `if` is a shell builtin

```
bash(1) sed
```

```
# Prefer this:
addition=$(( ${X} + ${Y} ))
substitution="${string/#foo/bar}"
```

(continues on next page)

(continued from previous page)

```
# Instead of this:
addition="$(expr ${X} + ${Y})"
substitution="$(echo "${string}" | sed -e 's/^foo/bar/')"
```

## 5.10

## JAVASCRIPT 6.0.0 - 6.0.0

## 6.1 6.0.0

Google JavaScript JavaScript JavaScript

## 6.2 Javascript 6.0.0

## 6.2.1 var 6.0.0

```
var
```

```
var
var
```

## 6.2.2 6.0.0

- NAMES\_LIKE\_THIS
- @const
- IE const

```
CONSTANT_VALUE_CASE
```

```
number string boolean
```



□□□□□□□□□□

```

2          @const      ?????????????????????const????????????????????const
?????????????IE????????????const?????
????????????? @const  ??????

```

??

```
??? @const ?????????? CONSTANT_VALUE_CASE ? ?? ?????
```

```
/* *
 * ████████████████████
 * @type {number}
 */
goog.example.TIMEOUT_IN_MILLISECONDS = 60;
```

[illegible]

```
/**
 * Map of URL to response string.
 * @const
 */
MyClass.fetchedUrlCache_ = new goog.structs.Map();
```

[illegible]

### 6.2.3

□□□□□□□□

□□□

□□□□□□□□□□□□□□

```
// 1.
MyClass.prototype.myMethod = function() {
    return 42;
} // ??????.

(function() {
    // ?????????????????
})();

var x = {
    'i': 1,
    'j': 2
} //?????.

// 2.  ???IE?firefox?????????.
//?????????????????.
[normalVersion, ffVersion][isIE]();

var THINGS_TO_EAT = [apples, oysters, sprayOnCheese] //????????
```

(continues on next page)

(continued from previous page)

```
// 3.
-1 == resultOfOperation() || die();
```

~~~~~

1. js42~~~~~42~~~~~
2. ~~~~“no sush property in undefined”~~~~ x[normalVersion, ffVer-  
sion][isIE]() ~~~~
3. die ~~~~ resultOfOperation() ~ NaN ~~~~ THINGS\_TO\_EAT ~~~~ die() ~~~~

~~~~~

```
js~~~~~)"}"}]~~~~~  
~~~~~
```

~~~~~

~~~~~

```
var foo = function() {  
  return true;  
}; // ~~~~  
  
function foo() {  
  return true;  
} // ~~~~~
```

## 6.2.4 ~~~~

~~~~~

~~~~~

## 6.2.5 ~~~~~

~~~~~

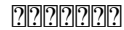
~~~~~

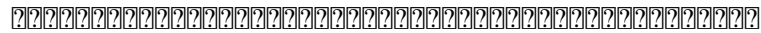
```
if (x) {  
  function foo() {}  
}
```

~~~~~ECMAScript~~~~~ ECMA-262 ~13~14~~~~~EcmaScript~~~~~ECM  
~~~~~

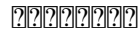
```
if (x) {  
  var foo = function() {}  
}
```

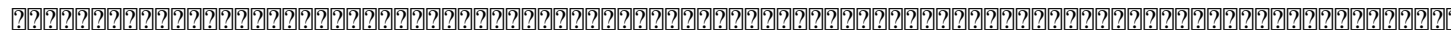
## 6.2.6



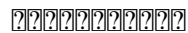


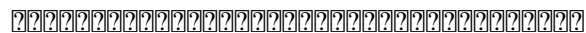
## 6.2.7





## 6.2.8

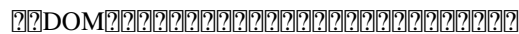




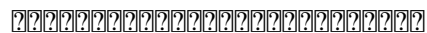
string.charAt(3)





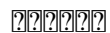
string[3]

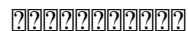



## 6.2.9

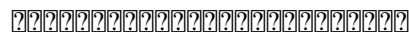


```
var x = new Boolean(false);
if (x) {
    alert('hi'); //“hi”
}
```



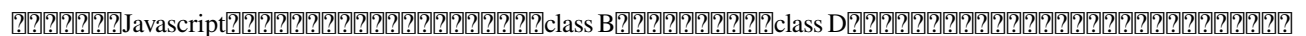





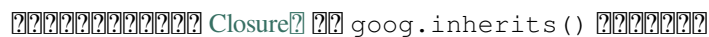


```
var x = Boolean(0);
if (x) {
    alert('hi'); //
}
typeof Boolean(0) == 'boolean';
typeof new Boolean(0) == 'object';
```



## 6.2.10



Javascriptclass Bclass D

Closure goog.inherits() 

```
function D() {
    goog.base(this)
}
goog.inherits( D, B );
```

(continues on next page)

(continued from previous page)

```
D.prototype.method =function() {
    ...
};
```

## 6.2.11

```
/** */ function SomeConstructor() { this.someProperty = 1; } Foo.prototype.
someMethod = function() { ... };
```

“new”

```
Foo.prototype.bar = function() {
    /* ... */
};
```

```
/** @constructor */
function Foo() {
    this.bar = value;
}
```

JavaScript “”

## 6.2.12

this.foo = null

```
o.prototype.dispose = function() {
    this.property_ = null;
};
```

```
Foo.prototype.dispose = function() {
    delete this.property_;
};
```

JavaScript if (key in obj)

6.2.13 ☐ ☐

????????????

JS

DOM

```
function foo(element, a, b) {
  element.onclick = function() { /* ☐ a ☐ b */ };
}
```

abcdefghijklmnopqrstuvwxyz

```
function foo(element, a, b) {
    element.onclick = bar(a, b);
}

function bar(a, b) {
    return function() { /* ☐ a ☐ b */ }
}
```

### 6.2.14 eval()

?????RPC????

```
eval() eval() eval() eval() eval() eval() eval() eval()
```

[illegible]

```
users = [
  {
    name: 'Eric',
    id: 37824,
    email: 'jellyvore@myway.com'
  },
  {
    name: 'xtof',
    id: 31337,
    email: 'b4d455h4x0r@google.com'
  },
  ...
];
```

??

```
eval() XMLHttpRequest XMLHttpRequest JavaScript
```

```
var userOnline = false;
var user = 'nusrat';
var xmlhttp = new XMLHttpRequest();
xmlhttp.open('GET', 'http://chat.google.com/isUserOnline?user=' + user, false);
xmlhttp.send('');
// 0000000
// userOnline = true;
if (xmlhttp.status == 200) {
    eval(xmlhttp.responseText);
}
```

(continues on next page)

(continued from previous page)

```

}
// userOnline ??? true

```

## 6.2.15 with() {}

??????

?? with ?????????? with ???

```

with (foo) {
    var x = 3;
    return x;
}

```

???????????????? x ??? foo ??????????????????setter????????????????3???????????????????? with ?

## 6.2.16 this

????????????????????????????????

this ??? eval ???DOM????????????HTML????????????????????????????????????  
call() ? apply() ???????

??? this ?????????????????????????????????

- ???????
- ?????????????????????

## 6.2.17 for-in ??

????????????????????????

for-in ????????????????????????????? 0 ? length-1 ???

```

function printArray(arr) {
    for (var key in arr) {
        print(arr[key]);
    }
}

printArray([0,1,2,3]); //????

var a = new Array(10);
printArray(a); //????

a = document.getElementsByTagName('*');
printArray(a); //????

a = [0,1,2,3];
a.buhu = 'wine';
printArray(a); //????

a = new Array;

```

(continues on next page)

(continued from previous page)

```
a[3] = 3;
printArray(a); //
```

```


```

```
function printArray(arr) {
    var l = arr.length;
    for (var i = 0; i < l; i++) {
        print(arr[i]);
    }
}
```

## 6.2.18

```


```

```
.....
```

```
JS Date RegExp String
```

## 6.2.19

```


```

```


```

```
var myString = 'A rather long string of English text, an error message \
    actually that just keeps going and going -- an error \
    message to make the Energizer bunny blush (right through \
    those Schwarzenegger shades)! Where was I? Oh yes, \
    you\'ve got an error and all the extraneous whitespace is \
    just gravy.  Have a nice day.';
```

```
ECMAScript
```

```


```

```
var myString = 'A rather long string of English text, an error message ' +
    'actually that just keeps going and going -- an error ' +
    'message to make the Energizer bunny blush (right through ' +
    'those Schwarzenegger shades)! Where was I? Oh yes, ' +
    'you\'ve got an error and all the extraneous whitespace is ' +
    'just gravy.  Have a nice day.';
```

## 6.2.20

~~~~~

~~~~~

~~~~~

```
// 3
var a1 = new Array(x1, x2, x3);

// 2
var a2 = new Array(x1, x2);

// If x1 is a number and it is a natural number the length will be x1.
// If x1 is a number but not a natural number this will throw an exception.
// Otherwise the array will have one element with x1 as its value.
var a3 = new Array(x1);

// 0
var a4 = new Array();
```

~~~~~

~~~~~

```
var a = [x1, x2, x3];
var a2 = [x1, x2];
var a3 = [x1];
var a4 = [];
```

~~~~~

```
var o = new Object();

var o2 = new Object();
o2.a = 0;
o2.b = 1;
o2.c = 2;
o2['strange key'] = 3;
```

~~~~~

```
var o = {};

var o2 = {
  a: 0,
  b: 1,
  c: 2,
  'strange key': 3
};
```





## getter/setter

EcmaScript 5 getter/setter getter

```
/**
 *--.
 */
var foo = { get next() { return this.nextId++; } };
};
```

getter/setter getFoo() setFoo(value)  
isFoo()

JavaScript

JavaScript

“Project Sloth” sloth.\*

```
var sloth = {};

sloth.sleep = function() {
  ...
};
```

JavaScript the Closure Library Dojo toolkit

```
goog.provide('sloth');

sloth.sleep = function() {
  ...
};
```

sloths hats Sloth sloth.  
hats

Example 6-1: foo.hats module

```

// foo.hats module
foo.hats = {
  // ...
  BowlerHat: function() {
    // ...
  }
};

```

```

foo.require('foo.hats');
/**
 * @constructor
 * @extends {foo.hats.RoundHat}
 */
foo.hats.BowlerHat = function() {
  // ...
};

```

Example 6-2: googleyhats module

```

foo.provide('googleyhats.BowlerHat');

foo.require('foo.hats');
/**
 * @constructor
 * @extends {foo.hats.RoundHat}
 */
googleyhats.BowlerHat = function() {
  // ...
};
goog.exportSymbol('foo.hats.BowlerHat', googleyhats.BowlerHat);

```

Example 6-3: myapp module

Example 6-4: goog.scope

```

/**
 * @constructor
 */
some.long.namespace.MyClass = function() {
  // ...
};

/**
 * @param {some.long.namespace.MyClass} a
 */
some.long.namespace.MyClass.staticHelper = function(a) {
  // ...
};

myapp.main = function() {
  var MyClass = some.long.namespace.MyClass;
  var staticHelper = some.long.namespace.MyClass.staticHelper;
  staticHelper(new MyClass());
};

```

Example 6-5: goog.scope

```

myapp.main = function() {
  var namespace = some.long.namespace;
  // ...
};

```

(continues on next page)

(continued from previous page)

```
namespace.MyClass.staticHelper(new namespace.MyClass());
};
```

```
/** @enum {string} */
some.long.namespace.Fruit = {
  APPLE: 'a',
  BANANA: 'b'
};

myapp.main = function() {
  var Fruit = some.long.namespace.Fruit;
  switch (fruit) {
    case Fruit.APPLE:
      ...
    case Fruit.BANANA:
      ...
  }
};
```

```
myapp.main = function() {
  var MyClass = some.long.namespace.MyClass;
  MyClass.staticHelper(null);
};
```

.js - \_ \_ -

## 6.3.2 toString()

toString()   
 toString() toString()

## 6.3.3

### 6.3.4

???

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX window XXXXXXXXXXXXXXXXXXXXXXXXXXXX window XXX window
XX

```

### 6.3.5 ????

**QUESTION** C++ **ANSWER**

???

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
if (something) {
    // ...
} else {
    // ...
}
```

□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
var arr = [1, 2, 3]; // [[1], [2], [3]]
var obj = {a: 1, b: 2, c: 3}; // [{"a": 1}, {"b": 2}, {"c": 3}]
```

[illegible]

```
//[REDACTED]
var inset = {
  top: 10,
  right: 20,
  bottom: 15,
  left: 12
};

//[REDACTED]
this.rows_ = [
  "Slartibartfast" <fjordmaster@magrathea.com>',
  "Zaphod Beeblebrox" <theprez@universe.gov>',
  "Ford Prefect" <ford@theguide.com>',
  "Arthur Dent" <has.no.tea@gmail.com>',
  "Marvin the Paranoid Android" <marv@googlemail.com>',
  'the.mice@magrathea.com'
];

//[REDACTED]
goog.dom.createDom(goog.dom.TagName.DIV, {
  id: 'foo',
  className: 'some-css-class',
  style: 'display:none'
}, 'Hello, world!');
```





(continued from previous page)

```

var dom = goog.dom;

// Alias new types after the constructor declaration.
my.module.SomeType = function() { ... };
var SomeType = my.module.SomeType;

// Declare methods on the prototype as usual:
SomeType.prototype.findButton = function() {
    // Button as aliased above.
    this.button = new Button(dom.getElementById('my-button'));
};
...
}); // goog.scope

```

## Example

Example of a function that takes a large number of arguments and returns a large number of values.

```

someWonderfulHtml = ' ' +
    getEvenMoreHtml(someReallyInterestingValues, moreValues,
                    evenMoreParams, 'a duck', true, 72,
                    slightlyMoreMonkeys(0xffff)) +
    ' ';

thisIsAVeryLongVariableName =
    hereIsAnEvenLongerOtherFunctionNameThatWillNotFitOnPrevLine();

thisIsAVeryLongVariableName = 'expressionPartOne' + someMethodThatIsLong() +
    thisIsAnEvenLongerOtherFunctionNameThatCannotBeIndentedMore();

someValue = this.foo(
    shortArg,
    'Some really long string arg - this is a pretty common case, actually.',
    shorty2,
    this.bar());

if (searchableCollection(allYourStuff).contains(theStuffYouWant) &&
    !ambientNotification.isActive() && (client.isAmbientSupported() ||
    client.alwaysTryAmbientAnyways())) {
    ambientNotification.activate();
}

```

## Example

Example of a function that takes a large number of arguments and returns a large number of values.

```

doSomethingTo(x);
doSomethingElseTo(x);
andThen(x);

nowDoSomethingWith(y);

andNowWith(z);

```



## 6.3.5 Ternary Operator

JavaScript has a ternary operator that can be used to write conditional expressions.

```

var x = a ? b : c; // All on one line if it will fit.

// Indentation +4 is OK.
var y = a ?
    longButSimpleOperandB : longButSimpleOperandC;

// Indenting to the line position of the first operand is also OK.
var z = a ?
    moreComplicatedB :
    moreComplicatedC;

```

## 6.3.6 Conditional Execution

```

var x = foo.bar().
    doSomething().
    doSomethingElse();

```

## 6.3.6 Switch Statement

JavaScript has a switch statement.

The switch statement is used to execute one of many possible code blocks.

```

switch (expression) {
    delete type of void return throw case in new
}

```

## 6.3.7 String Escaping

JavaScript has a way to escape special characters in strings.

```

var msg = "This is some HTML";

```

```

var msg = 'This is some HTML';

```

## 6.3.8 Annotations

```

@private @protected JSDoc

```

```

JSDoc @private @protected

```

```

--jscomp_warning=visibility

```

```

@private

```

```

@private @private @private
instanceof

```

```

@protected

```

```
// 1
// AA_PrivateClass_ 2 AA_init_ 3
//
/**
 * @private
 * @constructor
 */
AA_PrivateClass_ = function() {
};

/** @private */
function AA_init_() {
    return new AA_PrivateClass_();
}

AA_init_();
```

2 @private 3

2 @protected 3

3C++4JAVA5private 6protected 7C++8

```
// File 1.

/** @constructor */
AA_PublicClass = function() {
    /** @private */
    this.privateProp_ = 2;

    /** @protected */
    this.protectedProp = 4;
};

/** @private */
AA_PublicClass.staticPrivateProp_ = 1;

/** @protected */
AA_PublicClass.staticProtectedProp = 31;

/** @private */
AA_PublicClass.prototype.privateMethod_ = function() {};

/** @protected */
AA_PublicClass.prototype.protectedMethod = function() {};

// File 2.

/**
 * @return {number} The number of ducks we've arranged in a row.
 */
AA_PublicClass.prototype.method = function() {
    // Legal accesses of these two properties.
    return this.privateProp_ + AA_PublicClass.staticPrivateProp_;
};

// File 3.
```

(continues on next page)

(continued from previous page)

```

/**
 * @constructor
 * @extends {AA_PublicClass}
 */
AA_SubClass = function() {
  // Legal access of a protected static property.
  AA_PublicClass.staticProtectedProp = this.method();
};
goog.inherits(AA_SubClass, AA_PublicClass);

/**
 * @return {number} The number of ducks we've arranged in a row.
 */
AA_SubClass.prototype.method = function() {
  // Legal access of a protected instance property.
  return this.protectedProp;
};

```

JavaScript AA\_PrivateClass\_ public private

### 6.3.9 JavaScript

JSDoc EcmaScript 4

JavaScript

ES4 JavaScript JsDoc

ES4

| ???    | ??                        | ??                                | ???             |
|--------|---------------------------|-----------------------------------|-----------------|
| ???    | JavaScript2015????        | ????                              |                 |
|        | {null} ? {undefined}      |                                   |                 |
|        | ? {boolean} ? {number} ?? |                                   |                 |
|        | {string}                  |                                   |                 |
| ???    | {Object} ?????            | ???????????????????? @constructor |                 |
|        | {Function} ?????          | JSDoc???? @interface JS-          |                 |
|        | {EventTarget}             | Doc????                           |                 |
|        | ????EventTar-             | ????????????????????              | ???             |
|        | get????null????           |                                   |                 |
| ???    | {goog.events.             | ????????????????????,?? @enum     |                 |
|        | EventType}                | JSDoc?????? ?? ?????              |                 |
|        | ???? goog.                | ????????????????ES4????           |                 |
|        | events.EventType ?        |                                   |                 |
| ???    | {Array.<string>} ?????    | ????????????????Java???           |                 |
|        | {Object.                  |                                   |                 |
|        | <string, number>}         |                                   |                 |
|        | ????                      |                                   |                 |
| ???    | { (number boolean) }      | ????A??B??                        | { (number,      |
|        | ????                      | ????????????                      | boolean) }      |
|        |                           | {number boolean}                  | ?               |
|        |                           | {function(): (number boolean) }   | { (num-         |
|        |                           |                                   | ber  boolean) } |
| ?????  | {?number}                 | ????????????????syntactic sugar?? | {number?        |
|        | ????                      |                                   | }               |
| ???    | {!Object}                 | ????null????                      | {Object!        |
|        | ????                      |                                   | }               |
| ???    | { {myNum: number, myOb-   | ???? myNum ? number ???           |                 |
|        | ject} }                   | myObject ?????                    |                 |
|        | ????                      | ???????????? length               |                 |
|        |                           | ???? Array.<{length}> ?           |                 |
|        |                           | ????                              |                 |
| ???    | {function(string,         |                                   |                 |
|        | boolean)}                 |                                   |                 |
|        | ????                      |                                   |                 |
| ?????  | {function(): number}      | ????                              |                 |
|        | ????                      |                                   |                 |
| ??     | {function(this:goog.      | ????                              |                 |
| this   | ui.Menu, string)}         |                                   |                 |
| ??     | ????                      |                                   |                 |
|        | goog.ui.Menu              |                                   |                 |
| ??     | {function(new:goog.ui.    | ????                              |                 |
| new    | Menu, string)}            |                                   |                 |
| ??     | ????                      |                                   |                 |
|        | goog.ui.Menu ???          |                                   |                 |
| ???    | {function(string, ...     | ????                              |                 |
|        | [number]): number}        |                                   |                 |
|        | ????                      |                                   |                 |
| ?????  | @param {...number}        | ????                              |                 |
| @param | var_args                  |                                   |                 |
| ???    | ????                      |                                   |                 |
| ??     | {function(?string=,       | ????                              |                 |
| ????   | number=)}                 |                                   |                 |
|        | ????                      |                                   |                 |
| ??     | @param {number=}          | ????                              |                 |



# JavaScript

| Primitive      | Constructor | Example                                       |
|----------------|-------------|-----------------------------------------------|
| number         |             | <pre>1 1.0 -5 1e5 Math.PI</pre>               |
| Number         | Number      | <pre>new Number(true)</pre>                   |
| string         |             | <pre>'Hello' "World" String(42)</pre>         |
| String         | String      | <pre>new String('Hello') new String(42)</pre> |
| boolean        | Boolean     | <pre>true false Boolean(0)</pre>              |
| Boolean        | Boolean     | <pre>new Boolean(true)</pre>                  |
| RegExp         |             | <pre>new RegExp('hello') /world/g</pre>       |
| Date           |             | <pre>new Date new Date()</pre>                |
| null           |             | <pre>null</pre>                               |
| undefined      |             | <pre>undefined</pre>                          |
| void           |             | <pre>function f() {   return; }</pre>         |
| Array          |             | <pre>['foo', 0.3, null] []</pre>              |
| Array.<number> |             | <pre>[11, 22, 33]</pre>                       |
| Array.<string> |             | <pre>['one', 'two', 'three']</pre>            |
| Object         |             | <pre>{   foo: 'bar' }</pre>                   |

???

????????????????????????????????????????????????????????????

```
/** @type {number} */ (x)
```

??????????????

JavaScript????????????????????????????????????????????????

????????????????????

```
/**
 * ?????
 * @param {Object} value
 * @constructor
 */
function MyClass(value) {
  /**
   * Some value.
   * @type {Object}
   * @private
   */
  this.myValue_ = value;
}
```

myValue\_ null myValue\_ null, :

```
/**
 * null
 * @param {!Object} value
 * @constructor
 */
function MyClass(value) {
  /**
   * Some value.
   * @type {!Object}
   * @private
   */
  this.myValue_ = value;
}
```

MyClass null

undefined

```
/**
 * 
 * @param {Object=} opt_value
 * @constructor
 */
function MyClass(opt_value) {
  /**
   * Some value.
   * @type {Object|undefined}
   * @private
   */
```

(continues on next page)

(continued from previous page)

```

    this.myValue_ = opt_value;
}

```

```

myValue_ null undefined

```

```

:  opt_value {Object=} {Object|undefined} unde-
finedundefined

```

```


```

```

/**
 * 
 * @param {!Object} nonNull null
 * @param {Object} maybeNull null
 * @param {!Object=} opt_nonNull null
 * @param {Object=} opt_maybeNull null
 */
function strangeButTrue(nonNull, maybeNull, opt_nonNull, opt_maybeNull) {
    // ...
};

```

```


```

```


```

```

/**
 * @param {string} tagName
 * @param {(string|Element|Text|Array.<Element>|Array.<Text>)} contents
 * @return {!Element}
 */
goog.createElement = function(tagName, contents) {
    ...
};

```

```

@typedef

```

```

/** @typedef {(string|Element|Text|Array.<Element>|Array.<Text>)} */
goog.ElementContent;

/**
 * @param {string} tagName
 * @param {goog.ElementContent} contents
 * @return {!Element}
 */
goog.createElement = function(tagName, contents) {
    ...
};

```





```
/**
 * This is NOT the preferred indentation method.
 * @param {string} foo This is a param with a description too long to fit in
 * one line.
 * @return {number} This returns something that has a description too long to
 * fit in one line.
 */
project.MyClass.prototype.method = function(foo) {
  return 5;
};
```

## JSDocHTML

JavaDoc, JSDoc HTML  `<pre> <tt> <strong> <ul> <ol> <li> <a>`

JSDoc

```
/**
 * Computes weight based on three factors:
 * items sent
 * items received
 * last timestamp
 */
```

Computes weight based on three factors: items sent items received items received last.  
timestamp

```
/**
 * Computes weight based on three factors:
 * <ul>
 * <li>items sent
 * <li>items received
 * <li>last timestamp
 * </ul>
 */
```

JavaDoc doc

```
/**
 * @fileoverview Description of file, its uses and information
 * about its dependencies.
 */
```

## Class

Example 1: A class that makes something fun and easy.

```
/**
 * Class making something fun and easy.
 * @param {string} arg1 An argument that makes this more interesting.
 * @param {Array.<number>} arg2 List of numbers to be processed.
 * @constructor
 * @extends {goog.Disposable}
 */
project.MyClass = function(arg1, arg2) {
  // ...
};
goog.inherits(project.MyClass, goog.Disposable);
```

## Example 2

Example 3: A class that makes something fun and easy.

```
/**
 * Operates on an instance of MyClass and returns something.
 * @param {project.MyClass} obj Instance of MyClass which leads to a long
 *   comment that needs to be wrapped to two lines.
 * @return {boolean} Whether something occurred.
 */
function PR_someMethod(obj) {
  // ...
}
```

## Example 4

```
/** @constructor */
project.MyClass = function() {
  /**
   * Maximum number of things per pane.
   * @type {number}
   */
  this.someProperty = 4;
}
```

## JSDoc

| @author | @author username@google.com (first last)<br><pre> /**  * @fileoverview  * Utilities for  * handling textareas.  * @author kuth@google.  * com (Uthur Pendragon)  */ </pre>                                                                                                                                                                                                    | @fileoverview |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| @code   | <pre> { @code ... } </pre> <pre> /** </pre> <ul style="list-style-type: none"> <li>• Moves to the next position in the selection.</li> <li>• Throws { @code goog.iter.StopIteration when it</li> <li>• passes the end of the range.</li> <li>• @return {Node} The node at the next position.</li> </ul> <pre> */ goog.dom.RangeIterator.prc = function() {   // ... }; </pre> |               |

continues on next page

Table 1 – continued from previous page

| ??           | ?????                                                                                                                                                                                                                                                                                                                                                                                              | ??                                                                                                                                                                                                                                                                                                                |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| @const       | @const @const {type}<br>???<br><pre> /** @const */ var MY_ ↪BEER = 'stout'; /**  * My namespace's ↪favorite kind of ↪beer.  * @const {string}  */ myspace.MY_BEER = ↪'stout';  /** @const */ MyClass. ↪MY_BEER = 'stout';  /**  * Initializes the ↪request.  * @const  */ myspace.Request. ↪prototype.initialize ↪= function() {     // This method ↪cannot be overridden ↪in a subclass. } </pre> | ??????????????????<br>??? @const ?????????????????????????????????????????js????????<br>?????????????????????????????????????????????????????????<br>?????????????????????????????????????????????????????????<br>@const<br>?????????????????????????????????????????????????????????<br>@const ??????????? ?? ?? |
| @constructor | @constructor<br>???<br><pre> /**  * A rectangle.  * @constructor  */ function GM_Rect() {     ... } </pre>                                                                                                                                                                                                                                                                                         | ?????????????????????                                                                                                                                                                                                                                                                                             |
| @define      | @define {Type} description<br>???<br><pre> /** @define {boolean} ↪*/ var TR_FLAGS_ENABLE_ ↪DEBUG = true;  /** @define {boolean} ↪*/ goog.userAgent.ASSUME_ ↪IE = false; </pre>                                                                                                                                                                                                                     | ?????????????????????<br>????????????????????? --define='goog.<br>userAgent.ASSUME_IE=true'<br>????????????????????? goog.userAgent.ASSUME_IE<br>????????? true ?                                                                                                                                                 |

continues on next page

Table 1 – continued from previous page

| ??          | ??????                                                                                                                                                                                                                                                                                                                                         | ??                                                                                       |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| @deprecated | @deprecated Description<br>???<br><pre> /**  * Determines whether a  * ↪node is a field.  * @return {boolean}  * ↪True if the contents  * ↪of  *   the element are  * ↪editable, but the  * ↪element  *   itself is not.  * @deprecated Use  * ↪isField().  */ BN_EditUtil.   ↪isTopEditableField =   ↪function(node) {     // ...   }; </pre> | ????????????????????????????????????????????                                             |
| @dict       | @dict Description<br>???<br><pre> /**  * @constructor  * @dict  */ function Foo(x) {   this['x'] = x; } var obj = new Foo(123); var num = obj.x; //   ↪warning   (** @dict \*/ { x: 1 }   ↪).x = 123; //   ↪warning </pre>                                                                                                                     | ????? (????Foo)???? @dict ?????????????????<br>Foo ????????????????????????????????????? |
| @enum       | @enum {Type}<br>???<br><pre> /**  * Enum for tri-state  * ↪values.  * @enum {number}  */ project.TriState = {   TRUE: 1,   FALSE: -1,   MAYBE: 0 }; </pre>                                                                                                                                                                                     |                                                                                          |

continues on next page

Table 1 – continued from previous page

| ??       | ?????                                                                                                                                                                                                                  | ??                                                                                                                                                                                                                                                                     |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| @export  | @export<br><pre> /** @export */ foo.MyPublicClass.   ↳prototype.   ↳myPublicMethod =   ↳function() {     // ...   }; </pre>                                                                                            | <pre> --generate_exports goog.exportSymbol('foo.MyPublicClass.   ↳prototype.myPublicMethod',     foo.MyPublicClass.prototype.   ↳myPublicMethod); </pre> <p>1. ?? //javascript/closure/base.js ,<br/>??</p> <p>2. ??? goog.exportSymbol ? goog.<br/>exportProperty</p> |
| @expose  | @expose<br><pre> /** @expose */ My- Class.prototype.exposedProperty = 3; </pre>                                                                                                                                        | <pre> @expose </pre>                                                                                                                                                                                                                                                   |
| @extends | @extends    Type    @extends<br>{Type}<br><pre> /**  *  * • Immutable  *    empty    node  *    list.  * • @constructor  * • @extends  *    goog.ds.BasicNodeLis  *  */ goog.ds.EmptyNodeList = function() { }; </pre> | ? @constructor                                                                                                                                                                                                                                                         |
| @externs | @externs<br><pre> /**  *  * • @fileoverview  *    This is an  *    externs file.  * • @externs  *  */ var document; </pre>                                                                                             |                                                                                                                                                                                                                                                                        |

continues on next page

Table 1 – continued from previous page

| @fileoverview | <p>@fileoverview Description</p> <pre> /**  *  * @fileoverview  * Utilities for  * doing things  * that require  * this very long  *  * but not  * indented  * comment.  *  * @author  * kuth@google.com  * (Uthur Pen-  * dragon)  */ </pre>                             |                                               |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
| @implements   | <p>@implements Type @imple-</p> <pre> /**  *  * A shape.  * @interface  */ function Shape() {}; Shape.prototype.draw = func- tion() {}; /**  *  * @constructor  * @implements  * {Shape}  */ function Square() {}; Square.prototype.draw = func- tion() {   ... }; </pre> | <p>@constructor</p>                           |
| @inheritDoc   | <p>@inheritDoc</p> <pre> /** @inheritDoc */ project.SubClass.   ↳prototype.toString()   ↳{     // ...   }; </pre>                                                                                                                                                         | <p>@override</p> <p>@inheritDoc @override</p> |

continues on next page



Table 1 – continued from previous page

| ??                     | ?????                                                                                                                                                                                                                                                                                                    | ??                                                                                                                                                                                                                                 |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| @interface             | @interface<br>???<br><pre> /**  * A shape.  * @interface  */ function Shape() {}; Shape.prototype.draw =   ↪function() {};  /**  * A polygon.  * @interface  * @extends {Shape}  */ function Polygon() {}; Polygon.prototype.   ↪getSides =   ↪function() {}; </pre>                                     | ????????????????                                                                                                                                                                                                                   |
| @lends                 | @lends objectName @lends<br>{objectName}<br>???<br><pre> goog.object.extend(   Button.prototype,   /** @lends {Button.   ↪prototype} */ {     isButton:   ↪function() { return   ↪true; }   }); </pre>                                                                                                   | ?????????????????????????????????????????????????????????????<br>?????????????????????????????????????????????????????????????<br>@type {Foo} ???Foo???????? @lends {Foo}<br>???“Foo????”.<br>JSDoc Toolkit docs ????????????????? |
| @license or @pre-serve | @license Description<br>???<br><pre> /**  * @preserve Copyright   ↪2009 SomeThirdParty.  * Here is the full   ↪license text and   ↪copyright  * notice for this file.   ↪ Note that the   ↪notice can span   ↪several  * lines and is only   ↪terminated by the   ↪closing star and   ↪slash:  */ </pre> | ? @licenseor ? @preserve<br>?????????????????????????????<br>?????????????????????????????                                                                                                                                         |

continues on next page

Table 1 – continued from previous page

| @noalias       | <div> <div>@noalias</div> <div> <pre> /** @noalias */ function Range() {} </pre> </div> </div>                                                                                                                                                                                            |  |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| @nosideeffects | <div> <div>@nosideeffects</div> <div> <pre> /** @nosideeffects */ function ↳noSideEffectsFn1() {   // ... }; /** @nosideeffects */ var noSideEffectsFn2 = ↳function() {   // ... }; /** @nosideeffects */ a.prototype. ↳noSideEffectsFn3 = ↳function() {   // ... }; </pre> </div> </div> |  |
| @override      | <div> <div>@override</div> <div> <pre> /**  * @return {string} ↳Human-readable ↳representation of ↳project.SubClass.  * @override  */ project.SubClass. ↳prototype.toString() ↳{   // ... }; </pre> </div> </div>                                                                         |  |

continues on next page

Table 1 – continued from previous page

| ??         | ?????                                                                                                                                                                                                                                                                                                                                                                  | ??                                                                                                                                                                                                                                                                                                             |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| @param     | <p>@param {Type} varname De-<br/>scription<br/>???</p> <pre>/**  * Queries a Baz for  *   items.  * @param {number}  *   groupNum Subgroup id  *   to query.  * @param  *   {string number null}  *   term An itemName,  *   or itemId, or  *   null to search  *   everything.  */ goog.Baz.prototype.   query =   function(groupNum,   term) {     // ...   };</pre> | <p>????????????????????<br/>??? ??????????????????</p>                                                                                                                                                                                                                                                         |
| @private   | <p>@private @private {type}<br/>???</p> <pre>/**  * Handlers that are  *   listening to this  *   logger.  * @private {!Array.  *   &lt;Function&gt;}  */ this.handlers_ = [];</pre>                                                                                                                                                                                   | <p>???????????????????? ???<br/>@private ??????????????????</p>                                                                                                                                                                                                                                                |
| @protected | <p>@protected @protected {type}<br/>???</p> <pre>/**  * Sets the component's  *   root element to the  *   given element.  *   Considered  *   protected and final.  * @param {Element}  *   element Root element  *   for the component.  * @protected  */ goog.ui.Component.   prototype.   setElementInternal =   function(element) {     // ...   };</pre>         | <p>????????????? `????? &lt;<a href="http://google-styleguide.googlecode.com/svn/trunk/javascriptguide.xml#Visibility__private_and_protected_fields">http://google-styleguide.googlecode.com/svn/trunk/javascriptguide.xml#</a><br/>Visibility__private_and_protected_fields&gt;`<br/>????????????????????</p> |

continues on next page

Table 1 – continued from previous page

| @return | <div> <div>@return {Type} Description</div> <div> <pre> /**  * @return {string} The  * ↪hex ID of the last  * ↪item.  */ goog.Baz.prototype.   ↪getLastId =   ↪function() {     // ...     return id;   }; </pre> </div> </div>                                                                                                                                | <div> <div></div> <div>@return</div> <div></div> </div>                               |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| @see    | <div> <div>@see Link</div> <div> <pre> /**  * Adds a single item,  * ↪recklessly.  * @see #addSafely  * @see goog.Collect  * @see goog.  * ↪RecklessAdder#add  * ... </pre> </div> </div>                                                                                                                                                                      |                                                                                       |
| @struct | <div> <div>@struct Description</div> <div> <pre> /**  * @constructor  * @struct  */ function Foo(x) {   this.x = x; } var obj = new Foo(123); var num = obj['x']; //   ↪warning obj.y = "asdf"; //   ↪warning  Foo.prototype = /**   ↪@struct */ {   method1: function()   ↪{} }; Foo.prototype.method2   ↪= function() {}; //   ↪warning </pre> </div> </div> | <div> <div></div> <div>Foo</div> <div></div> <div>@struct</div> <div>Foo</div> </div> |

continues on next page

Table 1 – continued from previous page

| @supported | <p>@supported Description</p> <pre> /**  * @fileoverview Event_  * ↪Manager  * Provides an_  * ↪abstracted interface_  * ↪to the  * browsers' event_  * ↪systems.  * @supported So far_  * ↪tested in IE6 and_  * ↪FF1.5  */ </pre> |  |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| @suppress  | <p>@suppress {warning1 warning2}</p> <pre> /**  * @suppress  * ↪{deprecated}  */ function f() {   ↪   ↪deprecatedVersionOfF();   ↪ } </pre>                                                                                         |  |
| @template  | <p>@template</p> <pre> /**  * @param  * ↪{function(this:T, ...  * ↪)} fn  * @param {T} thisObj  * @param {...*} var_  * ↪args  * @template T  */ goog.bind =_ ↪function(fn, thisObj, ↪ var_args) {   ... }; </pre>                  |  |

continues on next page

Table 1 – continued from previous page

| @this    | <div> <div>@this Type @this {Type}</div> <div> <pre> pinto.chat.   ↪RosterWidget.extern(   ↪'getRosterElement',   /**    * Returns the roster_    ↪widget element.    * @this pinto.chat.    ↪RosterWidget    * @return {Element}    */   function() {     return this.     ↪getWrappedComponent_     ↪().getElement();   }); </pre> </div> </div> | <div> <div>this</div> <div></div> </div> |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------|
| @type    | <div> <div>@type Type @type {Type}</div> <div> <pre> /**   • The message     hex ID.   • @type     {string}    */ var hexId = hexId; </pre> </div> </div>                                                                                                                                                                                          |                                          |
| @typedef | <div> <div>@typedef</div> <div> <pre> /** @typedef   ↪{(string number)} */ goog.NumberLike; /** @param {goog.   ↪NumberLike} x A_   ↪number or a string._   ↪*/ goog.readNumber =_   ↪function(x) {     ...   } </pre> </div> </div>                                                                                                               |                                          |

JSDoc JSDoc Toolkit

- @augments
- @argument
- @borrows
- @class
- @constant
- @constructs

- @default
- @event
- @example
- @field
- @function
- @ignore
- @inner
- @link
- @memberOf
- @name
- @namespace
- @property
- @public
- @requires
- @returns
- @since
- @static
- @version

### 6.3.11 goog.provide

Example

goog.provide('namespace.MyClass');

Example

```
goog.provide('namespace.MyClass');
```

Example

```
goog.provide('namespace.MyClass');
goog.provide('namespace.MyClass.Enum');
goog.provide('namespace.MyClass.InnerClass');
goog.provide('namespace.MyClass.TypeDef');
goog.provide('namespace.MyClass.CONSTANT');
goog.provide('namespace.MyClass.staticMethod');
```

Example

```
goog.provide('foo.bar');
goog.provide('foo.bar.method');
goog.provide('foo.bar.CONSTANT');
```

### 6.3.12

JS Closure Compiler

### 6.3.13

JavaScript

TrueFalse

false

- null
- undefined
- “”
- 0

true

- “0”
- []
- {}

```
while (x != null) {
```

x0false

```
while (x) {
```

null

```
if (y != null && y != '') {
```

```
if (y) {
```

- Boolean(‘0’) == true ‘0’ != true
- 0 != null 0 == [] 0 == false
- Boolean(null) == false null != true null != false
- Boolean(undefined) == false undefined != true undefined != false
- Boolean([]) == true [] != true [] == false
- Boolean({}) == true {} != true {} != false



JavaScript 教程

JavaScript 教程

```
if (val != 0) {
    return foo();
} else {
    return bar();
}
```

JavaScript 教程

```
return val ? foo() : bar();
```

JavaScript 教程 HTML 教程

```
var html = '<input type="checkbox"' +
    (isChecked ? ' checked' : '') +
    (isEnabled ? '' : ' disabled') +
    ' name="foo">';
```

&& ||

JavaScript 教程, JavaScript 教程

“||” 运算符 ‘default’ 运算符

```
/** @param {*=} opt_win */
function foo(opt_win) {
    var win;
    if (opt_win) {
        win = opt_win;
    } else {
        win = window;
    }
    // ...
}
```

JavaScript 教程

```
/** @param {*=} opt_win */
function foo(opt_win) {
    var win = opt_win || window;
    // ...
}
```

“&&” 运算符

```
if (node) {
    if (node.kids) {
        if (node.kids[index]) {
            foo(node.kids[index]);
        }
    }
}
```

JavaScript 教程

```
if (node && node.kids && node.kids[index]) {
    foo(node.kids[index]);
}
```

??????

```
var kid = node && node.kids && node.kids[index];
if (kid) {
    foo(kid);
}
```

????????????????

```
node && node.kids && node.kids[index] && foo(node.kids[index]);
```

??????

????????????????????????????????????????????????????????????lengthO(n)lengthO(n^2)

```
var paragraphs = document.getElementsByTagName('p');
for (var i = 0; i < paragraphs.length; i++) {
    doSomething(paragraphs[i]);
}
```

??????

```
var paragraphs = document.getElementsByTagName('p');
for (var i = 0, paragraph; paragraph = paragraphs[i]; i++) {
    doSomething(paragraph);
}
```

????????????????(????????????false) ???

????????????????firstChild?nextSibling????

```
var parentNode = document.getElementById('foo');
for (var child = parentNode.firstChild; child; child = child.nextSibling) {
    doSomething(child);
}
```





搜索

TS TypeScript 类型

类型

- 2021 年 09 月 02 日 Frank Li 添加
- 2024 年 02 月 29 日 Frank Li 添加

## 7.2 类型

### 7.2.1 类型

类型

TS TypeScript 类型 ASCII 字符串 (字符串) [\] \w]+

| 类型                 | 类型  |
|--------------------|-----|
| 字符串 UpperCamelCase | 字符串 |
| 字符串 lowerCamelCase | 字符串 |
| 字符串 CONSTANT_CASE  | 字符串 |
| 字符串 #ident         | 字符串 |

类型

loadHttpRequest loadHTTPURL XMLHttpRequest

类型 \$

\$ 类型 Observable 类型

类型

Array<T> T UpperCamelCase

???

Closure testSuites xUnit \_ testX\_whenY\_doesZ()

\_

\_ \_

```
const [a, , b] = [1, 5, 10]; // a <- 1, b <- 10
```

???

lowerCamelCase snake\_case

```
import * as fooBar from './foo_bar';
```

- jQuery \$
- three.js THREE

??

CONSTANT\_CASE deep frozen

```
const UNIT_SUFFIXES = {
  'milliseconds': 'ms',
  'seconds': 's',
};
// UNIT_SUFFIXES
// 
// 
```

```
class Foo {
  private static readonly MY_SPECIAL_NUMBER = 5;

  bar() {
    return 2 * Foo.MY_SPECIAL_NUMBER;
  }
}
```

??

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

??

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX const
XXXXXXXXXXXX readonly XXXXXXXXXXXXXXXXXXXXXXX

```

```
const {Foo}= SomeType;
const CAPACITY = 5;

class Teapot {
    readonly BrewStateEnum = BrewStateEnum;
    readonly CAPACITY = CAPACITY;
}
```

□□□□

??????????

- [illegible]

□□□□□

[illegible][illegible]

### 7.2.2 7.2.2

UTF-8 44444

??? ASCII ???? Unicode ???? ∞ ???? Unicode ???? \u221e  
????

```
// ?????????????????????
const units = 'µs';

// ?????????????????????
const output = '\uffeff' + content; // ?????????Byte Order Mark?BOM?
```





```
/**
 * POST
 * @param amountLitres
 */
brew(amountLitres: number, logger: Logger) {
    // ...
}
```

## 

```


```

```
class Foo {
    constructor(private readonly bar: Bar) { }
}
```

```
Foo Bar bar
```

```
JS Doc @param
```

```
/** */
class ParamProps {
    /**
     * @param percolator
     * @param beans
     */
    constructor(
        private readonly percolator: Percolator,
        private readonly beans: CoffeeBean[]) {}
}
```

```
/** */
class OrdinaryClass {
    /** brew() */
    nextBean: CoffeeBean;

    constructor(initialBean: CoffeeBean) {
        this.nextBean = initialBean;
    }
}
```

## 

```


```

```
//
new Percolator().brew(/* amountLitres= */ 5);

// brew
new Percolator().brew({amountLitres: 5});
```

```
/** {@link CoffeeBrewer} */
export class Percolator implements CoffeeBrewer {
    /**
```

(continues on next page)

□□□□□□□□□□

```
// @ts-expect-error JSDoc @Component FooComponent
@Component({
  selector: 'foo',
  template: 'bar',
})
/** "bar" */
export class FooComponent {}
```

```
/** @? "bar" @??? */
@Component({
  selector: 'foo',
  template: 'bar',
})
export class FooComponent {}
```

### 7.3.1

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

- ```
class Foo {
    public bar = new Bar(); // ????????? public ???

    constructor(public readonly baz: Baz) {} // ??????readonly ????????? baz ???
→public ????????? public ???
}
```





~~~~~

~~~~~

```
// ~~~~~
class Foo {
    private readonly userList: string[];
    constructor() {
        this.userList = [];
    }
}
```

```
// ~~~~~
class Foo {
    private readonly userList: string[] = [];
}
```

~~~~~

~~~~~template AngularJS ~~~~~controller~~~~~ private  
~~~~~

~~~~~ public ~~~~~ protected ~~~~~Angular ~ Polymer ~~~~~ public ~  
AngularJS ~~~~~ protected ~

~~~~~ TypeScript ~~~~~ obj['foo'] ~~~~~

~~~~~

~~~~~private~~~~~

~ obj['foo'] ~~~~~ TypeScript ~~~~~  
~~~~~ ~~~~

~~~~~

~~~~~

```
class Foo {
    constructor(private readonly someService: SomeService) {}

    get someMember(): string {
        return this.someService.someVariable;
    }

    set someMember(newValue: string) {
        this.someService.someVariable = newValue;
    }
}
```

~~~~~internal ~ wrapped~~~~~  
public~~~~~ readonly~

```
class Foo {
    private wrappedBar = '';
    get bar() {
```

(continues on next page)

```
class Bar {  
    private barInternal = '';  
    // ?????????????????????????????????????????????????????????????? bar ?? public?  
    get bar() {  
        return this.barInternal;  
    }  
  
    set bar(value: string) {  
        this.barInternal = value;  
    }  
}
```

```

TypeScript String Boolean Number new
Boolean(false) true

```

```
// ???????  
const s = 'hello';  
const b = false;  
const n = 5;
```

**TypeScript** `Array()` `new`

(continues on next page)

(continued from previous page)

```
// 00 [0, 0, 0, 0, 0]
Array.from<number>({length: 5}).fill(0);
```

### 7.3.6 7.3.6

TypeScript String() Boolean() new NaN

```
const bool = Boolean(false);
const str = String(aNumber);
const bool2 = !!str;
const str2 = `result: ${bool2}`;
```

string

Number() NaN

Tip: Number('') Number(' ') Number('\t') 0 NaN Number('Infinity') Number('-Infinity') Infinity -Infinity

```
const aNumber = Number('123');
if (isNaN(aNumber)) throw new Error(...); // NaN
assertFinite(aNumber, ...); //
```

+ +

```
// 
const x = +y;
```

parseInt parseFloat 12 dwarves 12

```
const n = parseInt(someString, 10); // 
const f = parseFloat(someString); //
```

parseInt

```
if (!/^[a-zA-F0-9]+$/.test(someString)) throw new Error(...);
// 16 
// tslint:disable-next-line:ban
const n = parseInt(someString, 16); // parseInt
```

Number() Math.floor Math.trunc

```
let f = Number(someString);
if (isNaN(f)) handleError();
f = Math.floor(f);
```

if for while boolean

```
// 
const foo: MyInterface|null = ...;
if (!!foo) {...}
while (!!foo) {...}
```

```
// 
const foo: MyInterface|null = ...;
if (foo) {...}
while (foo) {...}
```

```
// 0 
if (arr.length > 0) {...}

// 
if (arr.length) {...}
```

### 7.3.7

const let const var

```
const foo = otherValue; // foo 
let bar = someValue;     // bar 
```

const let var JavaScript  
bug TypeScript var

```
// 
var foo = someValue;
```

### 7.3.8

new Error() Error() new

```
// 
throw new Error('Foo is not a valid bar.');
```

```
// 
throw Error('Foo is not a valid bar.');
```

### 7.3.9

for (... in ...) for (... in ...)

```
// 
for (const x in someObj) {
  // x someObj 
}
```

if for (... of Object.keys(...))



```
// 遍历
for (const x in someObj) {
    if (!someObj.hasOwnProperty(x)) continue;
    // 对 x 遍历 someObj 遍历
}
```

```
// 遍历
for (const x of Object.keys(someObj)) { // 遍历 for _of_ 遍历
    // 对 x 遍历 someObj 遍历
}
```

```
// 遍历
for (const [key, value] of Object.entries(someObj)) { // 遍历 for _of_ 遍历
    // 对 key 遍历 someObj 遍历
}
```

### 7.3.10 遍历

遍历 for (... in ...) 遍历 string 遍历

```
// 遍历
for (const x in someArray) {
    // 对 x 遍历 (string 遍历)
}
```

遍历 for (... of someArr) 遍历 for 遍历

```
// 遍历
for (const x of someArr) {
    // 对 x 遍历
}
```

```
// 遍历
for (let i = 0; i < someArr.length; i++) {
    // 遍历 for/of 遍历
    const x = someArr[i];
    // ...
}
```

```
// 遍历
for (const [i, x] of someArr.entries()) {
    // 遍历
}
```

遍历 Array.prototype.forEach 遍历 Set.prototype.forEach 遍历 Map.prototype.forEach 遍历

```
// 遍历
someArr.forEach((item, index) => {
    someFn(item, index);
});
```

遍历



## 7.3.12 for...of

```

// for...of

```

```

for (let i = 0; i < x; i++) {
    doSomethingWith(i);
    andSomeMore();
}
if (x) {
    doSomethingWithALongMethodName(x);
}

```

```

// for...of
if (x)
    x.doFoo();
for (let i = 0; i < x; i++)
    doSomethingWithALongMethodName(i);

```

```

// for...of if

```

```

// for...of
if (x) x.doFoo();

```

## 7.3.13 switch

```

// switch default

```

```

// switch
switch (x) {
    case Y:
        doSomethingElse();
        break;
    default:
        // ...
}

```

```

// switch case ...

```

```

// switch
switch (x) {
    case X:
        doSomething();
        // ...
    case Y:
        // ...
}

```

```

// switch

```

```

// switch
switch (x) {
    case X:
    case Y:
        doSomething();
}

```

(continues on next page)

(continued from previous page)

```

    break;
    default: // 
}

```

### 7.3.14

JavaScript `===` and `!==` operators. JavaScript `===` and `!==` operators.

```

// 
if (foo == 'bar' || baz != bam) {
    // 
}

```

```

// 
if (foo === 'bar' || baz !== bam) {
    // 
}

```

JavaScript `==` and `!=` operators. JavaScript `==` and `!=` operators.

```

// 
if (foo == null) {
    //  foo is null or undefined 
}

```

### 7.3.15

JavaScript `function` keyword. JavaScript `function` keyword.

JavaScript `const` keyword. JavaScript `const` keyword. TypeScript `const` keyword.

JavaScript `this` keyword. JavaScript `this` keyword.

```

// 
function foo() { ... }

```

```

// 
// 
foo = () => 3; // 
// 
const foo = function() { ... }

```

JavaScript `function` keyword. JavaScript `function` keyword. TypeScript `function` keyword.

JavaScript `function` keyword. JavaScript `function` keyword.

```

interface SearchFunction {
    (source: string, subString: string): boolean;
}

```

(continues on next page)

(continued from previous page)

```
const fooSearch: SearchFunction = (source, subString) => { ... };
```

### 7.3.16

#### 

ES6 function

```
// 
bar(() => { this.doSomething(); })
```

```
// 
bar(function() { ... })
```

this function this  
this

#### 

```
// 
function someFunction() {
  // 
  const receipts = books.map((b: Book) => {
    const receipt = payMoney(b.price);
    recordTransaction(receipt);
    return receipt;
  });

  // 
  const longThings = myValues.filter(v => v.length > 1000).map(v => String(v));

  function payMoney(amount: number) {
    // this
  }
}
```

```
// 
myPromise.then(v => console.log(v));
```

```
// 
myPromise.then(v => {
  console.log(v);
});

// 
const transformed = [1, 2, 3].map(v => {
  const intermediate = someComplicatedExpr(v);
```

(continues on next page)

```
const more = acrossManyLines(intermediate);
return worthWrapping(more);
});
```

[illegible]

```
// ?????????????????????
document.body.onclick = () => { document.body.textContent = 'hello'; };

// ?????????????????????
const setTextFn = (e: HTMLElement) => { e.textContent = 'hello'; };
document.body.onclick = setTextFn.bind(null, document.body);
```

```

##### this ##### this
#####
const handler = (x) => { this.listener(x); }; ##### const
handler = this.listener; handler(x); #####

```

```
// =====
class DelayHandler {
  constructor() {
    // ===== this =====
    // ===== this === DelayHandler =====
    setTimeout(this.patienceTracker, 5000);
  }
  private patienceTracker() {
    this.waitedPatiently = true;
  }
}
```

(continues on next page)








(continued from previous page)

```
// y [?] null[?].....
y!.bar();
```

[illegible]

□□□□□□

????? as ???

```
// 
const x = (<Foo>z).length;
const y = <Foo>z.length;
```

```
// ?????  
const x = (z as Foo).length;
```

□□□□□□□□□□

[illegible]

```
interface Foo {
  bar: number;
  baz?: string; // ??????????"bam"?????"baz"?
}

const foo = {
  bar: 123,
  bam: 'abc', // ?????????????????????????
} as Foo;

function func() {
  return {
    bar: 123,
    bam: 'abc', // ?????????????????????????
  } as Foo;
}
```

7.3.20

□□□□□□□□□□ ; □□□□□□□□□□

```
// ?????.ts
interface Foo {
  memberA: string;
  memberB: number;
}
```

????????????????, ?????

```
// ???????
interface Foo {
```

(continues on next page)

(continued from previous page)

```

    memberA: string,
    memberB: number,
}

```

```

// Example 1: Using a type alias

```

```

// Example 2: Using a type alias
type SomeTypeAlias = {
    memberA: string,
    memberB: number,
};

let someProperty: {memberC: string, memberD: number};

```

```

// Example 3: Using a type alias

```

```

// Example 4: Using a type alias

```

```

// Example 5: Using a type alias
// Example 6: Using a type alias
console.log(x['someField']);
console.log(x.someField);

```

```

// Example 7: Using a type alias

```

```

// Example 8: Using a type alias
declare interface ServerInfoJson {
    appVersion: string;
    user: UserJson;
}

const data = JSON.parse(serverResponse) as ServerInfoJson;
console.log(data.appVersion); // Example 9: Using a type alias

```

```

// Example 10: Using a type alias

```

```

// Example 11: Using a type alias
// Example 12: Using a type alias

```

```

// Example 13: Using a type alias
import {method1, method2} from 'utils';
class A {
    readonly utils = {method1, method2};
}

```

```

// Example 14: Using a type alias
import * as utils from 'utils';
class A {
    readonly utils = utils;
}

```

???

Web

### 7.3.21

enum TypeScript const enum JavaScript

### 7.3.22 debugger

debugger

```
//
function debugMe() {
    debugger;
}
```

### 7.3.23

@MyDecorator

- Angular @Component @NgModule
- Polymer @property

TC39 Bug

```
/** JSDoc */
@Component({...}) //
class MyComp {
    @Input() myField: string; // .....

    @Input()
    myOtherField: string; // .....
}
```

## 7.4

### 7.4.1

???

TypeScript . . root/path/to/file

./foo path/to/foo

../../../../ 日本語

```
import {Symbol1} from 'google3/path/from/root';
import {Symbol2} from '../parent/file';
import {Symbol3} from './sibling';
```

日本語

日本語 TypeScript 日本語 namespace 日本語 module

日本語 TypeScript 日本語 ES6 日本語 import {foo} from 'bar'

日本語 namespace Foo { ... } 日本語

日本語 require 日本語 import x = require('...'); 日本語 ES6 日本語

```
// 日本語
namespace Rocket {
  function launch() { ... }
}

// 日本語 <reference>
/// <reference path="...">

// 日本語 require()
import x = require('mydep');
```

Tip: TypeScript 日本語 module 日本語 module Foo { ... } 日本語 ES6 日本語

## 7.4.2 日本語

日本語

```
// Use named exports:
export class Foo { ... }
```

日本語

```
// 日本語
export default class Foo { ... }
```

日本語

```
// 日本語
import Foo from './bar'; // 日本語
import Bar from './bar'; // 日本語
```

日本語 foo.ts 日本語

```
// 日本語
const foo = 'blah';
export default foo;
```

bar.ts

```
// 
import {fizz} from './foo';
```

error TS2614: Module '"./foo"' has no exported member 'fizz'

```
// 
import fizz from './foo';
```

fizz === foo

```
// 
export default class Foo {
  static SOME_CONSTANT = ...
  static someHelpfulFunction() { ... }
  ...
}
```

Foo

```
// 
export const SOME_CONSTANT = ...
export function someHelpfulFunction()
export class Foo {
  // Foo 
}
```

TypeScript API

export let

```
// 
export let foo = 3;
// ES6 foo foo 
// TypeScript foo 
window.setTimeout(() => {
  foo = 4;
}, 1000 /* ms */);
```

```
// 
let foo = 3;
window.setTimeout(() => {
  foo = 4;
```

(continues on next page)

(continued from previous page)

```

}, 1000 /* ms */);
// 
export function getFoo() { return foo; };

```

```


```

```

function pickApi() {
  if (useOtherApi()) return OtherApi;
  return RegularApi;
}
export const SomeApi = pickApi();

```

```


```

```


```

```

// 
export class Container {
  static FOO = 1;
  static bar() { return 1; }
}

```

```


```

```

// 
export const FOO = 1;
export function bar() { return 1; }

```

### 7.4.3

ES6 TypeScript

| ES6                            | TypeScript | ES6 |
|--------------------------------|------------|-----|
| import * as foo from '...';    | TypeScript |     |
| import {Something} from '...'; | TypeScript |     |
| import Something from '...';   |            |     |
| import '...';                  |            |     |

```

// 
import * as ng from '@angular/core';
import {Foo} from './foo';

// 
import Button from 'Button';

// 
import 'jasmine';
import '@polymer/paper-button';

```

~~~~~

~~~~~

~~~~~ \* ~~~~~  
API ~~~~~

~~~~~ Jasmine describe  
it ~~~~~

```
// ~~~~~
import {TableViewItem, TableViewHolder, TableViewRow, TableViewModel,
TableViewRenderer} from './tableview';
let item: TableViewItem = ...;
```

```
// ~~~~~
import * as tableview from './tableview';
let item: tableview.Item = ...;
```

```
import * as testing from './testing';
```

```
// ~~~~~
// ~~~~~
// ~~~~~
testing.describe('foo', () => {
testing.it('bar', () => {
    testing.expect(...);
    testing.expect(...);
});
});
```

```
// ~~~~~
import {describe, it, expect} from './testing';

describe('foo', () => {
it('bar', () => {
    expect(...);
    expect(...);
});
});
...

```

~~~~~

~~~~~ import {Something as  
SomeOtherThing} ~~~

~~~~~

1. ~~~~~
2. ~~~~~
3. ~~~~~ RxJS from ~~~~~ observableFrom ~~~

```
import type export type
```

```
import type ... from export type ... from
```

Tip: export type Foo = ...;

```
// 
import type {Foo} from './foo';
export type {Bar} from './bar';
```

```
// 
import {Foo} from './foo';
export {Bar} from './bar';
```

TypeScript

```
import type import import type
import '...'
export type API import type
export type API
UserService AjaxUserService
```

## 7.4.4

```
products checkout backend views models
controllers
```

## 7.5

### 7.5.1

```
TypeScript google3
any
```

```
const x = 15; // x
```

```
string number boolean RegExp new
```

```
// boolean
const x: boolean = true;
```

```
// Set
const x: Set<string> = new Set();
```

```
// TypeScript
const x = new Set<string>();
```



[illegible]

□□□□

[illegible][illegible]

- [illegible]

### 7.5.2 Null ☐ Undefined ☐

```
TypeScript [?] null [?] undefined [?] string | null [?] undefined [?]  
null [?] undefined [?]
```

```
TypeScript undefined ?? null ?????????????????????????????????????????? JavaScript API ?? un-  
defined??? Map.get ????? DOM ? Google API ??????? null ??? Element.getAttribute ?????????  
null ? undefined ??????????????????
```

??/?/?/?/?/?/?

[illegible]

```

##### ?? ##### | null ?? | undefined ##### ?? #####

```

```
// ?????????????????????? undefined ?
type CoffeeResponse = Latte|Americano|undefined;

class CoffeeService {
  getLatte(): CoffeeResponse { ... };
}
```

```
// ?????????????????? undefined ?
type CoffeeResponse = Latte|Americano;

class CoffeeService {
    getLatte(): CoffeeResponse|undefined { ... };
}
```

```
// ?????????????????????
type CoffeeResponse = Latte|Americano;

class CoffeeService {
  getLatte(): CoffeeResponse {
    return assert(fetchResponse(), 'Coffee maker is broken, file a ticket');
  };
}
```

undefined

TypeScript ?

```
interface CoffeeOrder {
  sugarCubes: number;
  milk?: Whole|LowFat|HalfHalf;
}

function pourCoffee(volume?: Milliliter) { ... }
```

|undefined {sugarCubes: 1} CoffeeOrder milk

|undefined

```
class MyClass {
  field = '';
}
```

### 7.5.3

TypeScript

Mock

```
// 
const foo: Foo = {
  a: 123,
  b: 'abc',
}
```

```
// 
const badFoo = {
  a: 123,
  b: 'abc',
}
```

badFoo badFoo

badFoo Foo

```
interface Animal {
  sound: string;
  name: string;
}

function makeSound(animal: Animal) {}

/**
```

(continues on next page)

(continued from previous page)

```

    * 'cat' [REDACTED] '{sound: string}'
    */
const cat = {
    sound: 'meow',
};

/**
 * 'cat' [REDACTED]
 * [REDACTED] TypeScript [REDACTED]
 * [REDACTED] 'cat' [REDACTED]
 */
makeSound(cat);















/**
 * Horse [REDACTED]
 * [REDACTED] 'horse' [REDACTED] 'Animal' [REDACTED]
 */
const horse: Animal = {
    sound: 'niegh',
};

const dog: Animal = {
    sound: 'bark',
    name: 'MrPickles',
};

makeSound(dog);
makeSound(horse);

```

































#### 7.5.4 7.5.4 7.5.4 7.5.4

TypeScript              [illegible]

```
// ???????
interface User {
    firstName: string;
    lastName: string;
}
```

```
// ???????
type User = {
    firstName: string,
    lastName: string,
}
```

□□□□

```

TypeScript“

```

## 7.5.5 Array<T> 和 ReadonlyArray<T>

在 TypeScript 中，`Array<T>` 和 `ReadonlyArray<T>` 是两种不同的数组类型。前者是可变的，后者是不可变的。

```
// 示例
const a: string[];
const b: readonly string[];
const c: ns.MyObj[];
const d: Array<string|number>;
const e: ReadonlyArray<string|number>;
```

```
// 示例
const f: Array<string>; // 可变的
const g: ReadonlyArray<string>; // 不可变的
const h: {n: number, s: string}[]; // 不可变的
const i: (string|number)[];
const j: readonly (string|number)[];
```

## 7.5.6 字典类型 { [key: string]: number }

在 JavaScript 中，字典类型通常使用 `{ [key: string]: number }` 来表示。

```
const fileSizes: {[fileName: string]: number} = {};
fileSizes['readme.txt'] = 541;
```

在 TypeScript 中，字典类型通常使用 `{ [key: string]: number }` 来表示。

```
// 示例
const users: {[key: string]: number} = ...;
```

```
// 示例
const users: {[userName: string]: number} = ...;
```

在 TypeScript 中，字典类型通常使用 `{ [key: string]: number }` 来表示。ES6 中的 `Map` 和 `Set` 是 JavaScript 中的数据结构，ES6 中的 `Map` 和 `Set` 是 JavaScript 中的数据结构。

TypeScript 中的 `Record<Keys, ValueType>` 类型用于表示字典类型。

## 7.5.7 字典类型 { [key: string]: number }

TypeScript 中的 `Record<Keys, ValueType>` 类型用于表示字典类型。TypeScript 中的 `Record<Keys, ValueType>` 类型用于表示字典类型。

TypeScript 中的 `Record<Keys, ValueType>` 类型用于表示字典类型。

- 字典类型通常使用 `{ [key: string]: number }` 来表示。
- 字典类型通常使用 `{ [key: string]: number }` 来表示。
- 字典类型通常使用 `{ [key: string]: number }` 来表示。

- IDE “” Pick<T, Keys>

- 
- 
- 

TypeScript Pick<T, Keys> T

```
interface User {
  shoeSize: number;
  favoriteIcecream: string;
  favoriteChocolate: string;
}

// FoodPreferences favoriteIcecream favoriteChocolate shoeSize
type FoodPreferences = Pick<User, 'favoriteIcecream'|'favoriteChocolate'>;
```

FoodPreferences

```
interface FoodPreferences {
  favoriteIcecream: string;
  favoriteChocolate: string;
}
```

User FoodPreferences User FoodPreferences

```
interface FoodPreferences { /* */ }

interface User extends FoodPreferences {
  shoeSize: number;
  // User FoodPreferences
}
```

IDE

## 7.5.8 any

TypeScript any “” any —

any any

- 
- unknown any
- Lint any

????????

????????????????

```
// ????????????????? JSON
declare interface MyUserJson {
  name: string;
  email: string;
}

// ?????????????????
type MyType = number|string;

// ?????????????????
function getTwoThings(): {something: number, other: string} {
  // ...
  return {something, other};
}

// ????????????????? any
// ?????????????????
// ?????????????????"????????"
function nicestElement<T>(items: T[]): T {
  // 1 items
  // ????????????? T extends HTMLElement
}
```

?? unknown ?? any

any  
unknown unknown unknown any

```
// 
// null undefined val
// 
const val: unknown = value;
```

```
// 
const danger: any = value /* 
danger.whoops(); //
```

?? Lint ??? any ???

??? any Mock Lint any

```
// BookService
// Mock
// tslint:disable-next-line:no-any
const mockBookService = ({get() { return mockBook; }} as any) as BookService;
// 
// tslint:disable-next-line:no-any
const component = new MyComponent(mockBookService, /* unused ShoppingCart */ null as
→any);
```



## 7.5.11 类型别名

在 TypeScript 中，API 接口 API 可以定义为类型别名。

## 7.6 类型推断

在 TypeScript 中，类型推断是指编译器根据变量的使用方式来推断变量的类型。

### 7.6.1 类型推断

在 TypeScript 中，类型推断是指编译器根据变量的使用方式来推断变量的类型。例如，以下代码中，编译器会根据变量的使用方式来推断变量的类型。

1. 在 TypeScript 中，类型推断是指编译器根据变量的使用方式来推断变量的类型。

例如：

- any 类型：任何类型的值都可以被推断为 any 类型。
- TypeScript 类型：TypeScript 类型推断是指编译器根据变量的使用方式来推断变量的类型。
- 类型别名：类型别名是指为一个类型定义一个别名。
- 接口：接口是指一个类型的集合。
- private 属性：private 属性是指一个类的私有属性。

2. 在 TypeScript 中，类型推断是指编译器根据变量的使用方式来推断变量的类型。

在 TypeScript 中，类型推断是指编译器根据变量的使用方式来推断变量的类型。

在 TypeScript 中，JavaScript 类型推断是指编译器根据变量的使用方式来推断变量的类型。

例如：

- 类型别名：类型别名是指为一个类型定义一个别名。
- x as T 类型：x as T 类型是指将一个变量 x 推断为类型 T。
- Array<[number, number]> 类型：Array<[number, number]> 类型是指一个包含两个数字的数组。

3. 在 TypeScript 中，类型推断是指编译器根据变量的使用方式来推断变量的类型。

在 TypeScript 中，TypeScript 类型推断是指编译器根据变量的使用方式来推断变量的类型。

例如：

- 类型别名：类型别名是指为一个类型定义一个别名。
- 闭包：闭包是指一个函数及其相关的状态。
- TS 类型：TS 类型是指 TypeScript 类型。
- google3 类型：google3 类型是指 Google 公司的类型。

4. 在 TypeScript 中，类型推断是指编译器根据变量的使用方式来推断变量的类型。

在 TypeScript 中，类型推断是指编译器根据变量的使用方式来推断变量的类型。

在 TypeScript 中，Bug 是指程序中的错误。





HTML/CSS [?] - [?]

## 8.1 ??

HTML/CSSHTMLCSSGSS

## 8.2 ???????

### 8.2.1

□□□□□□□□□□

URL http: https: URL

```
<!-- 000 -->
<script src="http://www.google.com/js/gweb/analytics/autotrack.js"></script>

<!-- 00 -->
<script src="//www.google.com/js/gweb/analytics/autotrack.js"></script>
```

```
/* 222 */
.example {
  background: url(http://www.google.com/images/example);
}

/* 22 */
.example {
  background: url(//www.google.com/images/example);
}
```

### 8.3 ???????

### 8.3.1

□□□□□□□□□□

??TAB????TAB????

```
<ul>
  <li>Fantastic
  <li>Great
</ul>
```

```
.example {
  color: blue;
}
```

### 8.3.2

HTML text/CDATA CSS

```
<!-- -->
<A HREF="/">Home</A>

<!-- -->

```

```
/* */
color: #E5E5E5;

/* */
color: #e5e5e5;
```

### 8.3.3

```
<!-- -->
<p>What?_

<!-- -->
<p>Yes please.
```

## 8.4

### 8.4.1

UTF-8 BOM

UTF-8

HTML <meta charset="utf-8"> UTF-8

Handling character encodings in HTML and CSS



(continued from previous page)

```
<title>Test</title>
<article>This is only a test.</article>
```

### 8.5.3

??HTML??????

????????????“?”???????????????????????????? p ?????? a ????

?????HTML????????????????????

```
<!-- ??? -->
<div onclick="goToRecommendations();">All recommendations</div>

<!-- ?? -->
<a href="recommendations/">All recommendations</a>
```

#### 8.5.4 ☐☐☐☐☐

□□□□□□□□□□

[illegible]

```

XXXXXXXXXXXXXXXXXXXX alt XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX alt
XXXXXXXXXXXXXXXXXXXXXXXXCSSXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX alt="" XXXXX

```

```
<!-- 🚫 -->


<!-- 🚫 -->

```

### 8.5.5 ?????

□□□□□□□□□□□□

[illegible]

????????????HTML????HTML????????????????????????????????????????????????????????

????????????????????

????????????????????HTML????????????????????

```
<!-- ??? -->
<!DOCTYPE html>
<title>HTML sucks</title>
<link rel="stylesheet" href="base.css" media="screen">
<link rel="stylesheet" href="grid.css" media="screen">
<link rel="stylesheet" href="print.css" media="print">
<h1 style="font-size: 1em;">HTML sucks</h1>
<p>I've read about this on a few sites but now I'm sure:
  <u>HTML is stupid!!1</u>
```

(continues on next page)

(continued from previous page)

```

<center>I can't believe there's no way to control the styling of
  my website without doing everything all over again!</center>

<!-- ?? -->
<!DOCTYPE html>
<title>My first CSS-only redesign</title>
<link rel="stylesheet" href="default.css">
<h1>My first CSS-only redesign</h1>
<p>I've read about this on a few sites but today I'm actually
  doing it: separating concerns and avoiding anything in the HTML of
  my website that is presentational.
<p>It's awesome!

```

## 8.5.6

????????

????????????????????????????????????????UTF-8????????????????&mdash; ?&rdquo; ?&#x263a; ?????????  
 ?????????HTML????????????????????<?&????????????????????????????????????????

```

<!-- ??? -->
The currency symbol for the Euro is &ldquo;&eur;&rdquo;.

<!-- ?? -->
The currency symbol for the Euro is "€".

```

## 8.5.7

????????????????

????????????????????????????????????????HTML5?? ?????????????????????  
 ?????????????????????????????????????????????????????????????????????????????????????????Web????????????????????????????????????????????????????????????

```

<!-- ??? -->
<!DOCTYPE html>
<html>
  <head>
    <title>Spending money, spending bytes</title>
  </head>
  <body>
    <p>Sic.</p>
  </body>
</html>

<!-- ?? -->
<!DOCTYPE html>
<title>Saving money, saving bytes</title>
<p>Qed.

```

## 8.5.8 type

type

CSSJavaScripttype

HTML5 text/css text/javascript type

## 8.6 HTML

### 8.6.1

CSS display

li Linter

```
<blockquote>
  <p><em>Space</em>, the final frontier.</p>
</blockquote>

<ul>
  <li>Moe
  <li>Larry
  <li>Curly
</ul>

<table>
  <thead>
    <tr>
      <th scope="col">Income
      <th scope="col">Taxes
    </tr>
  <tbody>
    <tr>
      <td>$ 5.00
      <td>$ 4.50
    </tr>
  </tbody>
</table>
```

### 8.6.2 HTML

```
<!-- -->
<a class='maia-button maia-button-secondary'>Sign in</a>

<!-- -->
<a class="maia-button maia-button-secondary">Sign in</a>
```





## 8.7.4

```
id="example" class="error">
```

```
id="example" class="error">
```

```
id="example" class="error">
```

```
/*
 *
 */
ul#example {}
div.error {}

/*
 *
 */
#example {}
.error {}
```

## 8.7.5

```
font-family: palatino, georgia, serif;
```

```
CSS font-family: palatino, georgia, serif;
```

```
font-family: palatino, georgia, serif;
```

```
/*
 *
 */
border-top-style: none;
font-family: palatino, georgia, serif;
font-size: 100%;
line-height: 1.6;
padding-bottom: 2em;
padding-left: 1em;
padding-right: 1em;
padding-top: 0;

/*
 *
 */
border-top: 0;
font: 100%/1.6 palatino, georgia, serif;
padding: 0 1em 2em;
```

## 8.7.6

```
margin: 0; padding: 0;
```

```
margin: 0; padding: 0;
```

```
margin: 0; padding: 0;
```



## 8.7.11 Hacks

css “hacks”

css hacks

## 8.8 CSS

### 8.8.1

css

css-moz-webkit

```
background: fuchsia;
border: 1px solid;
-moz-border-radius: 4px;
-webkit-border-radius: 4px;
border-radius: 4px;
color: black;
text-align: center;
text-indent: 2em;
```

### 8.8.2

```
@media screen, projection {

  html {
    background: #fff;
    color: #444;
  }

}
```

### 8.8.3

```
/* */
.test {
  display: block;
  height: 100px;
}

/* */
```

(continues on next page)

(continued from previous page)

```
.test {
  display: block;
  height: 100px;
}
```

## 8.8.4 CSS

```


```

```


```

```
/*  */
h3 {
  font-weight:bold;
}

/*  */
h3 {
  font-weight: bold;
}
```

## 8.8.5

```


```

```


```

```


```

```
/*  */
#video{
  margin-top: 1em;
}

/*  */
#video
{
  margin-top: 1em;
}

/*  */
#video {
  margin-top: 1em;
}
```

### 8.8.6 ???? ???? ?

□□□□□□□□□□□□

□□□□□□□□□□□□□□□□

```
/* 222 */
a:focus, a:active {
  position: relative; top: 1px;
}

/* 22 */
h1,
h2,
h3 {
  font-weight: normal;
  line-height: 1.2;
}
```

### 8.8.7 CSS

????????

[illegible]

```
html {
    background: #fff;
}

body {
    margin: auto;
    width: 50%;
}
```

### 8.8.8 CSS??

**QUESTION**

?????????????????"?????""??? url?? ?????????

```
???????????? @charset ??? ?????????? ??????????
```

```

/* ??? */
@import url("//www.google.com/css/maia.css");

html {
    font-family: "open sans", arial, sans-serif;
}

/* ?? */
@import url(//www.google.com/css/maia.css);

html {
    font-family: 'open sans', arial, sans-serif;
}

```

## 8.9 CSS

### 8.9.1

css

```
/* Header */
#adw-header {}

/* Footer */
#adw-footer {}

/* Gallery */
.adw-gallery {}
```

### 8.10



JAVA [?] [?] [?] [?] - [?] [?] [?] [?]

9.1 0.

Liam Miller-Cushon  
 Ted Osborne  
 Git

Liam Miller-Cushon  
 Ted Osborne  
 Git

Liam Miller-Cushon  
 Ted Osborne  
 Git

Liam Miller-Cushon  
 Ted Osborne  
 Git

John.L

John.L

- Google Style Guide
- Google `fmt` - `fmt`

- Google Style Guide
- Google `fmt` - `fmt`

9.2 1.

Google Java™ Java Google

[illegible]

9.2.1 1.1.

????????????????

- 1. `??“?”?class?????????“?”? ??????????????????@interface ??`
- 2. `??“(??)??”?member????????????????????????????????????????????????????????`
- 3. `??“?”?comment?????????????????“???”? ??????????????????????“Javadoc”????`

“ ”



## 9.2.2 1.2.

Google

## 9.3 2.

### 9.3.1 2.1.

.java

### 9.3.2 2.2. UTF-8

UTF-8

### 9.3.3 2.3.

#### 2.3.1.

ASCII

- 1.
- 2. Tab

#### 2.3.2

Unicode

#### 2.3.3. ASCII

ASCII Unicode

Tip: Unicode

| String unitAbbrev = "μs";                              |  |
|--------------------------------------------------------|--|
| String unitAbbrev = "\u03bcs"; // "μs"                 |  |
| String unitAbbrev = "\u03bcs"; // Greek letter mu, "s" |  |
| String unitAbbrev = "\u03bcs";                         |  |
| return '\uffff' + content; // byte order mark          |  |

Tip: ASCII

## 9.4 3.

- 1.
- 2.
- 3. Import statements
- 4.

### 9.4.1 3.1.

### 9.4.2 3.2.

Column limit: 100

### 9.4.3 3.3.

#### 3.3.1.

#### 3.3.2.

Column limit: 100

#### 3.3.3.

- 
- 

```

ASCII
import packageA.ClassA; import packageA.
ClassA.; ASCII

```

### 3.3.4.

## 9.4.4 3.4.

### 3.4.1.

### 3.4.2.

“”

#### 3.4.2.1.

```
static private
```

## 9.5 4.

4.8.3.1

### 9.5.1 4.1.

#### 4.1.1.

```
if else for do while
lambda
```

#### 4.1.2. K & R

Kernighan & Ritchie“ ”

- 
- 
- 
- else

```
;

```

```
return () -> {
    while (condition()) {
        method();
    }
};

return new MyClass() {
    @Override public void method() {
        if (condition()) {
            try {
                something();
            } catch (ProblemException e) {
                recover();
            }
        } else if (otherCondition()) {
            somethingElse();
        } else {
            lastThing();
        }
        {
            int x = foo();
            frob(x);
        }
    }
};
```

4.8.1,

#### 4.1.3.

K & R 4.1.2 K & R try/catch/finally

```
// 
void doNothing() {}

// 
void doNothingElse() {
}
```

```
// 
try {
    doSomething();
} catch (Exception e) {}
```

## 9.5.2 4.2. `String`

String 4.1.2. `String` & `String`

## 9.5.3 4.3. `String`

`String`

## 9.5.4 4.4. `String`100

JavaString 100String“”UnicodeString4.5. `String`

Tip: `String`UnicodeString

`String`

- `String`JavadocURLJSNI
- `String`3.2 `String` 3.3 `String`
- `String`shell
- `String` google-java-format

## 9.5.5 4.5. `String`

`String` `String`

`String`

Tip: `String` `String`

Tip: `String` `String`

## 4.5.1. `String`

`String` `String` `String`

1. `String`GoogleC++JavaScript
  - `String`
    - \* `String` .
    - \* `String` ::
    - \* `String` <T extends Foo & Bar>
    - \* `catch` catch (FooException | BarException e)
2. `String`

- for “foreach”
- 3. (
- 4. ,
- 5. lambda

```
MyLambda<String, Long, Object> lambda =
    (String label, Long value, Object obj) -> {
        ...
    };

Predicate<String> predicate = str ->
    longExpressionInvolving(str);
```

Tip:

#### 4.5.2.

4.6.3

#### 9.5.6 4.6.

##### 4.6.1.

- 1. –
- 2. 3.3

##### 4.6.2.

Javadoc

- 1. if for catch (
- 2. else catch }
- 3. {
  - @SomeAnnotation({a, b})
  - String[][] x = {{"foo"}};
- 4. “”

- `<T extends Foo & Bar>`
- `catch (FooException | BarException e)`
- `for "foreach" :`
- `lambda (String str) -> str.length()`
- `Object::toString`
- `Object::toString`
- `object.toString()`
- 5. `, : ; )`
- 6. `//`
- 7. `//`
- 8. `List<String> list`
- 9. `new int[] {5, 6} and new int[] { 5, 6 }`
- `new int[] {5, 6} and new int[] { 5, 6 }`
- 10. `[] ...`

#### 4.6.3.

Google

```
private int x; // 
private Color color; // 

private int x; // 
private Color color; // 
```

Tip: conflicts

#### 9.5.7 4.7.

Java

## 9.5.8 4.8.

### 4.8.1.

```
private enum Answer {
    YES {
        @Override public String toString() {
            return "yes";
        }
    },
    NO,
    MAYBE
}
```

```
private enum Suit { CLUBS, HEARTS, SPADES, DIAMONDS }
```

### 4.8.2.

#### 4.8.2.1.

int a, b;

for

#### 4.8.2.2.

### 4.8.3.

#### 4.8.3.1.

```
new int[] {
    0, 1, 2, 3
}

new int[] {
    0, 1,
    2, 3
}

new int[] {
    0,
    1,
```

(continues on next page)



(continued from previous page)

```

    2,
    3
}

new int[]
{0, 1, 2, 3}

```

#### 4.8.3.2. `String[] args`

```
String[] args
```

#### 4.8.4. `switch`

```
switch (input) {
    case FOO:
    default:

```

##### 4.8.4.1. `break`

```
switch (input) {

```

```
    switch (input) {

```

##### 4.8.4.2. `break`

```
switch (input) {
    break;
    continue;
    return;
// fall through
switch (input) {

```

```

switch (input) {
    case 1:
    case 2:
        prepareOneOrTwo();
        // fall through
    case 3:
        handleOneTwoOrThree();
        break;
    default:
        handleLargeNumber(input);
}

```

```
switch (input) {
    case 1:

```

#### 4.8.4.3. `default`

```
switch (default)
```

```
enum switch default
IDE
```

#### 4.8.5.

##### 4.8.5.1.

```
@Target (ElementType.TYPE_USE)

```

```
final @Nullable String name;
public @Nullable Person getPersonByName (String name)
```

##### 4.8.5.2.

```
4.5.2 
```

```
@Deprecated
@CheckReturnValue
public final class Frozzler { ... }
```

##### 4.8.5.3.

```
 
```

```
@Deprecated
@Override
public String getNameIfPresent () { ... }
```

##### 4.8.5.4.

```
@Partial @Mock DataLoader loader;
```

#### 4.8.5.5.

#### 4.8.6.

Javadoc Javadoc

##### 4.8.6.1.

/\* ... \*/ // ... /\* ... \*/ \*  
 \* \* \*

```
/*
 * This is          // And so          /* Or you can
 * okay.            // is this.         * even do this. */
 */
```

Tip: /\* ... \*/  
 // ...

#### 4.8.7.

Java

```
public protected private abstract default static final transient volatile_
↪synchronized native strictfp
```

#### 4.8.8.

long L 1 3000000000L  
 3000000000L

## 9.6 5.

### 9.6.1 5.1.

ASCII \w+

Google name\_ mName s\_name kName

## 9.6.2 5.2. `com.example.deepspace`

### 5.2.1. `com.example.deepspace`

```
com.example.deepspace com.example.deepSpace com.example.deep_space
```

### 5.2.2. `com.example.deepspace`

```
com.example.deepspace.UpperCamelCase com.example.deepspace.Character com.example.deepspace.ImmutableList com.example.deepspace.List com.example.deepspace.Readable com.example.deepspace.Test com.example.deepspace.HashIntegrationTest com.example.deepspace.HashImplTest
```

### 5.2.3. `com.example.deepspace`

```
com.example.deepspace.lowerCamelCase com.example.deepspace.sendMessage com.example.deepspace.stop com.example.deepspace.transferMoney_deductsFromSource
```

### 5.2.4. `com.example.deepspace`

```
com.example.deepspace.UPPER_SNAKE_CASE com.example.deepspace.static final com.example.deepspace.null
```

```
// Constants
static final int NUMBER = 5;
static final ImmutableList<String> NAMES = ImmutableList.of("Ed", "Ann");
static final Map<String, Integer> AGES = ImmutableMap.of("Ed", 35, "Ann", 32);
static final Joiner COMMA_JOINER = Joiner.on(','); // because Joiner is immutable
static final SomeMutableType[] EMPTY_ARRAY = {};

// Not constants
static String nonFinal = "non-final";
final String nonStatic = "non-static";
static final Set<String> mutableCollection = new HashSet<String>();
static final ImmutableSet<SomeMutableType> mutableElements = ImmutableSet.of(mutable);
static final ImmutableMap<String, SomeMutableType> mutableValues =
    ImmutableMap.of("Ed", mutableInstance, "Ann", mutableInstance2);
static final Logger logger = Logger.getLogger(MyClass.getName());
static final String[] nonEmptyArray = {"these", "can", "change"};
```

```
com.example.deepspace
```

5.2.5. `computeValues`

```

// Compute values for the given index
// computedValues[index]

```

5.2.6. `public`

```

// public method
// public

```

5.2.7. `final`

```

// final
// final

```

5.2.8. `RequestT, FooBarT`

```

// RequestT, FooBarT
// RequestT, FooBarT

```

9.6.3 5.3. `Miller's algorithm`

```

// Miller's algorithm
// Miller's algorithm

```

- 1. `Miller's algorithm`
- 2. `Miller's algorithm`
- 3. `Miller's algorithm`
- 4. `Miller's algorithm`

```

// Miller's algorithm

```

| XML HTTP request        | newCustomerID     | innerStopWatch    |
|-------------------------|-------------------|-------------------|
| “XML HTTP request”      | newCustomerID     | innerStopWatch    |
| “new customer ID”       | newCustomerID     | innerStopWatch    |
| “inner stopwatch”       | innerStopWatch    | innerStopWatch    |
| “supports IPv6 on iOS?” | supportsIPv6OnIos | supportsIPv6OnIOS |
| “YouTube importer”      | YouTubeImporter   | YoutubeImporter * |

\*2022

Tip: 2022 “nonempty” “non-empty” 2022 checkNonempty 2022 checkNonEmpty 2022

## 9.7 6. 2022

### 9.7.1 6.1 @Override 2022

2022 @Override 2022  
2022 @Deprecated 2022 @Override 2022

### 9.7.2 6.2. 2022

2022 2022 “2022” 2022 Assertion-Error 2022

2022 catch 2022

```
try {
    int i = Integer.parseInt(response);
    return handleNumericResponse(i);
} catch (NumberFormatException ok) {
    // it's not numeric; that's fine, just continue
}
return handleTextResponse(response);
```

2022 2022 expected 2022

```
try {
    emptyStack.pop();
    fail();
} catch (NoSuchElementException expected) {
}
```

### 9.7.3 6.3. 2022

2022

```
Foo aFoo = ...;
Foo.aStaticMethod(); // 2022
aFoo.aStaticMethod(); // 2022
somethingThatYieldsAFoo().aStaticMethod(); // 2022
```

## 9.7.4 6.4. `Object.finalize`

`Object.finalize` [Effective Java](#)

Tip: [Effective Java](#)

## 9.8 7. Javadoc

### 9.8.1 7.1. `Javadoc`

#### 7.1.1. `Javadoc`

Javadoc

```
/**
 * Javadoc,
 * ...
 */
public int method(String p1) { ... }
```

```
/** Javadoc */
```

Javadoc `@return`

#### 7.1.2. `Javadoc`

HTML `<ul>` `<table>` `<p>`

#### 7.1.3. `Javadoc`

`@param` `@return` `@throws` `@deprecated`

### 9.8.2 7.2. `Javadoc`

Javadoc

```
A { @code Foo } is
a... This method returns... Save the record.
```

Tip: `Javadoc` `/** @return the customer ID */` `/** Returns the customer ID. */`

### 9.8.3 7.3. Javadoc

```
public public public protected Javadoc
7.3.4 Javadoc Javadoc
```

#### 7.3.1.

```
getFoo() Javadoc>Returns the foo"
```

---

Tip: `getCanonicalName()` "canonical name" `/** Returns the canonical name. */`

---

#### 7.3.2.

```
Javadoc
```

#### 7.3.4. Javadoc7.3.3

```
Javadoc
```

```
Javadoc /**
```

```
Javadoc7.1.17.1.27.1.37.2
```