

00A0 2203 $\exists$  2200 $\forall$  2286 $\subseteq$  2713x 27FA $\iff$  221A $\surd$  221B $\surd$  2295 $\oplus$  2297 $\otimes$

---

**Google**

**Aug 06, 2021**



---

# Contents

---

<b>1</b>	<b>Google ( )</b>	<b>1</b>
<b>2</b>	<b>C++ -</b>	<b>3</b>
2.1	0. . . . .	3
2.2	1. . . . .	4
2.3	2. . . . .	8
2.4	3. . . . .	12
2.5	4. . . . .	18
2.6	5. Google . . . . .	21
2.7	6. C++ . . . . .	22
2.8	7. . . . .	37
2.9	8. . . . .	41
2.10	9. . . . .	47
2.11	10. . . . .	61
2.12	11. . . . .	62
<b>3</b>	<b>Objective-C -</b>	<b>63</b>
3.1	Google Objective-C Style Guide . . . . .	63
3.2	. . . . .	66
3.3	. . . . .	70
3.4	. . . . .	73
3.5	Cocoa Objective-C . . . . .	75
3.6	Cocoa . . . . .	83
<b>4</b>	<b>Python -</b>	<b>85</b>
4.1	. . . . .	85
4.2	. . . . .	85
4.3	Python . . . . .	86
4.4	Python . . . . .	98
4.5	. . . . .	117
<b>5</b>	<b>Shell -</b>	<b>119</b>
5.1	. . . . .	119
5.2	. . . . .	119
5.3	Shell . . . . .	120
5.4	. . . . .	120
5.5	. . . . .	121

5.6	.....	122
5.7	.....	127
5.8	.....	130
5.9	.....	132
5.10	.....	133
<b>6 Javascript</b>	<b>-</b>	<b>135</b>
6.1	.....	135
6.2 Javascript	.....	135
6.3 Javascript	.....	144

# CHAPTER 1

---

Google ( )

---

- ReadTheDocs :
- GitHub [zh-google-styleguide](#)
- [release](#)

---

**Note:** .

Google , .

Google , [Google Style Guide](#)

---

: ( ). , .  
“ ” , “ (camelCase)” “ ” “ ”. Google . Google, ,

:

1. Google C++
2. Google Objective-C
3. Google Python
4. Google JSON
5. Google Shell

reStructuredText , Sphinx HTML / CHM / PDF .

- cpplint - , [google-c-style.el](#), Google Emacs .
- , [JavaScript Style Guide](#) [XML Document Format Style Guide](#), Yang.Y.



## CHAPTER 2

---

C++ -

---

### Contents

- C++ -

## 2.1 0.

4.45

Benjy Weinberger  
Craig Silverstein  
Gregory Eitzmann  
Mark Mentovai  
Tashana Landray

YuleFox  
Yang.Y  
acgtyrant  
lilinsanity

- Google Style Guide
- Google -



### 2.1.1 0.1

Google , . Google , . Google , . Google .  
:  
 , , . , , .  
Google , 5 , . , .  
 , , . Google , . , ,  
 .  
 , .  
 , , bug . , .  
 , Artistic License/GPL .  
:  
• 2015-08 : @lilinsanity Google CPP Style Guide 4.45  
• 2015-07 4.45 : acgtyrant C++ C++ C++11 Google  
C++ innocentim, farseerfc Arch Linux C++ Primer  
• 2009-06 3.133 : YuleFox 1.0 , , .  
Yang.Y YuleFox , : Google - .  
3.133 , , . Yang.Y , YuleFox .  
• 2008-07 1.0 : YuleFox Blog, .

### 2.1.2 0.2

C++ Google . C++ , C++ , bug, .  
C++ . , C++ .  
 , , C++ . “ ” , .  
 . “ ” . , . , .  
C++ . C++ . , . , .  
Google .  
: C++ , C++ .

## 2.2 1.

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

## 2.2.1 1.1. Self-contained

---

<b>Tip:</b>	self-contained,	.h	.inc	-inl.h	.
-------------	-----------------	----	------	--------	---

---

1.2. #define symbols.

self-contained	platform-specific	.inc
.h	.cc	-inl.h
	.cc	-inl.h

## 2.2.2 1.2. #define

---

**Tip:** #define , : <PROJECT>\_<PATH>\_<FILE>\_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.

---

**Tip:** #include

---

forward declaration .

- #include
- #include
- 
- API.
- std:: symbol
- #include includes

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

// good_user.cc:
#include "b.h"
```

(continues on next page)

(continued from previous page)

```
void f(B*);
void f(void*);
void test(D* x) { f(x); } // calls f(B*)
```

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

- symbol include
- .
- .
- #include.
- #include.

1.5. #include

2.2.4 1.4.

Tip: 10 .

```
:
, , .
:
, . , , .
:
. , . , . , .
:
, 10 . , , !
: switch ( , switch ).
, ; . , . YuleFox : , , ).
, , , .
```

2.2.5 1.5. #include

Tip: , : , C , C++ , .h, .h.

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

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

```
, dir/foo.cc dir/foo_test.cc dir2/foo2.h , foo.cc :
1. dir2/foo2.h ( , )
```

2. C
3. C++
4. .h
5. .h

```

dir2/foo2.h      dir/foo.cc  dir/foo_test.cc
dir/foo.cc  dir2/foo2.h      ( base/basicitypes_unitytest.cc  base/basicitypes.h), .

```

```

(symbols)      include      (forward declarations)      bar.h      ,      foo.h      bar.
h,      bar.h,      foo.h      bar.h      symbol.      cc      cc      foo.cc      foo.h

```

, google-awesome-project/src/foo/internal/fooserver.cc :

```

#include "foo/public/fooserver.h" //
#include <sys/types.h>
#include <unistd.h>

#include <hash_map>
#include <vector>

#include "base/basicitypes.h"
#include "base/commandlineflags.h"
#include "foo/public/bar.h"

```

system-specific conditional includes includes

```

#include "foo/public/fooserver.h"

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

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

```

### 2.2.6 (YuleFox)

1. ;
2. ;
3. ;
4. -inl.h ( :D);
5. ( , );
6. . .. , , , , " " ( :D) , , , ,

## 2.2.7 acgtyrant

1. #includes .inc
2. Google -inl.h
3. incomplete type
4. .cc
5. #include , C , C++ , .h .h

## 2.3 2.

### 2.3.1 2.1.

---

**Tip:** .cc static . , . using using-directive inline namespace

---

```

:
    , , .
:
    (YuleFox : ), .
    , Foo, . , project1::Foo project2::Foo .

```

```

namespace X {
inline namespace Y {
void foo();
} // namespace Y
} // namespace X

```

X::Y::foo() X::foo() ABI

```

:
    ,
    C++ (One Definition Rule (ODR)).
:
    .

```

- 
- 
- , gflags / , :

```
// .h
namespace mynamespace {

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

} // namespace mynamespace
```

```
// .cc
namespace mynamespace {

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

} // namespace mynamespace
```

.cc , , gflags using

```
#include "a.h"

DEFINE_FLAG(bool, someflag, false, "dummy flag");

namespace a {

...code for a... //

} // namespace a
```

- std , . std , . , .
- using

```
// --
using namespace foo;
```

- API

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

```
// .h
namespace librarian {
namespace impl { //
namespace sidetable = ::pipeline_diagnostics::sidetable;
} // namespace impl
```

(continues on next page)



```

namespace myproject {
class FooBar {
public:
    static void Function1();
    static void Function2();
};
} // namespace myproject

```

```

, ; . , .
, .cc , 2.1. static ( static int Foo() {...}) .

```

### 2.3.4 2.4.

#### Tip:

C++ , . , . , . , :

```

int i;
i = f(); // --

```

```

int j = g(); // --

```

```

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

```

```

vector<int> v = {1, 2}; // --v

```

if, while for :

```

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

```

**Warning:** , , , . .

```

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

```

```

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

```



### 2.3.5 2.5.

---

**Tip:** POD POD

---

bug constexpr  
 (POD : Plain Old Data): int, char float, POD  
 C++ bug. POD getenv() getpid()

---

**Note:** Xris :  
 (unspecified behaviour)

---

main() exit() string  
 quick\_exit() exit() atexit() handlers. quick\_exit()  
 handler log \_at\_quick\_exit(). exit() quick\_exit() handler,  
 POD vector ( C ) string ( const char [])  
 class main() pthread\_once() raw

---

**Note:** Yang.Y :  
 , : , , , .

---

### 2.3.6 (YuleFox)

1. cc , , using ;
2. , , public;
3. , , ;
4. ( ) class ( STL ), bug.
5. , , , , / .

### 2.3.7 acgtyrant

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

### 2.4 3.

C++ . , . .

---

### 2.4.1 3.1.

- `const` , .
- `bool IsValid()` , .
- `Init()` .

(non-trivial) , `Init()` . Avoid `Init()` methods on objects with no other states that affect which public methods may be called ( ).

### 2.4.2 3.2.

`explicit` .

`int` `double` .

`( operator bool())` .

`explicit` ( C++11 ) , `cast` , C++11 :

```
class Foo {
    explicit Foo(int x, double y);
    ...
};

void Func(Foo f);
```

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

- `explicit` .

```

    •
    • ,
    • , , ,
    • , ,
    •
    • explicit , explicit
    • ,
    • , ,
    , explicit , explicit,
    explicit. std::initializer_list explicit, ( MyType m = {1, 2};).

```

### 2.4.3 3.3.

```

, / ,
, , . string
, ( ). std::unique_ptr<int>
/
, API , , ,
API ,
/ , Clone(), CopyFrom() or Swap(), , = default.
, ,
, , (Registerer), (Cleanup), (Mutex) /
, , ,
/ , , ,
, ,
/ , /

```

```

class Foo {
public:
    Foo(Foo&& other) : field_(other.field) {}

```

(continues on next page)

(continued from previous page)

```

// , , .

private:
    Field field_;
};

, / ( ). , public virtual Clone() protected
.

/ , public = delete .

// MyClass is neither copyable nor movable.
MyClass(const MyClass&) = delete;
MyClass& operator=(const MyClass&) = delete;

```

## 2.4.4 3.4. VS.

```

struct, class.

C++ struct class . , .
struct , , , . , , , Initialize(), Reset(), Validate()
, .
, class . , class.
STL , class struct.
: .

```

## 2.4.5 3.5.

```

(YuleFox : GoF <<Design Patterns>> ) . , public .

```

```

, . C++ , : , ; , .

```

```

. , . , API. API , .

```

```

, , . , . , .

```

```

public . , .

```

```

. " " ("is-a", YuleFox : "has-a" ) : Bar " " Foo, Bar Foo.

```

```

, virtual. , .

```

```

, protected . , .

```

, override, ( ) final . ( C++11) virtual . , , override,  
 final virtual . override final , , , , , .

### 2.4.6 3.6.

. : ; Interface .

. .

( ), .

. , , .

, . , Interface .

, Windows .

### 2.4.7 3.7.

, Interface ( ).

, :

- ("=0") ( ).
- .
- . , , protected.
- , Interface .

, . , ( 1 , ). Stroustrup *The C++ Programming Lan-*  
*guage, 3rd edition* 12.4 .

Interface . , Java , .

Interface , , .

, Interface , , Interface .

### 2.4.8 3.8.

```

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

```

```

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

```

- , , , Bug.
- , .
- , .
- , .
- grep , C++ .
- , . : foo < bar , &foo < &bar .
- . , & , . &&, || , .
- , , . , , .
- C++ .

```

, . | , shell .

```

```

. , .cc . , . , . ,
, . , < , , < > true.
. , . a < b b < a , .
. , ==, =, << Equals(), CopyFrom() PrintTo(). , , ,
std::set , <.
&&, ||, , &. operator" , ,
. = . << . , .

```

### 2.4.9 3.9.

```

private, static const ( ). , Google Test protected.

```

### 2.4.10 3.10.

```

, public .

```

```
public: , protected:, private:.
, : ( typedef, using ), , , , , , .
. , .
```

### 2.4.11 (YuleFox)

1. ;
2. , , , ;
3. , explicit;
4. , , private ;
5. struct;
6. > > > , virtual , ;
7. , , , ;
8. Interface , , , , , , protected;
9. , , , ;
10. ;
11. : public -> protected -> private;
12. , , ;

## 2.5 4.

### 2.5.1 4.1.

```
: , .
C/C++ , , . const , / const . , , , ,
, .
. / ( ) . , .
```

### 2.5.2 4.2.

```
, .
, . 40 , .
, , , bug. , .
, . : / , , .
```

### 2.5.3 4.3.

const.

C , , , int foo(int \*pval). C++ , : int foo(int &val).

(\*pval)++ . . .

, .

, const:

```
void Foo(const string &in, string *out);
```

Google Code : const , . const , const , , swap().

, const T\* const T& . :

- .
- .

, const T&. const T\* . const T\*, , .

### 2.5.4 4.4.

, , . .

const string& , const char\* :

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

, . , .

(acgyrant ), C++ , . , , .

, . , AppendString() AppendInt() , Append() . ,  
std::vector .





, , . ( Lambda ) , , , , .

## 2.6 5. Google

Google / C++ , C++ .

### 2.6.1 5.1.

>

>

```
* -> . , . std::unique_ptr C++11 , , ;
std::unique_ptr , . std::unique_ptr , move . std::shared_ptr , , ;
```

>

- , .
- , , .
- ” ” , .
- , , .
- , , .
- const , , .

>

- , API , , , .
- , .
- API , .
- , .
- std::unique\_ptr C++11 move , , .
- , , .
- , .
- ( ), .
- .

>

, , , , . std::unique\_ptr ,

```
std::unique_ptr<Foo> FooFactory();
void FooConsumer(std::unique_ptr<Foo> ptr);
```

```

std::shared_ptr<T> foo() {
    return std::shared_ptr<T>(new T(42));
}

std::auto_ptr<T> foo() {
    return std::auto_ptr<T>(new T(42));
}

```

## 2.6.2 5.2. CppLint

```

> cpplint.py --quiet --recursive --exclude=*.h --exclude=*.hpp --exclude=*.inl
> cpplint.py --quiet --recursive --exclude=*.h --exclude=*.hpp --exclude=*.inl // NOLINT, // NOLINTNEXTLINE,
    cpplint.py --quiet --recursive --exclude=*.h --exclude=*.hpp --exclude=*.inl cpplint.py.

```

## 2.6.3 acgtyrant

1. [acgtyrant](#)
2. Rust Ownership C++
3. `scoped_ptr` `auto_ptr` `shared_ptr` `weak_ptr`
4. [acgtyrant](#)
5. Arch Linux [AUR](#) `cpplint`

## 2.7 6. C++

### 2.7.1 6.1.

---

**Tip:** `const`.

---

```

:
C, int foo(int *pval). C++, : int foo(int &val).
:
(*pval)++ . NULL .
:
,
:
, const:

```

```
void Foo(const string &in, string *out);
```

Google Code : `const` , `const` , `const` `swap()`.

```
const T* const T&
```

- `null`

•

const T&    const T\*    const T\*,

## 2.7.2 6.2.

**Tip:**    .    std::forward.

```

:
    ,    .    , void f(string&& s);    .
:
    (    )    .    ,    v1    vector<string>,    auto v2(std::move(v1))
    ,    .
    ,    .
    ,    .
    ,    std::unique_ptr, std::move    .
:
    ( C++11 ),    .    ,    .
:
    ,    std::forward    .    std::move    .

```

## 2.7.3 6.3.

**Tip:**    call site

```

:
    const string&    ,    const char*    :

```

```

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

```

```

:
    ,    .    ,    .
:
    acgtyrant    C++
:
    ,    AppendString() AppendInt()    Append().

```

## 2.7.4 6.4.

**Tip:**

```

:
:
:           function signature           call
site  acgtyrant
:
:           acgtyrant
:
:           .cc

```

```

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

```

## 2.7.5 6.5. alloca()

**Tip:** alloca().

```

:
:           .   alloca()   .
:
:   alloca()   C++   ,   ,   bugs: “   ,   ”.
:
:           allocator   std::vector   std::unique_ptr<T[]>.

```

## 2.7.6 6.6.

**Tip:**

```

,   .   FooBuilder   Foo   ,   FooBuilder   Foo   ,   .   ,
.
( )   .   ,   public,   ,   .   .

```

### 2.7.7 6.7.

**Tip:** C++ .

```

:
• failures acgtyrant error code, int
• C++ Python, Java C++
• C++
• acgtyrant factory function, C++ Init() , " "
•
:
• throw f() g(),g() h(), h f g,
•
• RAII . , , " " ( ). ,
•
•
:
, . , . , . Google C++ ,
.
Google , . , . , , .
, . Google , , Google .
Windows , .
(YuleFox : , , C++ , Google , , , )

```

### 2.7.8 6.8.

TODO

**Tip:** RTTI.

```

:
RTTI C++ . typeid dynamic_cast .
:
RTTI ( ) . , , .
RTTI . , . RTTI .
RTTI . :

```

```

bool Base::Equal(Base* other) = 0;
bool Derived::Equal(Base* other) {
    Derived* that = dynamic_cast<Derived*>(other);
    if (that == NULL)
        return false;
    ...
}

```

:  
 RTTI . switch . , .  
 :  
 RTTI , . RTTI, . , RTTI . ,  
 :  
 , , .  
 , dynamic\_cast. , dynamic\_cast .  
 , . :

```

if (typeid(*data) == typeid(D1)) {
    ...
} else if (typeid(*data) == typeid(D2)) {
    ...
} else if (typeid(*data) == typeid(D3)) {
    ...
}

```

, . , , .  
 RTTI . RTTI , . , .

### 2.7.9 6.9.

---

**Tip:** C++ , static\_cast<>(). int y = (int)x int y = int(x) ;

---

:  
 C++ C , .  
 :  
 C ; ( (int)3.5), ( (int)"hello"). , C++ .  
 :  
 .  
 :  
 C . C++ .  
 • static\_cast C , .

- `const_cast` `const` .
- `reinterpret_cast` . . . . .
- `dynamic_cast` 6.8. .

### 2.7.10 6.10.

Tip:

```

:
printf() scanf().
:
, . ( gcc printf ).
:
pread() . printf , ( .*s) . (%1s), .
:
, . printf .
, . .
:
, . (Only One Way): I/O , I/O . , printf
+ read/write. , . , .
, . . . . , : , . :

```

```

cout << this; //
cout << *this; //

```

```

<< , . .
printf , , . , , ?

```

```

cerr << "Error connecting to '" << foo->bar()->hostname.first
<< ":" << foo->bar()->hostname.second << ": " << strerror(errno);

fprintf(stderr, "Error connecting to '%s:%u: %s",
foo->bar()->hostname.first, foo->bar()->hostname.second,
strerror(errno));

```

```

, " " , , ? , , .
, " , " , printf + read/write.

```

### 2.7.11 6.11.

Tip:

`(++i)` , .



```

:
    (++i i++) (--i i--) , ( ).
:
    , (++i) (i++) . ( ) i . i , . , ?
:
    C , , , for . , , (i) (++) .
:
    ( ), . , ( ).

```

### 2.7.12 6.12. const

---

**Tip:** const. C++11 constexpr

---

```

:
    const ( const int foo ). const ( class Foo { int
    Bar(char c) const; });.
:
    . , , . , . , .
:
    const : const , const ( const_cast ), .
:
    const , , ; . , const:
    • , const.
    • const. const. , const , const const.
    • , const.
    , const. const int * const * const x; , x. : const int**
    x .
    mutable , , .
const :
    int const *foo , const int* foo, : const . , “ ”
    . const , (const) (int) .
    , const . ! (Yang.Y : const , , , .)

```

### 2.7.13 6.13. constexpr

---

**Tip:** C++11 constexpr

---

```

:

```

```

    constexpr          constexpr,   constexpr
:
    constexpr
:
    constexpr          constexpr
:
    constexpr      C++          constexpr          constexpr          const-
    expr

```

### 2.7.14 6.14.

---

**Tip:** C++ , int. , <stdint.h> , int16\_t. 2<sup>31</sup> (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, unsigned long long . C
, int. , size_t ptrdiff_t.
, int, . int. int 32 , 32 . 64 , int64_t uint64_t.
, int64_t.
uint32_t , , . , . , .
size
acgtyrant integer promotions, int unsigned int unsigned int
:
, , . , C , bug . :
for (unsigned int i = foo.Length()-1; i >= 0; --i) ...
! gcc bug , . bug . C
, , !

```

## 2.7.15 6.15. 64

**Tip:** 64 32 . , , :

- , printf() 32 64 . C99 . , MSVC 7.1 , , ( inttypes.h ):

```
// 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

PRI\* . , . PRI\* % . , printf("x = %30"PRIuS"\n", x) 32 Linux printf("x = %30" "u" "\n", x), printf("x = %30u\n", x) (Yang.Y : MSVC 6.0 , VC 6 ).

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

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

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

## 2.7.16 6.16.

**Tip:** , , .

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

( ) ( # , ## ). , .  
; , :

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

### 2.7.17 6.17. 0, nullptr NULL

**Tip:** 0, 0.0, nullptr NULL, () '\0'.

0, 0.0, .  
( ), 0, NULL nullptr. C++11 nullptr; C++03 NULL, C++ NULL  
sizeof(NULL) sizeof(0)  
( ) '\0', .

### 2.7.18 6.18. sizeof

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

sizeof(varname) . 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;
}
```

## 2.7.19 6.19. auto

Tip: auto

C++11 auto, auto

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

C++

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

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

auto

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

auto,

```
auto i = x.Lookup(key);
```

x

auto const auto&

auto C++11

```
auto x(3); //
auto y{3}; //
```

—x int, y std::initializer\_list<int>. acgyrant normally-invisible  
 proxy types, C++ Why is vector<bool> not a STL container?

auto, —API

auto

auto

auto C++11 trailing return type lambda

## 2.7.20 6.20.

**Tip:**

C++03    aggregate types

```
struct Point { int x; int y; };
Point p = {1, 2};
```

C++11

```
// Vector
vector<string> v{"foo", "bar"};

//
//
vector<string> v = {"foo", "bar"};

// new
auto p = new vector<string>{"foo", "bar"};

// map pair,
map<int, string> m = {{1, "one"}, {2, "2"}};

//
vector<int> test_function() { return {1, 2, 3}; }

//
for (int i : {-1, -2, -3}) {}

//
void TestFunction2(vector<int> v) {}
TestFunction2({1, 2, 3});
```

std::initializer\_list&lt;T&gt;

```
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&lt;T&gt;

```
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. .

## 2.7.21 6.21. 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`.

- STL Lambdas
- Lambdas, `std::functions` `std::bind` general purpose callback mechanism
- Lambdas
- Lambdas
- `format` lambda
- `[](int x) {return x + n};`, `[n](int x) {return x + n};` n

- [acgtyrant](#) [lambda](#)
- [lambd](#) [auto.](#)

### 2.7.22 6.22.

---

**Tip:**

---

- [c++](#) ,
- [Google Test](#), [std::tuple](#), [std::function](#) [Boost.Spirit](#).
- [c++](#) , [. debug](#)
- [: , .](#)
- [\(Visual Assist X, Refactor for C++ \)](#) , [AST](#)
- [, . ,](#)
- [, SFINAE, sizeof trick](#) , [c++](#) ,
- [, , , ,](#)

### 2.7.23 6.23. 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

## 2.7.24 6.24. C++11

---

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

---

C++11 ' <<https://en.wikipedia.org/wiki/C%2B%2B11>>'

C++11 C++ C++

C++11 1300 vs 800

6.23. Boost C++11 —

C++11 C++11

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

## 2.7.25 acgyrant

1. `void a()` `void a(int b = 0),` `int`
- 2.
- 3.
4. `friend` `friend` `.cc`
- 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.

```

: , , , , , , , .
, , , , , .

```

### 2.8.1 7.1.

```

, , ; .

```

```

, , * , .

```

```

int price_count_reader; //
int num_errors; // "num"
int num_dns_connections; // "DNS"

```

```

int n; // .
int nerr; // .
int n_comp_conns; // .
int wgc_connections; // .
int pc_reader; // "pc" .
int cstmr_id; // .

```

```

, , i T .
: , .

```

## 2.8.2 7.2.

```

,      (-) (-),      ,      “_” .

:
• my_useful_class.cc
• my-useful-class.cc
• myusefulclass.cc
• myusefulclass_test.cc // _unittest _regtest .
C++ .cc , .h . .inc , .
/usr/include (Yang.Y : ), db.h.
.http_server_logs.h logs.h . , foo_bar.h foo_bar.cc, FooBar.
.h . , .h .

```

## 2.8.3 7.3.

```

,      : MyExcitingClass, MyExcitingEnum.

```

```

—— , , (typedef), , —— , , , . :

```

```

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

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

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

//
enum UrlTableErrors { ...

```

## 2.8.4 7.4.

```

( ) , . , , : a_local_variable, a_struct_data_member,
a_class_data_member_.

```

:

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

, , .

```
class TableInfo {
    ...
private:
    string table_name; // - .
    string tablename; // .
    static Pool<TableInfo>* pool_; // .
};
```

, , :

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

, *vs.* .

## 2.8.5 7.5.

```
constexpr const , , "k" , . :
```

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

( , ) . , , . , .

## 2.8.6 7.6.

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

```

, ( " " " "), . , ( , StartRpc() StartRPC()).
AddTableEntry()
DeleteUrl()
OpenFileOrDie()
( , API , , .)
. , . int count() void set_count(int count).

```

**2.8.7 7.7.**

```

. . .
. , .
. , .
websearch::index_util) ( websearch::util). std . (websearch::index,
internal , internal ( , ). , ( frobber.h,
websearch::index::frobber_internal).

```

**2.8.8 7.8.**

```

: kEnumName ENUM_NAME.
. . UrlTableErrors ( AlternateUrlTableErrors) , .

```

```

enum UrlTableErrors {
    kOK = 0,
    kErrorOutOfMemory,
    kErrorMalformedInput,
};
enum AlternateUrlTableErrors {
    OK = 0,
    OUT_OF_MEMORY = 1,
    MALFORMED_INPUT = 2,
};

```

2009 1 , . , . , . , .

**2.8.9 7.9.**

```

, ? , : MY_MACRO_THAT_SCARES_SMALL_CHILDREN.
; . , , :

```

```
#define ROUND(x) ...
#define PI_ROUNDED 3.0
```

## 2.8.10 7.10.

```
C/C++ , .
bigopen(): , open()
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.

```
, . . : , . , .
, . , !
```

### 2.9.1 8.1.

```
// /* */, .
// /* */ ; // . .
```

### 2.9.2 8.2.

```
.
. , , , , . .
```

```

    .( , Apache 2.0, BSD, LGPL, GPL)
    ,

```

```

.h , , , ,
.h .cc ,

```

### 2.9.3 8.3.

```

// 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 {
    ...
};

```

```

, , , ,
,
( .h .cc ), ,

```

### 2.9.4 8.4.

```

;

```

```

, . ( , ). ("Opens the file") ("Open the 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;
```

, . “ false”, :

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

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

, . , , , . , .  
.h . , .

2.9.5 8.5.

. , .

( ) . ( , , ) , . , .

, NULL -1 , . :



```
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_;
```

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

## 2.9.6 8.6.

```
// 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.
```

(continues on next page)

(continued from previous page)

```

}
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);

```

```

.
,      ,      C++      .      C++      ,      /      :
,      .
:

```

```

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

```

:

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

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

### 2.9.7 8.7. ,

```
, ; .
. , . , , .
, . , .
```

### 2.9.8 8.8. TODO

```
, , TODO .
TODO TODO, , , bug ID, TODO issue. ( ) TODO . TODO
, TODO , .
```

```
// 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
```

TODO “ ”, “Fix by November 2005”, (“Remove this code when all clients can handle XML responses.”).

### 2.9.9 8.9.

```
DEPRECATED comments .
DEPRECATED , . , .
DEPRECATED , , .
, . C++ , , .
DEPRECATED , , callsites , .
, . , .
```

### 2.9.10 (YuleFox)

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

## 2.10 9.

, , . , , , , .  
 , emacs .

### 2.10.1 9.1.

80.  
 , , .  
 . 80 . ?  
 . 80 60 ; , .  
 80 .  
 , 80 , . , URL 80 .  
`#include 80 .`  
 .

### 2.10.2 9.2. ASCII

ASCII , UTF-8 .  
 , UTF-8 , ASCII . , , ASCII ; ( ) ASCII  
 . , UTF-8 , UTF-8 .  
 , — "\xEF\xBB\xBF", u8"\uFEFF", Unicode , UTF-8 ,  
 .  
 (Yang.Y : "\xEF\xBB\xBF" UTF-8 with BOM )

u8 uXXX 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.

, 2 .

.

### 2.10.4 9.4.

,

:

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

,

```
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*/) {}
```

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

, , , :

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

## 2.10.5 9.5. Lambda

Lambda ; , .

, & .

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

lambda .

```
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.

, , . , , .

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

, , ,

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

,

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

## 2.10.8 9.8.

```
. if else .
. , .
. , . , . , . , .
```



```

if (condition) { // ...
    ... // 2 .
} else if (...) { // else if ...
    ...
} else {
    ...
}

```

:

```

if ( condition ) { // -
    ... // 2 .
} else { // else if .
    ...
}

```

if . :

```

if(condition) // - IF .
if (condition){ // - { .
if(condition){ // .

```

```

if (condition) { // - IF { .

```

, . else :

```

if (x == kFoo) return new Foo();
if (x == kBar) return new Bar();

```

else :

```

// - ELSE IF
if (x) DoThis();
else DoThat();

```

, , ; . if :

```

if (condition)
    DoSomething(); // 2 .

if (condition) {
    DoSomething(); // 2 .
}

```

if-else , :

```

// - IF ELSE .
if (condition) {
    foo;
} else
    bar;

// - ELSE IF .

```

(continues on next page)

(continued from previous page)

```

if (condition)
    foo;
else {
    bar;
}

```

```

//
if (condition) {
    foo;
} else {
    bar;
}

```

## 2.10.9 9.9.

```

switch (var) {
    case 0:
        ...
        break;
    case 1:
        ...
        break;
    default:
        warning();
} continue.

```

```

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

```

```

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

```

```

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");
}

```

```

{} continue,

```

```

while (condition) {
    //
}

```

(continues on next page)

(continued from previous page)

```
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) {
    ...
}
```

, (&&) . Google , . , , && ~, and compl.

### 2.10.12 9.12.

return .

x = expr return expr; .

```
return result; // , .
// , .
return (some_long_condition &&
        another_condition);
```

```
return (value); // var = (value);
return(result); // return
```

### 2.10.13 9.13.

=, () {} .

=, () {},

```
int x = 3;
int x(3);
int x{3};
string name("Some Name");
string name = "Some Name";
string name{"Some Name"};
```

{...} std::initializer\_list . std::initializer\_list, , .  
std::initializer\_list , .

```
vector<int> v(100, 1); // 100 1 .
vector<int> v{100, 1}; // 100 1 .
```

, , .

```
int pi(3.14); // - pi == 3.
int pi{3.14}; // : .
```

## 2.10.14 9.14.

```

// -
    if (lopsided_score) {
#ifdef DISASTER_PENDING // -
        DropEverything();
# if NOTIFY // - #
        NotifyClient();
# endif
#endif
        BackToNormal();
    }

```

```

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

```

## 2.10.15 9.15.

public:, protected:, private:, 1 .

( , ) :

```

class MyClass : public OtherClass {
public: //
    MyClass(); //
    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();

```

(continues on next page)

(continued from previous page)

```

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

```

```

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

```

```

namespace foo {
namespace bar {

```

## 2.10.18 9.19.

```

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

// .
class Foo : public Bar {

```

(continues on next page)

(continued from previous page)

```

public:
    //
    //
    Foo(int b) : Bar(), baz_(b) {} //
    void Reset() { baz_ = 0; } //
    ...

```

IDE) . , . , ; ). (Yang.Y : , / , ,

```

if (b) { // if
} else { // else
}
while (test) {} //
switch (i) {
for (int i = 0; i < 5; ++i) {
switch ( i ) { //
if ( test ) { // , .
for ( int i = 0; i < 5; ++i ) {
for ( ; i < 5 ; ++i) { // ; , ;
switch (i) {
    case 1: // switch case
        ...
    case 2: break; // , .

```

```

//
x = 0;

//
//
v = w * x + y / z;
v = w*x + y/z;
v = w * (x + z);

//
x = -5;
++x;
if (x && !y)
    ...

```

```

// (< and >) , < , > (
vector<string> x;

```

(continues on next page)



(continued from previous page)

```

y = static_cast<char*>(x);

//
vector<char *> x;

```

### 2.10.19 9.19.

- 
- if-else

### 2.10.20 (YuleFox)

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

### 2.10.21 acgtyrant

1. 80 , , .
2. Linux Locale , Windows.
3. Google if-else , , . Apple .
4. , int\* a, b vs int \*a, b, b int \* , , .

5. C++ Alternative operator representations, .
6. Constructor Initializer List Initializer List , .
7. , , . , ; , if (true) true.
8. void return , Google leveldb ; Is a blank return statement at the end of a function whos return type is void necessary? , return; return ; cpplint , , .

## 2.11 10.

### 2.11.1 10.1.

#### 2.11.2 10.2. Windows

Windows , Windows Microsoft . , C++ .

- Windows , :
- ( iNum). Google , .cc .
- Windows (YuleFox : , ), DWORD, HANDLE . Windows API . , C++ , const TCHAR \* LPCTSTR.
- Microsoft Visual C++ , 3 , (warnings) (errors) .
- #pragma once; Google . (Yang.Y : #ifndef SRC\_DIR\_BAR\_H\_, #define ).
- , , #pragma \_\_declspec. \_\_declspec(dllimport) \_\_declspec(dllexport) , , DLLIMPORT DLEXPOT, .
- , Windows :
- , COM ATL/WTL . COM ATL/WTL / , .
- , ATL STL Visual C++ STL) . ATL , \_ATL\_NO\_EXCEPTIONS . STL , , . ( STL, ).
- , StdAfx.h precompile.h . , ( precompile.cc ), /FI .
- resource.h , .

## 2.12 11.

```
    , .  
    , . if , . (*) , .  
    , . , , , , , .  
    ; . !
```

## 3.1 Google Objective-C Style Guide

2.36

Mike Pinkerton  
Greg Miller  
Dave MacLachlan

ewangke  
Yang.Y

- Google Style Guide
- Google -

### 3.1.1

**ewanke**

style guide 7 vim HTML  
2011.03.27

"ewangke at gmail.com"

**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;
    
```

(continues on next page)

(continued from previous page)

```

// Designated initializer. |bar| is a thing that represents a thing that
// 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];
}

```

(continues on next page)

```
- (NSString *)bar {
    return bar_;
}

- (void)setBar:(NSString *)bar {
    [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 MyProtocoledClass : NSObject<NSWindowDelegate> {
    @private
    id<MyFancyDelegate> delegate_;
}
```

(continues on next page)

(continued from previous page)

```
- (void)setDelegate:(id<MyFancyDelegate>)aDelegate;
@end
```

### 3.2.8

**Tip:** block target/selector 4

- 
- 
- 4
- 20
- `^{        ^ (        ) {`
- 

```
// 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];
        }
    }];
```

(continues on next page)

(continued from previous page)

```

// 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];
         }
     }
 ];

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

```

### 3.3

Objective-C	Objective-C naming rules	C++	Google	C++
Objective-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



(continued from previous page)

```
userInfo = [someObject object];
port = [network port];
```

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

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	retained			
CoreFoundation	C++	Objective-C	retained	__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_;
}
}
```

(continues on next page)

(continued from previous page)

```

@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

### 3.5.4 NSObject

**Tip:** NSObject @implementation

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



### 3.5.5

**Tip:**    init            0   nil

0   isa    NSObject   isa                    0   nil

### 3.5.6 +new

**Tip:**    NSObject    new            alloc    init

Objective-C    alloc    init        retain        new

### 3.5.7 API

**Tip:**    “    kitchen-sink ”    API

C++    Objective-C                    -                    Objective-C                    API

```

// GTMFoo.m
#import "GTMFoo.h"

@interface GTMFoo (PrivateDelegateHandling)
- (NSString *)doSomethingWithDelegate; // Declare private method
@end

@implementation GTMFoo(PrivateDelegateHandling)
...
- (NSString *)doSomethingWithDelegate {
    // Implement this method
}
...
@end
    
```

Objective-C 2.0                    @interface                    @implementation  
 @implementation  
 Objective-C 2.0

```

@interface GMFoo () { ... }
    
```

                  @implementation  
 “ ”                    Bug  
 Objective-C                    @implementation                    “middle truncation”                    NSString

### 3.5.8 #import and #include

**Tip:** #import Objective-C/Objective-C++ #include C/C++

```

        #import #include
    • Objective-C Objective-C++ #import
    • C C++ #include #define
Objective-C #define #import Objective-C Objective-C #import
Objective-C C C++ C C++ C C++ #import #include Objective-C
#include
        Mac C C++ #define Mac #import #include
#include

```

```

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

```

### 3.5.9

**Tip:** #import

Cocoa Foundation #import #include Objective-C

```

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

```

### 3.5.10 autorelease

**Tip:** autorelease release

release return

```

// 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 ivals

```
- (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

---

**Tip:** dealloc @interface

---

dealloc retained  
dealloc retained @interface dealloc

### 3.5.14 setter NSStrings

---

**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;
    NSError *exception = [NSError exceptionWithName:@"foo"
                                                reason:@"bar"
                                                userInfo:nil];
    @throw exception;
}

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

(continues on next page)

(continued from previous page)

```
@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 documen-
tation				
C/C++	NULL“	C/C++	C/C++	

### 3.5.17 BOOL

Tip: BOOL BOOL YES

Objective-C	BOOL	BOOL	YES` `(1)	` `NO` `(0)	` `BOOL	BOOL
	NO	BOOL	YES NO	256	256 512 ...	
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;
}
- (BOOL)isValid {
    return [self stringValue] != nil;
}
- (BOOL)isEnabled {
```

(continues on next page)

(continued from previous page)

```
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

---

<b>Tip:</b>	Property	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;
```

(continues on next page)

(continued from previous page)

```

@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

---

#### Tip:

---

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

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

### 3.5.20 synthesize

---

**Tip:**    iOS            synthesize

---

```
synthesize        @synthesize var = var_;        self.var = blah;        var = blah;
synthesize CType CType    @dynamic        CType retain        retain release        getter setter
                 @dynamic
```

```
// 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

---

1.        delegate\_
2.            delegate setDelegate:
3. delegate\_        retain

### 3.6.2 / / MVC

---

**Tip:**            API @protocol

---

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





## 4.1

2.6

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

guoqiao v2.19  
xuxinkun v2.59  
captainfffsama v2.6

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

## 4.2

Python [Google](#) [python](#)  
[Vim](#) [Emacs](#) [yapf](#)

## 4.3 Python

### 4.3.1 Lint

---

**Tip:** `pylintrc` `pylint`

---

```
: pylint Python bug . C C++ ( : less dynamic) , bug . Python , . . .
:
: pylint . , : a) b) c) , d) .
: pylint. , . :
```

```
dict = 'something awful' # Bad Idea... pylint: disable=redefined-builtin
```

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

```
pylint --list-msgs pylint . pylint --help-msg=C6409 , .
pylint: disable-msg , pylint: disable .
del . del , "Unused" , :
```

```
def viking_cafe_order(spam, beans, eggs=None):
    del beans, eggs # Unused by vikings.
    return spam + spam + spam
```

```
'_ ' , unused_ , _ .
```

### 4.3.2

---

**Tip:** , 'typing'

---

```
:
: . x.Obj Obj x .
: . , .
:
1. import x .
2. from x import y , x , y .
3. from x import y as z , 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)
```

```
typing six.moves
```

### 4.3.3

Tip:

```
:
:
:
:
```

yes:

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

FLAGS = absl.flags.FLAGS
```

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

FLAGS = flags.FLAGS
```

No: ( *jodie.py* *doctor/who/* )

```
#
#     sys.path.
import jodie
```

```
sys.path           sys.path           import jodie           jodie           jodie.py
```

### 4.3.4

Tip:

```
:
:
:
:
:
```

1. `ValueError`, `assert` API, `assert`, `assert, raise`, :

Yes:

```
def connect_to_next_port(self, minimum):
    """Connects to the next available port.

    Args:
        minimum: A port value greater or equal to 1024.

    Returns:
        The new minimum port.

    Raises:
        ConnectionError: If no available port is found.
    """
    if minimum < 1024:
        # Note that this raising of ValueError is not mentioned in the
        # string's "Raises:" section because it is not appropriate to
        # guarantee this specific behavioral reaction to API misuse.
        raise ValueError(f'Min. port must be at least 1024, not {minimum}')
    port = self._find_next_open_port(minimum)
    if not port:
        raise ConnectionError(
            f'Could not connect to service on port {minimum} or higher.')
    assert port >= minimum, (
        f'Unexpected port {port} when minimum was {minimum}.')
    return port
```

No:

```
def connect_to_next_port(self, minimum):
    """Connects to the next available port.

    Args:
        minimum: A port value greater or equal to 1024.

    Returns:
        The new minimum port.
    """
    assert minimum >= 1024, 'Minimum port must be at least 1024.'
    port = self._find_next_open_port(minimum)
    assert port is not None
    return port
```

2. `Exception`, `Error`.
3. `except:`, `Exception` `StandardError`, `( )`, `Python`, `except:` `Python` `except:` `bug`.
4. `try/except` `try`, `try/except`.
5. `finally` `try`, `finally`.

### 4.3.5

---

#### Tip:

---

```

:
:
:
: MAX_HOLY_HANDGRENADE_COUNT = 3.
:
:
:
:

```

### 4.3.6 / /

---

#### Tip:

---

```

:
:
: (pickled).
:
:
:
:

```

### 4.3.7 &

---

#### Tip:

---

```

: , & , map(), filter(), lambda.( : , () )
:
:
:
: : , for , . for .

```

Yes:

```

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)
          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)
]

```

(continues on next page)

(continued from previous page)

```

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')

```

No:

```

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)
: , , . . .
: ( , has_key() ).
: , , , , , , dict.iter*() python2 .

```

Yes:

```

for key in adict: ...
if key not in adict: ...
if obj in alist: ...
for line in afile: ...
for k, v in dict.iteritems(): ...

```

No:

```

for key in adict.keys(): ...
if not adict.has_key(key): ...
for line in afile.readlines(): ...

```

### 4.3.9

---

#### Tip:

---

```

: , (yield) , , . , .
: , , . , .
: .
: . "Yields:" "Returns:".
( : )

```

### 4.3.10 Lambda

---

#### Tip:

---

```

: , lambda . map() filter() .
: .
: . lambda , .
: . 60-80 , ( ) .
operator lambda . , operator.mul , lambda x, y: x * y .

```

### 4.3.11

---

#### Tip:

---

```

: ( ) if . : x = 1 if cond else 2 .
: if .
: if .
: . ,if ,else . if .

```

```

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')

```



```

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

```

### 4.3.12

#### Tip:

```

:         , , def foo(a, b = 0): . foo , b 0. , b .
:         , ( ) . , Python , " " .
:         , ( ), .
:         :

```

```

Yes: def foo(a, b=None):
      if b is None:
          b = []
Yes: def foo(a, b: Optional[Sequence] = None):
      if b is None:
          b = []
Yes: def foo(a, b: Sequence = ()): # Empty tuple OK since tuples are immutable

```

```

No: def foo(a, b=[]):
    ...
No: def foo(a, b=time.time()): # The time the module was loaded???
    ...
No: def foo(a, b=FLAGS.my_thing): # sys.argv has not yet been parsed...
    ...
No: def foo(a, b: Mapping = {}): # Could still get passed to unchecked code
    ...

```

### 4.3.13 (properties)

( : fluent python. “property” ””, “attribute” . python ”(attribute)”,  
” (property)”.)

**Tip:** , , . (properties) .

```

:         , (attribute) .
:         (attribute) get set , . Pythonic . , . (properties) .

```

```

: (properties) get set , : set get (properties) ( @property ). object .
. . ( : , @property )
: , . @property .
, . , ( ).

```

Yes:

```

import math

class Square:
    """A square with two properties: a writable area and a read-only perimeter.

    To use:
    >>> sq = Square(3)
    >>> sq.area
    9
    >>> sq.perimeter
    12
    >>> sq.area = 16
    >>> sq.side
    4
    >>> sq.perimeter
    16
    """

    def __init__(self, side):
        self.side = side

    @property
    def area(self):
        """Area of the square."""
        return self._get_area()

    @area.setter
    def area(self, area):
        return self._set_area(area)

    def _get_area(self):
        """Indirect accessor to calculate the 'area' property."""
        return self.side ** 2

    def _set_area(self, area):
        """Indirect setter to set the 'area' property."""
        self.side = math.sqrt(area)

    @property
    def perimeter(self):
        return self.side * 4

```

( : , , ?)

#### 4.3.14 True/False

**Tip:** false

: Python false. , " " false. 0 None, [], {}, "" false.

: Python . , .

: C/C++ , .

: false, : if foo: if foo != []: . :

1. None , is is not. None . false! ( : is id()),  
 , CPython , id)

2. == false . if not x: . false None, if not x and x is not None: .

3. ( , , ), false. if not seq: if seq: if len(seq): if not len(seq): .

4. , false ( None 0 ). ( len() ) 0 .

Yes:

```

if not users:
    print('no users')

if foo == 0:
    self.handle_zero()

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

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

No:

```

if len(users) == 0:
    print 'no users'

if foo is not None and not foo:
    self.handle_zero()

if not i % 10:
    self.handle_multiple_of_ten()

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

5. '0'( ) true.

### 4.3.15

**Tip:** . apply(). , for filter(), map() reduce().

: Python .  
 : Python , .

```
Yes: words = foo.split(':')

[x[1] for x in my_list if x[2] == 5]

map(math.sqrt, data)    # Ok. No inlined lambda expression.

fn(*args, **kwargs)
```

```
No: words = string.split(foo, ':')

map(lambda x: x[1], filter(lambda x: x[2] == 5, my_list))

apply(fn, args, kwargs)
```

### 4.3.16 (Lexical Scoping)

#### Tip:

: Python , . , . Python , . global ,  
 .  
 :

```
def get_adder(summand1):
    """Returns a function that adds numbers to a given number."""
    def adder(summand2):
        return summand1 + summand2

    return adder
```

```
( : , : sum = get_adder(summand1)(summand2) )
```

: , . Lisp Scheme( Haskell, ML ) .  
 : bug. PEP-0227 :

```
i = 4
def foo(x):
    def bar():
        print i,
        # ...
        # A bunch of code here
        # ...
    for i in x: # Ah, i *is* local to Foo, so this is what Bar sees
        print i,
    bar()
```

```
foo([1, 2, 3])    1 2 3 3, 1 2 3 4.
( : x , for x i. i , foo i , bar() . C++ .)
```

: .

### 4.3.17

**Tip:** , , `staticmethod` `` ```classmethod`.

: ( @ ). @classmethod @staticmethod, . , . , my\_decorator ,  
:

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

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

: . , (enforce invariants), .

: , . , . . .

: , . . python . . .

( , socket, ), ( pydoc ).

” ”: Main .

API `staticmethod` .

`classmethod` .

### 4.3.18

**Tip:** .

Python , (: `__hash__` `__eq__` Python ) . ( ).

Queue Queue . , `threading` (locking primitives). , `threading.Condition`

### 4.3.19

**Tip:**

: Python , , (metaclasses), , (on-the-fly compilation), , (object reparenting),  
(import hacks), , (modification of system internals), .

: , .

```

: """ , . ( ), , .
:
    abc.ABCMeta, collection.namedtuple, dataclasses, "enum".

```

### 4.3.20 python: python3 from \_\_future\_\_ imports

**Tip:** python3, python3 . .

```

: python3 python , python2.7 , , python3 .
: , python3 .
:
: from __future__ imports

```

```

    from __future__ import , , :
```

```

from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

```

```

    absolute imports , division behavior, print function .
    python3 , , , from __future__ .
    unicode_literals , , python2.7 b u unicode .
    six, future, past
    python2 python3 , six , future , past .

```

### 4.3.21

**Tip:** PEP-484 python3 , pytype . , pyi . , , pyi .

```

: :

```

```

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

```

```

    PEP-526 :
```

```

a: SomeType = some_func()

```

```

    python :
```

```

a = some_func() #type: SomeType

```

```

: . , .
: . .
: python . API , pytype . python , ( ) . ,
  TODO ,
  ( : IDE vim )

```

## 4.4 Python

### 4.4.1

---

**Tip:** `url`, `url`.

---

### 4.4.2

---

**Tip:** 80

---

:

- 1.
2. URL,
3. `url`

1. Pylint `. "# pylint: disable=invalid-name`

with `.`

Python `,`, `.`, `.`

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

    if (width == 0 and height == 0 and
        color == 'red' and emphasis == 'strong'):
```

`,` `:`

```
x = ('This will build a very long long '
     'long long long long long long string')
```

URL

```
Yes: # See details at
     # http://www.example.com/us/developer/documentation/api/content/v2.0/csv_file_name_
     ↪ extension_full_specification.html
```

```
No: # See details at
     # http://www.example.com/us/developer/documentation/api/content/\
     # v2.0/csv_file_name_extension_full_specification.html
```

with `.` with.

```
Yes: 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)
```

---

```
No: with VeryLongFirstExpressionFunction() as spam, \
      VeryLongSecondExpressionFunction() as beans:
      PlaceOrder(eggs, beans, spam, beans)
```

```
Yes: with very_long_first_expression_function() as spam:
      with very_long_second_expression_function() as beans:
      place_order(beans, spam)
```

```
;
      80 yapf      80      80 .
```

### 4.4.3

---

#### Tip:

---

```
Yes: if foo:
      bar()
      while x:
          x = bar()
      if x and y:
          bar()
      if not x:
          bar()
      # For a 1 item tuple the ()s are more visually obvious than the comma.
      onesie = (foo,)
      return foo
      return spam, beans
      return (spam, beans)
      for (x, y) in dict.items(): ...
```

```
No: if (x):
      bar()
      if not(x):
          bar()
      return (foo)
```

### 4.4.4

---

#### Tip: 4

---

```
tab, tab . , ( ), 4 ( ):
```

```
Yes: # Aligned with opening delimiter
      foo = long_function_name(var_one, var_two,
                               var_three, var_four)
```

(continues on next page)



(continued from previous page)

```
# Aligned with opening delimiter in a dictionary
foo = {
    long_dictionary_key: value1 +
                        value2,
    ...
}

# 4-space hanging indent; nothing on first line
foo = long_function_name(
    var_one, var_two, var_three,
    var_four)

# 4-space hanging indent in a dictionary
foo = {
    long_dictionary_key:
        long_dictionary_value,
    ...
}
```

```
No: # Stuff on first line forbidden
foo = long_function_name(var_one, var_two,
    var_three, var_four)

# 2-space hanging indent forbidden
foo = long_function_name(
    var_one, var_two, var_three,
    var_four)

# No hanging indent in a dictionary
foo = {
    long_dictionary_key:
        long_dictionary_value,
    ...
}
```

#### 4.4.5

---

**Tip:** ], ), } . YAPF .

---

```
Yes: golomb3 = [0, 1, 3]
Yes: golomb4 = [
    0,
    1,
    4,
    6,
]
```

No: `golomb4 = [`  
`0,`  
`1,`  
`4,`  
`6`  
`]`

#### 4.4.6

Tip: `,`

`,` `,` `,` `,` `,` `,` `,` `,` `,` `,`

#### 4.4.7

Tip:

Yes: `spam(ham[1], {eggs: 2}, [])`

No: `spam( ham[ 1 ], { eggs: 2 }, [ ] )`

`,` `,` `,` `( )`.

Yes: `if x == 4:`  
`print(x, y)`  
`x, y = y, x`

No: `if x == 4 :`  
`print(x , y)`  
`x , y = y , x`

Yes: `spam(1)`

no: `spam (1)`

Yes: `dict['key'] = list[index]`

No: `dict ['key'] = list [index]`

`,` `(=)`, `(==, <, >, !=, <>, <=, >=, in, not in, is, is not)`, `(and, or, not)`.

Yes: `x == 1`

No: `x<1`

`=`, `.`, `,`, `=`, `.`

Yes: `def complex(real, imag=0.0): return magic(r=real, i=imag)`

Yes: `def complex(real, imag: float = 0.0): return Magic(r=real, i=imag)`

No: `def complex(real, imag = 0.0): return magic(r = real, i = imag)`

No: `def complex(real, imag: float=0.0): return Magic(r = real, i = imag)`

`,` `(` `:` `#` `,` `=` `)`:

Yes:

```
foo = 1000 # comment
```

```
long_name = 2 # comment that should not be aligned
```

```
dictionary = {
```

```
    "foo": 1,
```

```
    "long_name": 2,
```

```
}
```

No:

```
foo      = 1000 # comment
```

```
long_name = 2    # comment that should not be aligned
```

```
dictionary = {
```

```
    "foo"      : 1,
```

```
    "long_name": 2,
```

```
}
```

## 4.4.8 Shebang

---

**Tip:** `.py` `#!` `.` PEP-394, `main` `#!/usr/bin/python2` `#!/usr/bin/python3` `.`

---

( `:` `,` `,` Shebang ( Hashbang) `(#!)`, `.` Shebang , Unix Shebang , `,` `,` Shebang `.` , `#!/bin/sh` `/bin/sh` .)

`#!` Python , `,` `.` `#!` .

## 4.4.9

---

**Tip:** `,` `,`

---

```

Python : """( PEP-257 ). : __doc__ , pydoc . ( pydoc ,
). : """( PEP-257 ). : , ( ). . ;
.
.
. ( , Apache 2.0, BSD, LGPL, GPL), .

```

```

"""A one line summary of the module or program, terminated by a period.

```

```

Leave one blank line. The rest of this docstring should contain an
overall description of the module or program. Optionally, it may also
contain a brief description of exported classes and functions and/or usage
examples.

```

```

Typical usage example:

```

```

foo = ClassFoo()
bar = foo.FunctionBar()
"""

```

```

, , , .
, :

```

- 1.
- 2.
- 3.

```

, . , " ", . , , , . , .
See base class . , .

```

```

2 .

```

```

Args: , , 80 , 2 4 ( ).
*foo( ) **bar ( ), *foo **bar.

```

```

Returns: ( Yields: ) . None, .

```

```

Raises: .

```

```

def fetch_smalltable_rows(table_handle: smalltable.Table,
                           keys: Sequence[Union[bytes, str]],
                           require_all_keys: bool = False,
) -> Mapping[bytes, Tuple[str]]:
    """Fetches rows from a Smalltable.

```

```

Retrieves rows pertaining to the given keys from the Table instance
represented by table_handle. String keys will be UTF-8 encoded.

```

```

Args:

```

```

    table_handle: An open smalltable.Table instance.
    keys: A sequence of strings representing the key of each table
    row to fetch. String keys will be UTF-8 encoded.
    require_all_keys: Optional; If require_all_keys is True only

```

(continues on next page)

(continued from previous page)

rows with values set for all keys will be returned.

**Returns:**

A dict mapping keys to the corresponding table row data fetched. Each row is represented as a tuple of strings. For example:

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

Returned keys are always bytes. If a key from the keys argument is missing from the dictionary, then that row was not found in the table (and require\_all\_keys must have been False).

**Raises:**

`IOError`: An error occurred accessing the smalltable.

```
"""
```

Args:       :

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

    Retrieves rows pertaining to the given keys from the Table instance
    represented by table_handle. String keys will be UTF-8 encoded.

    Args:
        table_handle:
            An open smalltable.Table instance.
        keys:
            A sequence of strings representing the key of each table row to
            fetch. String keys will be UTF-8 encoded.
        require_all_keys:
            Optional; If require_all_keys is True only rows with values set
            for all keys will be returned.

    Returns:
        A dict mapping keys to the corresponding table row data
        fetched. Each row is represented as a tuple of strings. For
        example:

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

    Returned keys are always bytes. If a key from the keys argument is
    missing from the dictionary, then that row was not found in the
    table (and require_all_keys must have been False).
```

(continues on next page)

(continued from previous page)

```

Raises:
IOError: An error occurred accessing the smalltable.
"""

```

```

, (Attributes), (Attributes) .

```

```

class SampleClass(object):
    """Summary of class here.

    Longer class information...
    Longer class information...

    Attributes:
        likes_spam: A boolean indicating if we like SPAM or not.
        eggs: An integer count of the eggs we have laid.
    """

    def __init__(self, likes_spam=False):
        """Inits SampleClass with blah."""
        self.likes_spam = likes_spam
        self.eggs = 0

    def public_method(self):
        """Performs operation blah."""

```

```

# We use a weighted dictionary search to find out where i is in
# the array. We extrapolate position based on the largest num
# in the array and the array size and then do binary search to
# get the exact number.

```

```

if i & (i-1) == 0:          # True if i is 0 or a power of 2.

```

```

, 2 .
, Python,

```

```

# BAD COMMENT: Now go through the b array and make sure whenever i occurs
# the next element is i+1

```

#### 4.4.10 ,

Tip:

```

, . , , .

```

## 4.4.11

---

**Tip:** `object` . `( python2 )`

---

```
Yes: class SampleClass(object):
    pass

class OuterClass(object):

    class InnerClass(object):
        pass

class ChildClass(ParentClass):
    """Explicitly inherits from another class already."""
```

```
No: class SampleClass:
    pass

class OuterClass:

    class InnerClass:
        pass
```

`object` (properties) , `PEP-3000` . `__new__` , `__init__` , `__delattr__` , `__getattr__` , `__setattr__` , `__hash__` , `__repr__` , and `__str__` .

## 4.4.12

---

**Tip:** `%` . `+` `%` .

---

```
Yes: x = a + b
x = '%s, %s!' % (imperative, expletive)
x = '{}, {}'.format(imperative, expletive)
x = 'name: %s; score: %d' % (name, n)
x = 'name: {}; score: {}'.format(name, n)
```

```
No: x = '%s%s' % (a, b) # use + in this case
x = '{}{}'.format(a, b) # use + in this case
x = imperative + ', ' + expletive + '!'
x = 'name: ' + name + '; score: ' + str(n)
```

`+=` . `join` . ( `cStringIO` . )





### 4.4.13 sockets

**Tip:** sockets , .

- , sockets , , :
- 1. , . , .
- 2. .
- 3. sockets, . , , .
- , , sockets , , . :
- 1. . Python , .
- 2. , ( , ).
- “with” :

```
with open("hello.txt") as hello_file:
    for line in hello_file:
        print line
```

”with” , contextlib.closing():

```
import contextlib

with contextlib.closing(urllib.urlopen("http://www.python.org/")) as front_page:
    for line in front_page:
        print line
```

Legacy AppEngine Python 2.5 ”with” , from \_\_future\_\_ import with\_statement .

### 4.4.14 TODO

**Tip:** TODO , . , .

TODO ”TODO” , , email . . , . TODO , ( ).  
 TODO . TODO, .

```
# TODO(kl@gmail.com): Use a "*" here for string repetition.
# TODO(Zeke) Change this to use relations.
```

TODO ” ” , (“2009 11 ”) (“ XML ”).

### 4.4.15

**Tip:** , typing

```
Yes: import os
      import sys
      from typing import Mapping, Sequence
```

```
No: import os, sys
```

```
1. __future__
```

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
```

```
1.
```

```
import sys
```

```
1.
```

```
import tensorflow as tf
```

```
1.
```

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

```
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

# Older style code may have these imports down here instead:
#from myproject.backend.hgwells import time_machine
#from myproject.backend.state_machine import main_loop
```

#### 4.4.16

**Tip:**

---

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

Yes:

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

No:

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

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

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

### 4.4.17

---

**Tip:** Python, , , . , (property) .  
( : , : ! , . Pythonic )

---

, , , get\_foo() set\_foo() . (property) , , , ,

### 4.4.18

---

**Tip:** : module\_name ; : package\_name ; : ClassName ; : method\_name ; : ExceptionName ; : function\_name ; : GLOBAL\_CONSTANT\_NAME ; : global\_var\_name ; : instance\_var\_name ; : function\_parameter\_name ; : local\_var\_name . , , , , . .py , .

---

1. , , try/except e, with f.
  2. / (-)
  3. (Python , \_\_init\_\_)
- 
1. " (Internal)" , , .
  2. ( ) protected ( from module import \* ).
  3. ( ) .

4. . Java, .
5. ( CapWords, Pascal ), ( lower\_with\_under.py). CapWords.py ,

python .py - . , exec "\$0.py" "\$@" bash .

### Python Guido

Type	Public	Internal
Modules	lower_with_under	__lower_with_under
Packages	lower_with_under	
Classes	CapWords	__CapWords
Exceptions	CapWords	
Functions	lower_with_under()	__lower_with_under()
Global/Class Constants	CAPS_WITH_UNDER	__CAPS_WITH_UNDER
Global/Class Variables	lower_with_under	__lower_with_under
Instance Variables	lower_with_under	__lower_with_under (protected) or __lower_with_under (private)
Method Names	lower_with_under()	__lower_with_under() (protected) or __lower_with_under() (private)
Function/Method Parameters	lower_with_under	
Local Variables	lower_with_under	

### 4.4.19 Main

**Tip:** , . (main functionality) , . main() .

Python , pydoc . if \_\_name\_\_ == '\_\_main\_\_', .  
 absl, app.run :

```

from absl import app
...

def main(argv):
    # process non-flag arguments
    ...

if __name__ == '__main__':
    app.run(main)
    
```

```

, :

def main():
    ...

if __name__ == '__main__':
    main()
    
```

· , , pydoc .

### 4.4.20

**Tip:** , ,

· 40 , . , , bug. , .  
 , , .

### 4.4.21

1. 'PEP-484 <<https://www.python.org/dev/peps/pep-0484/>>'
2. self cls
3. Any
4.
  1. API
  - 2.
  - 3.
  - 4.
  5. .

```
def my_method(self,
              first_var: int,
              second_var: Foo,
              third_var: Optional[Bar]) -> int:
    ...
```

```
· , , ·
def my_method(self, first_var: int) -> int:
    ...
```

```
, , , 4 .
def my_method(
    self, first_var: int) -> Tuple[MyLongType1, MyLongType1]:
    ...
```

, , 4 , ) def

```

Yes:
def my_method(
    self, other_arg: Optional[MyLongType]
) -> Dict[OtherLongType, MyLongType]:
...

```

```
pylint ) ( , .
```

```

No:
def my_method(self,
              other_arg: Optional[MyLongType]
) -> Dict[OtherLongType, MyLongType]:
...

```

```

def my_method(
    self,
    first_var: Tuple[List[MyLongType1],
                    List[MyLongType2]],
    second_var: List[Dict[
        MyLongType3, MyLongType4]]) -> None:
...

```

```
, alias. “” 4 .
```

```

Yes:
def my_function(
    long_variable_name:
        long_module_name.LongTypeName,
) -> None:
...

```

```

No:
def my_function(
    long_variable_name: long_module_name.
        LongTypeName,
) -> None:
...

```

```

class MyClass:

    def __init__(self,
                 stack: List["MyClass"]) -> None:

```

PEP-008, = .

```
Yes:
def func(a: int = 0) -> int:
    ...
```

```
No:
def func(a:int=0) -> int:
    ...
```

### NoneType

python ,NoneType “ ”, ,None NoneType . None, . Union, , Optional. Optional. PEP-484 a: Text = None a: Optional[Text] = None, , .

```
Yes:
def func(a: Optional[Text], b: Optional[Text] = None) -> Text:
    ...
def multiple_nullable_union(a: Union[None, Text, int]) -> Text
    ...
```

```
No:
def nullable_union(a: Union[None, Text]) -> Text:
    ...
def implicit_optional(a: Text = None) -> Text:
    ...
```

, . , “\_” . , :

```
_ShortName = module_with_long_name.TypeWithLongName
ComplexMap = Mapping[Text, List[Tuple[int, int]]]
```

```
# type: ignore . pytype ( lint):
```

```
# pytype: disable=attribute-error
```

```
, :
```

```
# type::
```

```
a = SomeUndecoratedFunction() # type: Foo
```

```
, :
```

```
a: Foo = SomeUndecoratedFunction()
```

### Tuples vs Lists

Lists . Tuples , . ( : , python ,list tuple , , list tuple )

```
a = [1, 2, 3] # type: List[int]
b = (1, 2, 3) # type: Tuple[int, ...]
c = (1, "2", 3.5) # type: Tuple[int, Text, float]
```

## TypeVars

python . TypeVars.

```
from typing import List, TypeVar
T = TypeVar("T")
...
def next(l: List[T]) -> T:
    return l.pop()
```

TypeVar

```
AddableType = TypeVar("AddableType", int, float, Text)
def add(a: AddableType, b: AddableType) -> AddableType:
    return a + b
```

typing AnyStr. bytes, unicode .

```
from typing import AnyStr
def check_length(x: AnyStr) -> AnyStr:
    if len(x) <= 42:
        return x
    raise ValueError()
```

python . python3 , str. Text . , . python2 ,  
Text. , str . unicode, python3 . , python , "str" .

```
No:
def py2_code(x: str) -> unicode:
    ...
```

, bytes.

```
def deals_with_binary_data(x: bytes) -> bytes:
    ...
```

python2 "str" "unicode", python3 str.

```
from typing import Text
...
def py2_compatible(x: Text) -> Text:
    ...
def py3_only(x: str) -> str:
    ...
```

, Union , .

```
from typing import Text, Union
...
```

(continues on next page)



(continued from previous page)

```
def py2_compatible(x: Union[bytes, Text]) -> Union[bytes, Text]:
...
def py3_only(x: Union[bytes, str]) -> Union[bytes, str]:
...
```

```
, AnyStr. python3
```

```
typing , typing , :
```

```
from typing import Any, Dict, Optional
```

```
, typing , , "import x as y" :
```

```
from typing import Any as AnyType
```

```
, if TYPE_CHECKING:
1. string, python3.6 . python3.6 ,
2. , , ,
3.
4.
5. ,
```

```
import typing
if typing.TYPE_CHECKING:
    import sketch
def f(x: "sketch.Sketch"): ...
```

```
, Any . alias
, (Any Any, ).
```

```
from typing import Any

some_mod = Any # some_mod.py imports this module.
...

def my_method(self, var: "some_mod.SomeType") -> None:
...
```

```
, Any .
```

```
def get_names(employee_ids: List[int]) -> Dict[int, Any]:
...
```

```
# These are both interpreted as get_names(employee_ids: List[Any]) -> Dict[Any,
↔ Any]
def get_names(employee_ids: list) -> Dict:
...

def get_names(employee_ids: List) -> Dict:
...
```

Any , . , TypeVar .

```
def get_names(employee_ids: List[Any]) -> Dict[Any, Text]:
    """Returns a mapping from employee ID to employee name for given IDs."""
```

```
T = TypeVar('T')
def get_names(employee_ids: List[T]) -> Dict[T, Text]:
    """Returns a mapping from employee ID to employee name for given IDs."""
```

## 4.5

```
, , . , . , .
, " ", " "
```

Revision 2.60

Amit Patel  
 Antoine Picard  
 Eugene Jhong  
 Gregory P. Smith  
 Jeremy Hylton  
 Matt Smart  
 Mike Shields  
 Shane Liebling



**Contents**

- *Shell* -

## 5.1

1.26

Paul Armstrong

Bean Zhang v1.26

- Google Style Guide
- Google -

## 5.2

### 5.2.1 Shell

---

**Tip:** Bash shell

---

```
#!/bin/bash      set  shell  bash <script_name>
shell bash      shell
                Solaris SVR4    Bourne shell
```

## 5.2.2 Shell

---

**Tip:** Shell

---

Shell

- shell
- shell
- \${PHESTATUS} Python
- 100 Python Shell

## 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

---

```
shell      SUID/SGID shell      bash  SUID
          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.
```

### 5.5.2

---

**Tip:**

---

- 
- 
- 
- 

```
#!/bin/bash
#
# Perform hot backups of Oracle databases.

export PATH='/usr/xpg4/bin:/usr/bin:/opt/csw/bin:/opt/goog/bin'

#####
```

(continues on next page)

(continued from previous page)

```
# Cleanup files from the backup dir
# Globals:
#   BACKUP_DIR
#   ORACLE_SID
# Arguments:
#   None
# Returns:
#   None
#####
cleanup() {
    ...
}
```

### 5.5.3

---

**Tip:**

---

### 5.5.4 TODO

---

**Tip:** TODO

---

C++  
TODOs      TODO                  TODO      bug ticket

```
# TODO(mrmonkey): Handle the unlikely edge cases (bug ####)
```

## 5.6

### 5.6.1

---

**Tip:**

---

### 5.6.2

---

**Tip:** 80

---







- 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
- 
- 
- [[
- \$@ \$\*

```
# '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 "#, @$"
```



```

# 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:

- ./ \* \*

```

# 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'
removed `somefile'

# As opposed to:
psa@bilby$ rm -v ./ *

```

(continues on next page)

(continued from previous page)

```
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
    total+="${count}"
    last_file="${filename}"
```

(continues on next page)

(continued from previous page)

```
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}"
```

shell while “ ” awk shell

```
# 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

---

                              ::                    Google

```
# Single function
my_func() {
  ...
}

# Part of a package
mypackage::my_func() {
  ...
}
```

( )    function

### 5.8.2

---

**Tip:**

---

```
for zone in ${zones}; do
  something_with "${zone}"
done
```

### 5.8.3

Tip:

```
# Constant
readonly PATH_TO_FILES='/some/path'

# Both constant and environment
declare -xr ORACLE_SID='PROD'
```

getopts    getopts                    declare            readonly export

```
VERBOSE='false'
while getopts 'v' flag; do
  case "${flag}" in
    v) VERBOSE='true' ;;
  esac
done
readonly VERBOSE
```

### 5.8.4

Tip:

Google    maketemplate    make\_template    make-template

### 5.8.5

Tip:    readonly    declare -r

shell

```
zip_version="$(dpkg --status zip | grep Version: | cut -d ' ' -f 2)"
if [[ -z "${zip_version}" ]]; then
  error_message
else
  readonly zip_version
fi
```

### 5.8.6

Tip:    local



local

local

```
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

### 5.9.1

---

Tip:

---

\$? if

```

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

```

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

PIPESTATUS

[ PIPESTATUS

```

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

**Tip:** shell

bash(1)

sed

```

# Prefer this:
addition=$(( ${X} + ${Y} ))
substitution="${string/#foo/bar}"

# Instead of this:
addition="$(expr ${X} + ${Y})"
substitution="$(echo "${string}" | sed -e 's/^foo/bar/')"

```

## 5.10

C++

## 6.1

Google JavaScript JavaScript

## 6.2 Javascript

### 6.2.1 var

var

var

var

document window

var

### 6.2.2

- NAMES\_LIKE\_THIS
- @const
- IE const

CONSTANT\_VALUE\_CASE

```
number string boolean
```

```
@const          const          const      IE      const
  @const
```

```
@const          CONSTANT_VALUE_CASE
```

```
/**
 *
 * @type {number}
 */
goog.example.TIMEOUT_IN_MILLISECONDS = 60;
```

```
1 60          @const
```

```
/**
 * Map of URL to response string.
 * @const
 */
MyClass.fetchedUrlCache_ = new goog.structs.Map();
```

### 6.2.3

```
// 1.
MyClass.prototype.myMethod = function() {
  return 42;
} // .

(function() {
  //
})();

var x = {
  'i': 1,
  'j': 2
} // .

// 2.  IE firefox  .
```

(continues on next page)

(continued from previous page)

```
//
[normalVersion, ffVersion][isIE]();

var THINGS_TO_EAT = [apples, oysters, sprayOnCheese] //

// 3.
-1 == resultOfOperation() || die();
```

```
1. js 42 42
2. "no sush property in undefined" x[normalVersion, ffVersion][isIE]()
3. die resultOfOperation() NaN THINGS_TO_EAT die()
```

```
js "}" "}" "}" "(" "{" "]" js
```

```
var foo = function() {
  return true;
}; //

function foo() {
  return true;
} //
```

## 6.2.4

## 6.2.5

```
if (x) {
  function foo() {}
}
```

```
if (x) {  
    var foo = function() {}  
}
```

## 6.2.6

## 6.2.7

hack

## 6.2.8

*string.charAt(3)*    *string[3]*    DOM

## 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

Javascript      class B      class D

Closure `goog.inherits()`

```
function D() {
  goog.base(this)
}
goog.inherits( D, B );

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 his.property_;
};
```

JavaScript

`if (key in obj)`



## 6.2.13

JS

DOM

```
function foo(element, a, b) {
  element.onclick = function() { /* a b */ };
}
```

a b

```
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

RPC

```
users = [
  {
    name: 'Eric',
    id: 37824,
    email: 'jellyvore@myway.com'
  },
  {
    name: 'xtof',
    id: 31337,
    email: 'b4d455h4x0r@google.com'
  },
  ...
];
```

eval()

RPC

XMLHttpRequest

RPC

JavaScript

```
var userOnline = false;
var user = 'nusrat';
var xmlhttp = new XMLHttpRequest();
xmlhttp.open('GET', 'http://chat.google.com/isUserOnline?user=' + user, false);
xmlhttp.send('');
//
// userOnline = true;
```

(continues on next page)

(continued from previous page)

```

if (xmlhttp.status == 200) {
    eval(xmlhttp.responseText);
}
// 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';

```

(continues on next page)

(continued from previous page)

```
printArray(a); //  
  
a = new Array;  
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();
```

2

```
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
};
```

## 6.2.21

Object.prototype Array.prototype

Function.prototype

## 6.2.22 Internet Explorer

```
var f = function () {
  /*@cc_on if (@_jscript) { return 2* @*/ 3; /*@ } @*/
};
```

JavaScript

## 6.3 Javascript

### 6.3.1

```
functionNamesLikeThis variableNamesLikeThis ClassNamesLikeThis EnumNamesLikeThis
methodNamesLikeThis CONSTANT_VALUES_LIKE_THIS foo.namespaceNamesLikeThis.bar
filenameslikethis.js
```

- 
- 

```
opt_
var_args var_args arguments
@param
```

### 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

“ ”      foo.hats.\*      foo.hats.\*

```
foo.require('foo.hats');
/**
 * --
 * @constructor
 * @extends {foo.hats.RoundHat}
 */
foo.hats.BowlerHat = function() {
};
```

API      API      API

```
foo.provide('googleyhats.BowlerHat');

foo.require('foo.hats');
/**
 * @constructor
 * @extends {foo.hats.RoundHat}
 */
googleyhats.BowlerHat = function() {
```

(continues on next page)

```
...
};
goog.exportSymbol('foo.hats.BowlerHat', googleyhats.BowlerHat);
```

```
/**
 * @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());
};
```

goog.scope

```
myapp.main = function() {
  var namespace = some.long.namespace;
  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

window                    window   window

### 6.3.5

C++

```
if (something) {  
  // ...  
} else {  
  // ...  
}
```



```
var arr = [1, 2, 3]; // [ ]
var obj = {a: 1, b: 2, c: 3}; // [ ]
```

```
//
var inset = {
  top: 10,
  right: 20,
  bottom: 15,
  left: 12
};

//
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'
];

//
goog.dom.createDom(goog.dom.TagName.DIV, {
  id: 'foo',
  className: 'some-css-class',
  style: 'display:none'
}, 'Hello, world!');
```

```
CORRECT_Object.prototype = {
  a: 0,
  b: 1,
  lengthyName: 2
};
```

```
WRONG_Object.prototype = {
  a      : 0,
  b      : 1,
  lengthyName: 2
};
```

80

80

```
// 80
//
```

(continues on next page)

(continued from previous page)

```

goog.foo.bar.doThingThatIsVeryDifficultToExplain = function(
    veryDescriptiveArgumentNumberOne, veryDescriptiveArgumentTwo,
    tableModelEventHandlerProxy, artichokeDescriptorAdapterIterator) {
    // ...
};

//
//
goog.foo.bar.doThingThatIsVeryDifficultToExplain = function(
    veryDescriptiveArgumentNumberOne,
    veryDescriptiveArgumentTwo,
    tableModelEventHandlerProxy,
    artichokeDescriptorAdapterIterator) {
    // ...
};

//      80
//
function foo(veryDescriptiveArgumentNumberOne, veryDescriptiveArgumentTwo,
    tableModelEventHandlerProxy, artichokeDescriptorAdapterIterator) {
    // ...
}

//
//
function bar(veryDescriptiveArgumentNumberOne,
    veryDescriptiveArgumentTwo,
    tableModelEventHandlerProxy,
    artichokeDescriptorAdapterIterator) {
    // ...
}

```

4

```

if (veryLongFunctionNameA(
    veryLongArgumentName) ||
    veryLongFunctionNameB(
    veryLongArgumentName)) {
    veryLongFunctionNameC(veryLongFunctionNameD(
        veryLongFunctionNameE(
            veryLongFunctionNameF)));
}

```

function

```

prefix.something.reallyLongFunctionName('whatever', function(a1, a2) {
    if (a1.equals(a2)) {
        someOtherLongFunctionName(a1);
    } else {

```

(continues on next page)

(continued from previous page)

```

    andNowForSomethingCompletelyDifferent(a2.parrot);
  }
});

var names = prefix.something.myExcellentMapFunction(
  verboselyNamedCollectionOfItems,
  function(item) {
    return item.name;
  });

```

**goog.scope**

goog.scope the Closure Library

```

    goog.scope
    goog.scope(function() {          goog.provide  goog.require          scope  //
goog.scope
C++      goog.scope      0

```

```

goog.scope(function() {
var Button = goog.ui.Button;

Button = function() { ... };
...

```

```

goog.provide('my.module');

goog.require('goog.dom');
goog.require('goog.ui.Button');

goog.scope(function() {
var Button = goog.ui.Button;
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.getElement('my-button'));
};
...
}); // goog.scope

```

4 2

```

someWonderfulHtml = '' +
    getEvenMoreHtml(someReallyInterestingValues, moreValues,
                    evenMoreParams, 'a duck', true, 72,
                    slightlyMoreMonkeys(0xfff)) +
    '';

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();
}

```

```

doSomethingTo(x);
doSomethingElseTo(x);
andThen(x);

nowDoSomethingWith(y);

andNowWith(z);

```

```

,

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.

```

(continues on next page)

(continued from previous page)

```
var z = a ?
    moreComplicatedB :
    moreComplicatedC;
```

```
var x = foo.bar().
    doSomething().
    doSomethingElse();
```

### 6.3.6

```
delete   typeof   void           return   throw   case   in   new
```

### 6.3.7

```
' '
' ' HTML
```

```
var msg = 'This is some HTML';
```

### 6.3.8

```
@private @protected JSDoc
JSDoc @private @protected
--jscomp_warning=visibility
@private
@private @private instanceof
@protected
```

```
// 1
// AA_PrivateClass_ AA_init_

/**
 * @private
 * @constructor
 */
AA_PrivateClass_ = function() {
};

/** @private */
function AA_init_() {
    return new AA_PrivateClass_();
}
```

(continues on next page)

(continued from previous page)

```

}
AA_init_();

```

```

@private

```

```

@protected

```

```

C++ JAVA private protected

```

```

C++

```

```

// 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.

/**
 * @constructor
 * @extends {AA_PublicClass}
 */
AA_SubClass = function() {
  // Legal access of a protected static property.
  AA_PublicClass.staticProtectedProp = this.method();
};

```

(continues on next page)

(continued from previous page)

```
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

	JavaScript 5 {null} {undefined} {boolean} {number} {string}		
	{Object} {Function} {EventTarget} EventTar- get null	@constructor JSDoc @interface JSDoc	
	{goog.events.EventType} goog.events. EventType	, @enum JSDoc ES4	
	{Array.<string>} {Object.<string, number>}	Java	
	{(number boolean)}	A B  {number boolean} {function(): (number boolean)}	{(number, boolean)}  {(number  boolean)}
	{?number}	syntactic sugar	{number? }
	{!Object}	null	{Object! }
	{myNum: number, myObject}	myNum number myObject length Array.<{length}>	
	{function(string, boolean)}		
	{function(): number}		
this	{function(this:goog.ui. Menu, string)} goog.ui.Menu		
new	{function(new:goog.ui. Menu, string)} "new" goog.ui.Menu		
	{function(string, ... [number]): number}		
@param	@param {...number} var_args		
	{function(?string=, number=)}	"_"	
@param	@param {number=} opt_argument number		
	{*}		
	{?}		



## JavaScript

number	<code>1</code> <code>1.0</code> <code>-5</code> <code>1e5</code> <code>Math.PI</code>	
Number	<code>new Number(true)</code>	Number
string	<code>'Hello'</code> <code>"World"</code> <code>String(42)</code>	
String	<code>new String('Hello')</code> <code>new String(42)</code>	String
boolean	<code>true</code> <code>false</code> <code>Boolean(0)</code>	Boolean
Boolean	<code>new Boolean(true)</code>	Boolean
RegExp	<code>new RegExp('hello')</code> <code>/world/g</code>	
Date	<code>new Date</code> <code>new Date()</code>	
null	<code>null</code>	
undefined	<code>undefined</code>	
void	<code>function f() {</code> <code>  return;</code> <code>}</code>	
Array	<code>['foo', 0.3, null]</code> <code>[]</code>	
Array.<number>	<code>[11, 22, 33]</code>	
<b>156</b>		<b>Chapter 6. Javascript -</b>
Array.<Array.<string>>	<code>[['one', 'two', 'three</code> <code>↵'], ['foo', 'bar']]</code>	

```
/** @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.
```

(continues on next page)

(continued from previous page)

```

    * @type {Object|undefined}
    * @private
    */
    this.myValue_ = opt_value;
}

```

```

    myValue_      null    undefined
:   opt_value    {Object=}  {Object|undefined}    undefined    undefined

```

```

/**
 *
 * @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      this      this

```
/**
 * @param {function(this:T, ...)} fn
 * @param {T} thisObj
 * @param {...*} var_args
 * @template T
 */
goog.bind = function(fn, thisObj, var_args) {
  ...
};
//
goog.bind(function() { this.someProperty; }, new SomeClass());
// this
goog.bind(function() { this.someProperty; });
```

### 6.3.10

JSDoc

c++

JSDoc

//

JSDoc   JavaDoc   JSDoc

```
/**
 * A JSDoc comment should begin with a slash and 2 asterisks.
 * Inline tags should be enclosed in braces like {@code this}.
 * @desc Block tags should always start on their own line.
 */
```

JSDoc

```
/**
 * Illustrates line wrapping for long param/return descriptions.
 * @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;
};
```

@fileoverview

```
/**
 * 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;
};
```

## JSDoc HTML

JavaDoc , JSDoc HTML <code> <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

```
/**
 * 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);
```

```
/**
 * 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) {
  // ...
}
```

```
/** @constructor */
project.MyClass = function() {
  /**
   * Maximum number of things per pane.
   * @type {number}
   */
  this.someProperty = 4;
}
```

## JSDoc

<p>@author</p>	<pre> @author          user- name@google.com (first last)  /**  * @fileoverview  * ↪Utilities for  * ↪handling textareas.  * @author kuth@google.  * ↪com (Uthur Pendragon)  */         </pre>	<p>@fileoverview</p>
<p>@code</p>	<pre> {@code ...}  /**     • Moves to the       next position       in the selec-       tion.     • Throws       {@code       goog.iter.StopIteration}       when it     • passes the       end of the       range.     • @return       {Node} The       node at       the next       position.  * / goog.dom.RangeIterator.prototype.next = function() {     // ... };         </pre>	

Continued on next page

Table 1 – continued from previous page

<pre>@const</pre>	<pre>@const @const {type}  /** @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>	<pre>@const          js                 @const                 @const</pre>
<pre>@constructor</pre>	<pre>@constructor  /**  * A rectangle.  * @constructor  */ function GM_Rect() {   ... }</pre>	
<pre>@define</pre>	<pre>@define {Type} description  /** @define {boolean} */ var TR_FLAGS_ENABLE_ ↳DEBUG = true;  /** @define {boolean} */ goog.userAgent.ASSUME_ ↳IE = false;</pre>	<pre>--define='goog.userAgent. ASSUME_IE=true'          goog.userAgent. ASSUME_IE    true</pre>

Continued on next page



Table 1 – continued from previous page

<p>@deprecated</p>	<p>@deprecated Description</p> <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>	
<p>@dict</p>	<p>@dict Description</p> <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>	<p>( Foo) @dict Foo</p>
<p>@enum</p>	<p>@enum {Type}</p> <pre> /**  * Enum for tri-state  ↪values.  * @enum {number}  */ project.TriState = {   TRUE: 1,   FALSE: -1,   MAYBE: 0 }; </pre>	

Continued on next page

Table 1 – continued from previous page

<pre>@export</pre>	<pre>@export  /** @export */ foo.MyPublicClass. ↳prototype. ↳myPublicMethod = ↳function() {   // ... };</pre>	<pre>--generate_exports  goog.exportSymbol('foo.MyPublicClass. ↳prototype.myPublicMethod',   foo.MyPublicClass.prototype. ↳myPublicMethod);    @export   1. //javascript/closure/base.js ,   2.     goog.exportSymbol      goog.      exportProperty</pre>
<pre>@expose</pre>	<pre>@expose  /** @expose */ MyClass.prototype.exposedProperty = 3;</pre>	<pre>@expose</pre>
<pre>@extends</pre>	<pre>@extends Type @extends {Type}  /**   • Immutable   empty node   list.   •   @constructor   • @extends   goog.ds.BasicNodeList   */ <b>goog.ds.EmptyNodeList = function() {</b>   ... };</pre>	<pre>@constructor</pre>
<pre>@externs</pre>	<pre>@externs  /**   •   @fileoverview   This is an   externs file.   • @externs   */ var document;</pre>	

Continued on next page

Table 1 – continued from previous page

<p>@fileoverview</p>	<p>@fileoverview Description</p> <pre> /**   *   *   •   *     @fileoverview   *     Utilities for   *     doing things   *     that require   *     this very   *     long   *   • but not in-   *     dented com-   *     ment.   *   • @author   *     kuth@google.com   *     (Uthur Pen-   *     dragon)   * / </pre>	
<p>@implements</p>	<p>@implements Type @imple- ments {Type}</p> <pre> /**   *   *   • A shape.   *   • @interface   * / function Shape() {} Shape.prototype.draw = function() {}; /**   *   *   • @constructor   *   • @implements   *     {Shape}   * / function Square() {} Square.prototype.draw = function() {   ... }; </pre>	<p>@constructor</p>
<p>@inheritDoc</p>	<p>@inheritDoc</p> <pre> /** @inheritDoc */ project.SubClass.   ↳prototype.toString() {   // ... }; </pre>	<p>@override</p> <p>@inheritDoc</p> <p>@override</p>

Continued on next page

Table 1 – continued from previous page

<pre>@interface</pre>	<pre>@interface  /**  * A shape.  * @interface  */ function Shape() {}; Shape.prototype.draw =   ↪function() {};  /**  * A polygon.  * @interface  * @extends {Shape}  */ function Polygon() {}; Polygon.prototype   ↪getSides = function()   ↪{};</pre>	
<pre>@lends</pre>	<pre>@lends objectName @lends {objectName}  goog.object.extend(   Button.prototype,   /** @lends {Button.   ↪prototype} */ {     isButton:   ↪function() { return   ↪true; }   });</pre>	<pre>“ ” @type {Foo} Foo @lends {Foo} “Foo ”. JSDoc Toolkit docs</pre>
<pre>@license or @pre- serve</pre>	<pre>@license Description  /**  * @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>	<pre>@licenseor @preserve</pre>

Continued on next page

Table 1 – continued from previous page

@noalias	<pre>@noalias  /** @noalias */ function Range() {}</pre>	
@nosideeffects	<pre>@nosideeffects  /** @nosideeffects */ function ↳noSideEffectsFn1() {   // ... }; /** @nosideeffects */ var noSideEffectsFn2 = ↳function() {   // ... }; /** @nosideeffects */ a.prototype. ↳noSideEffectsFn3 = ↳function() {   // ... };</pre>	
@override	<pre>@override  /**  * @return {string} ↳Human-readable ↳representation of ↳project.SubClass.  * @override  */ project.SubClass. ↳prototype.toString() {   // ... };</pre>	

Continued on next page

Table 1 – continued from previous page

<p>@param</p>	<p>@param {Type} varname Description</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>@private</p>	<p>@private @private {type}</p> <pre> /**  * Handlers that are  *   ↪ listening to this  *   ↪ logger.  * @private {!Array.  *   ↪ &lt;Function&gt;}  */ this.handlers\_ = []; </pre>	<p>@private</p>

Continued on next page

Table 1 – continued from previous page

<p>@protected</p>	<pre> @protected @protected {type}  /**  * 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#Visibility__private_and_protected_fields</a>&gt;“</p>
<p>@return</p>	<pre> @return {Type} Description  /**  * @return {string} The  * ↪hex ID of the last  * ↪item.  */ goog.Baz.prototype.   ↪getLastId =   ↪function() {     // ...     return id;   };         </pre>	<p>“ ” “ true false” @return</p>
<p>@see</p>	<pre> @see Link  /**  * Adds a single item,  * ↪recklessly.  * @see #addSafely  * @see goog.Collect  * @see goog.  * ↪RecklessAdder#add  * ...         </pre>	

Continued on next page

Table 1 – continued from previous page

<p>@struct</p>	<p>@struct Description</p> <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>	<p>Foo      Foo      @struct      -</p>
<p>@supported</p>	<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>	
<p>@suppress</p>	<p>@suppress {warn- ing1 warning2}</p> <pre> /**  * @suppress {deprecated}  */ function f() {   ␣ ↳deprecatedVersionOfF(); ↳ } </pre>	<p> </p>

Continued on next page



Table 1 – continued from previous page

<pre>@template</pre>	<pre>@template  /**  * @param  * ↪{function(this:T, ...  * ↪)} fn  * @param {T} thisObj  * @param {...*} var_args  * @template T  */ goog.bind = function(fn,  * ↪ thisObj, var_args) {  *     ...  * };</pre>	
<pre>@this</pre>	<pre>@this Type @this {Type}  pinto.chat.RosterWidget.  * ↪extern(  * ↪'getRosterElement',  /**  * Returns the roster  * ↪widget element.  * @this pinto.chat.  * ↪RosterWidget  * @return {Element}  */ function() {  *     return this.  *     ↪getWrappedComponent_  *     ↪().getElement();  * };</pre>	<pre>this</pre>
<pre>@type</pre>	<pre>@type Type @type {Type}  /**  *  * • The message  *   hex ID.  * • @type  *   {string}  * */ var hexId = hexId;</pre>	

Continued on next page

Table 1 – continued from previous page

@typedef	@typedef	
	<pre> /** @typedef ↳{(string number)} */ goog.NumberLike; /** @param {goog. ↳NumberLike} x A↳ ↳number or a string. */ goog.readNumber =↳ ↳function(x) {     ... } </pre>	

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

```
goog.provide('namespace.MyClass');
```

```
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');
```

```
goog.provide('foo.bar');  
goog.provide('foo.bar.method');  
goog.provide('foo.bar.CONSTANT');
```

### 6.3.12

JS      Closure Compiler

### 6.3.13

JavaScript

#### True False

false

- null
- undefined
- ''
- 0

true

- "0"
- []
- {}

```
while (x != null) {
```

```
    x 0    false
```

```
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

```
if (val != 0) {
    return foo();
} else {
    return bar();
}
```

```
return val ? foo() : bar();
```

HTML

```
var html = '<input type="checkbox" ' +
    (isChecked ? ' checked' : '') +
    (isEnabled ? '' : ' disabled') +
    ' name="foo">';
```

&& ||

,  
“||” ‘default’

```

/** @param {*=} opt_win */
function foo(opt_win) {
  var win;
  if (opt_win) {
    win = opt_win;
  } else {
    win = window;
  }
  // ...
}

```

```

/** @param {*=} opt_win */
function foo(opt_win) {
  var win = opt_win || window;
  // ...
}

```

“&&”

```

if (node) {
  if (node.kids) {
    if (node.kids[index]) {
      foo(node.kids[index]);
    }
  }
}
}

```

```

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]);

```

length    O(n)    length    O(n<sup>2</sup>)

```

var paragraphs = document.getElementsByTagName('p');
for (var i = 0; i < paragraphs.length; i++) {

```

(continues on next page)

(continued from previous page)

```
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);  
}
```