

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

Google [?][?][?][?][?][?][?][?]

Mar 15, 2025

CONTENTS

1	Google ??????????—????	1
2	C++ ????? - ?????	3
2.1	??	3
2.2	1. ???	6
2.3	2. ???	10
2.4	3. ?	18
2.5	4. ??	25
2.6	5. ?? Google ???	27
2.7	6. ?? C++ ??	29
2.8	7. ?????	46
2.9	8. ??	51
2.10	9. ??	57
2.11	10. ?????	70
2.12	11. ???	71
3	Objective-C ????? - ?????	73
3.1	Google Objective-C Style Guide ???	73
3.2	??????	76
3.3	??	80
3.4	??	83
3.5	Cocoa ? Objective-C ??	85
3.6	Cocoa ??	94
4	Python ????? - ?????	97
4.1	??	97
4.2	??	97
4.3	Python?????	98
4.4	Python?????	112
4.5	?????	135
5	Shell ????? - ?????	137
5.1	??	137
5.2	??	137
5.3	Shell??????????	138
5.4	??	139
5.5	??	139
5.6	??	141
5.7	??????	145
5.8	?????	149

5.9	????	151
5.10	??	153
6	Javascript ????? - ?????	155
6.1	0. ??	155
6.2	??	155
6.3	Javascript????	155
6.4	Javascript????	165
7	TypeScript ?????	201
7.1	0. ??	201
7.2	??	201
7.3	????	202
7.4	????	208
7.5	????	225
7.6	????	230
7.7	???	238
8	HTML/CSS ????? - ?????	239
8.1	0. ??	239
8.2	??	239
8.3	???????	239
8.4	???????	240
8.5	???????	241
8.6	HTML????	241
8.7	HTML????	244
8.8	css????	245
8.9	CSS?????	249
8.10	CSS????	251
8.11	??	252
9	Java ????? - ?????	253
9.1	0. ??	253
9.2	1. ??	253
9.3	2. ?????	254
9.4	3. ?????	255
9.5	4. ??	256
9.6	5. ??	265
9.7	6. ?????	267
9.8	7. Javadoc	268

GOOGLE 开发风格指南——中文版

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

 Note

注意

开发风格指南 Google 开发风格指南

开发风格指南 Google 开发风格指南 Google Style Guide

开发风格指南 8 个部分包括：

“开发风格指南”包含“开发风格指南camelCase”和“开发风格指南”等“开发风格指南”

开发风格指南 Google 开发风格指南 Google 开发风格指南

开发风格指南 8 个部分包括：

1. [Google C++ 开发风格指南](#)
2. [Google Objective-C 开发风格指南](#)
3. [Google Python 开发风格指南](#)
4. [Google Shell 开发风格指南](#)
5. [Google JavaScript 开发风格指南](#)
6. [Google TypeScript 开发风格指南](#)
7. [Google HTML/CSS 开发风格指南](#)
8. [Google Java 开发风格指南](#)

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

- [开发风格指南 cpplint](#) —— [开发风格指南 google-c-style.el](#) Google 开发风格指南 Emacs 开发风格指南
- [开发风格指南 XML Document Format Style Guide](#) 开发风格指南 Yang.Y

C++ [\[?\]\[?\]\[?\]\[?\]](#) - [\[?\]\[?\]\[?\]\[?\]](#)

Contents

- C++ [\[?\]\[?\]\[?\]\[?\]](#) - [\[?\]\[?\]\[?\]\[?\]](#)

2.1 [\[?\]\[?\]](#)

[\[?\]\[?\]\[?\]\[?\]](#)

2024/04/12

[\[?\]\[?\]\[?\]](#)

Benjy Weinberger
Craig Silverstein
Gregory Eitzmann
Mark Mentovai
Tashana Landray

[\[?\]\[?\]](#)

[YuleFox](#)

[Yang.Y](#)

[acgyrant](#)

[lilinsanity](#)

[\[?\]\[?\]](#)

[\[?\]\[?\]\[?\]\[?\]](#)

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

2.1.1 2022

Google 2022, 2022. Google 2022, 2022. Google 2022.

2022:

2022. 2022, 2022, 2022. 2022, 2022, 2022, 2022.

Google 2022, 5 2022, 2022. 2022. 2022.

2022, 2022, 2022, 2022. 2022, 2022, 2022 Google 2022, 2022. 2022, 2022, 2022.

2022, 2022. 2022.

2022, 2022, bug 2022. 2022, 2022.

2022, 2022 Artistic License/GPL 2022.

2022:

- 2015-08 : 2022 @lilinsanity 2022 Google CPP Style Guide 4.45 2022
- 2015-07 4.45 : acgyrant 2022 C++ 2022 C++ 2022 C++11 2022 Google 2022 C++ 2022 innocentim, 2022 farseerfc 2022 Arch Linux 2022 C++ Primer 2022
- 2009-06 3.133 : YuleFox 2022 1.0 2022, 2022, 2022. Yang.Y 2022 YuleFox 2022, 2022 : Google 2022 - 2022. 2022 3.133 2022, 2022, 2022. Yang.Y 2022, YuleFox 2022.
- 2008-07 1.0 : 2022 YuleFox 2022 Blog, 2022.

2022.

2.1.2 2022

C++ 2022. C++ 2022, 2022 (feature), 2022, 2022, 2022.

2022 C++ 2022. 2022, 2022 C++ 2022.

2022 (style, 2022 (readability)) 2022 C++ 2022. “2022” 2022, 2022.

2022.

2022: 2022 C++ 2022, 2022 C++.

2.1.3 2022 2022

2022 2022

2022 2022 2022 2022. 2022 2022 2022 2022. 2022 2022, 2022 2022, 2022 2022 2022 2022 2022 2022. 2022 2022 2022 2022 2022 2022 (2022 2022), 2022 2022 2022 2022 2022 2022.

2022 2022 2022 2022:

2022 2022

2022 2022 2022 2022, 2022 2022 2022 2022. 2022 2022 2022 2022 2022 2022, 2022 2022 2022, 2022 2022, 2022 2022 2022 2022. 2022 2022 2022 2022 2022 2022. 2022, goto 2022 2022, 2022 2022 2022, 2022 2022 2022.

2022, 2022 2022

2022 2022 (2022 2022) 2022 2022. 2022, 2022 2022 2022 2022 2022 2022. 2022 2022 2022. “2022 2022” 2022 2022. 2022 2022 2022 (2022 2022), 2022 2022 2022 2022 2022 (2022 std::unique_ptr 2022 2022 2022).

2022 2022

2022 2022 2022 2022, 2022 2022 (2022) 2022. 2022 2022 2022: 2022 2022 2022 #include 2022, 2022 2022 2022 2022 2022. 2022, 2022 “2022” 2022 2022 “2022 2022”; 2022, 2022 2022 2022 2022. 2022, 2022 2022 2022. 2022 2022 2022 2022 2022. 2022 2022 2022 (2022, 2022 2022 2022). 2022 2022 2022 2022, 2022 2022. 2022 2022 2022 2022 2022 2022.

2022 C++ 2022

2022 2022 2022 2022, 2022 2022 2022 2022. 2022 C++ 2022 2022 2022, 2022 2022 2022, 2022 2022 2022. 2022, 2022 2022 2022, 2022 2022 2022 2022 2022. 2022 (2022 2022) 2022 2022 2022. 2022, 2022 C++ 2022, 2022 2022 2022 2022. 2022 2022 2022 2022 2022 2022.

2022 2022 2022 2022

2022 C++ 2022 2022 2022. 2022 2022 2022 2022 2022 2022. 2022 2022 2022 2022 2022.

2022 2022 2022 C++ 2022 2022 2022 2022

2022 C++ 2022 2022, 2022 2022. 2022 2022, 2022 2022 2022 2022 2022 2022 2022 2022. 2022, 2022 2022 2022. 2022 2022 2022, 2022 2022 2022 2022: 2022 2022 2022 2022 2022, 2022 2022 2022.

2022 2022

2022 2022 2022 2022 2022 2022 2022 2022 2022 2022 2022. 2022, 2022 2022 2022 (global namespace): 2022 2022 2022 2022 2022 (name collision), 2022 2022.

2022 2022

2022 2022 2022 2022 2022 2022.

2022 2022 2022 2022 2022 2022. 2022 2022 2022 2022 C++ 2022 2022 2022 2022 2022. 2022 2022 2022 2022 2022: 2022 “2022 2022”. 2022 2022. 2022, 2022, 2022 2022 2022.


2.1.4 C++

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

2.2 1.

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


2.2.1 1.1.

 Tip

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

, (header guards, 1.2. #define) .
(inline function) (template), (instantiate) , (definition). (-inl.h) ; , .cc , (explicit) , .cc (definition), .
, .inc ., , .

2.2.2 1.2. #define

 Tip

#define : <>_<>_<>_H_ .

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

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

2.2.3 1.3. `__attribute__((symbol, include, declaration, definition))`

Tip

`__attribute__((symbol, include, declaration, definition))`. `__attribute__((symbol, include, declaration, definition))`.

Example: `#include "foo.h", #include "bar.h", #include "foo.h", #include "bar.h", #include "foo.h", #include "bar.h".`

2.2.4 1.4. `__attribute__((forward_declaration, definition))`

Tip

`__attribute__((forward_declaration, definition))`.

Example:

`__attribute__((forward_declaration, definition))`.

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

Example:

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

Example:

- `__attribute__((forward_declaration, definition))`.
- `#include "foo.h", #include "bar.h", #include "foo.h", #include "bar.h", #include "foo.h", #include "bar.h".`
- `(library) __attribute__((forward_declaration, definition))` API, `(widening)`, `__attribute__((forward_declaration, definition))`, `__attribute__((forward_declaration, definition))`.
- `std::` `(undefined behavior)`.
- `__attribute__((forward_declaration, definition))`, `#include "foo.h", #include "bar.h", #include "foo.h", #include "bar.h".`

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

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

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

4. `foo.h`
5. C++ `foo.h` (`foo.h`), `#include <algorithm>` `#include <stddef.h>`.
6. `foo.h`
7. `foo.h` `foo.h`.
8. `foo.h`
9. `foo.h` `foo.h`.

`foo.h`.

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

`dir/foo.cc` `foo` `dir2/foo2.h` `foo` (`base/basicctypes_unittest.cc` `foo` `base/basicctypes.h`), `foo`.

`foo` C `foo` (`stddef.h`) `foo` C++ `foo` (`stddef`) `foo`. `foo`, `foo`.

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

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

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

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

#include <string>
#include <vector>

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

`foo`

























`foo` (`system-specific`) `foo` (`conditional include`) `foo`. `foo`,

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

















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

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

2.2.7 (YuleFox)


1. `int main() { return 0; }`
2. `int main() { return 0; }`
3. `int main() { return 0; }`
4. `-inl.h  (:D);`
5. `int main() { return 0; }` (` `);
6. `int main() { return 0; }` . ` .. `, ` `, ` `, ` `, ` `, ` `, ` `, ` `, ` `, ` `.















2.2.8 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

`  (namespace) .   .  using  ( using namespace foo).  (inline) .   (unnamed namespace) .`

``:
` (global scope)   (name collision).`

``:
` .`
`  Foo  (class),  (symbol) .`
` project1::Foo  project2::Foo  .`
` `

```

namespace outer {
inline namespace inner {
    void foo();
} // namespace inner
} // namespace outer
```

` outer::inner::foo()  outer::foo() .` ` ABI `

??:

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

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

????, ????????????????? (fully-qualified name) ??????. ?????????????????????.

??:

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

- ?? ?????? ??.
- ?????????, ????????????? (???: ?????????????).
- ?????? gflags ??/???????????????????????? (forward declaration) ??, ?????????????????????:

```
// .h ??
namespace mynamespace {

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

} // namespace mynamespace
```

```
// .cc ??
namespace mynamespace {

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

} // namespace mynamespace
```

????? .cc ?????????, ?????? (flag) ? using ??.

```
#include "a.h"

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

namespace mynamespace {

using ::foo::Bar;

...????????????... // ??????.

} // namespace mynamespace
```

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

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

- foo.cc, foo.h, foo API. foo:

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

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

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

- foo.
- foo “internal”, foo API.

```
// Absl foo.
using ::absl::container_internal::ImplementationDetail;
```

- foo, foo.

foo: foo

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

2.3.2 2.2. foo

 Tip

foo.cc, foo (unnamed namespace) foo static, foo (internal linkage). foo.h foo.

foo:

```
foo static foo. foo.
foo, foo.
```

foo:

```
foo.cc foo. foo.h foo.
foo, foo:
```

```
namespace {
...
} // namespace
```

2.3.3 2.3. [C++ Namespaces](#)

Tip

`using` (nonmember) `using namespace`; `using namespace` (completely global function). `using namespace` (static member) `using namespace` (class). `using namespace`.

Tip:

`using namespace`. `using namespace`, `using namespace`.

Tip:

`using namespace`, `using namespace`.

Tip:

`using namespace`. `using namespace`. `using namespace`, `using namespace`. `using namespace`; `using namespace`, `using namespace`. `using namespace` .cc `using namespace`, `using namespace`.

2.3.4 2.4. [C++ Scope](#)

Tip

`using namespace` (scope), `using namespace`.

`using namespace` C++ `using namespace`. `using namespace`, `using namespace`. `using namespace`, `using namespace`. `using namespace`, `using namespace`, `using namespace`.

```
int i;
i = f(); // using namespace.
```

```
int i = f(); // using namespace
```

```
int jobs = NumJobs();
// using namespace...
f(jobs); // using namespace.
```

```
int jobs = NumJobs();
f(jobs); // using namespace (using namespace) using namespace.
```

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


constexpr

constexpr (constant expression); constexpr, constexpr.
 constexpr (constexpr int) constexpr, constexpr.
 constexpr constexpr.

```
const int kNum = 10; //

struct X { int n; };
const X kX[] = {{1}, {2}, {3}}; //

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

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

```
// :.
const std::string kFoo = "foo";

// , kBar ().
const std::string& kBar = StrCat("a", "b", "c");

void bar() {
    // :.
    static std::map<int, int> kData = {{1, 0}, {2, 0}, {3, 0}};
}
```

, , , . , , static T& t = *new T; ;

constexpr

constexpr, constexpr, constexpr (initializer) .

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

constexpr, constexpr.

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

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

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

. constexpr constinit .

constexpr:

```
// .
time_t time(time_t*); // constexpr!
```

(continues on next page)

(continued from previous page)

```
int f(); // constexpr!
struct Bar { Bar() {} };

// constexpr.
time_t m = time(nullptr); // constexpr.
Foo y(f()); // constexpr.
Bar b; // constexpr Bar::Bar() constexpr.
```

constexpr. constexpr, constexpr, constexpr, constexpr, constexpr. constexpr.

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

constexpr (constexpr).

constexpr

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

2.3.6 2.6. thread_local

Tip

constexpr (compile-time constant) constexpr thread_local constexpr, constexpr ABSL_CONST_INIT constexpr. constexpr thread_local, constexpr.

constexpr:

constexpr thread_local constexpr:

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

constexpr. constexpr, constexpr. thread_local constexpr constexpr. constexpr, constexpr, constexpr, constexpr.

thread_local constexpr, constexpr thread_local constexpr, constexpr. constexpr thread_local constexpr. constexpr thread_local constexpr, constexpr (constexpr).

thread_local: thread_local, thread_local (C++). thread_local thread_local, (use-after-free,) .

:

- (race) (), thread_local .
- , thread_local .

:

- thread_local , .
- thread_local . , .
- , thread_local , .
- (data member) thread_local.
- thread_local , . , () thread_local . .
- / thread_local. , , , thread_local , . , thread_local , .

:

thread_local (). ABSL_CONST_INIT constexpr , .

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

thread_local , . , thread_local , thread_local :

```
Foo& MyThreadLocalFoo() {
    thread_local Foo result = ComplicatedInitialization();
    return result;
}
```

, thread_local . () thread_local , . thread_local .

thread_local , .

2.3.7 (YuleFox)

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

2.3.8 acgyrant

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

2.4 3.

(class) C++ . C++ .

2.4.1 3.1.

Tip

(constructor) (virtual method).

-
- const ,

- (dispatch) (implementation).
- , ()
- . IsValid() ,
- ,

. , . 42 Init() (factory function). Init() , , .

2.4.2 3.2.

Tip

. explicit (keyword).

double int

operator bool().

explicit (cast)

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

void Func(Foo f);

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

explicit

??

-
- (overload), string_view, std::string const char*
-

??

-
-
-
- explicit, explicit
- (call-site ambiguity),
-

??

explicit. initializer_list explicit (copy-initialization, MyType m = {1, 2};).

2.4.3 3.3. `std::move`

 Tip

When using `std::move`, you must use a variable of a movable type.

??

`std::move` (movable type) `std::move`.

```
std::unique_ptr<int> p1(1);
std::unique_ptr<int> p2(std::move(p1)); // OK
std::string s1("hello");
std::string s2(std::move(s1));
```

Constructors, copy constructor, copy-assignment operator, move constructor, move-assignment operator, etc.

Pass by value, pass by reference.

??

API, mutability, pass by reference, composition.

```
std::move(p1), Clone(), CopyFrom(), Swap()
= default
```

rvalue, lvalue.

??

Cleanup, object slicing.

std::move, std::move_cast.

??

```
public
Copyable(const Copyable& other) = default;
Copyable& operator=(const Copyable& other) = default;
```

```
class Copyable {
public:
    Copyable(const Copyable& other) = default;
    Copyable& operator=(const Copyable& other) = default;
    // ...
};
```

(continues on next page)

(continued from previous page)

```
class MoveOnly {
public:
    MoveOnly(MoveOnly&& other) = default;
    MoveOnly& operator=(MoveOnly&& other) = default;

    // Deleted, disabled:
    MoveOnly(const MoveOnly&) = delete;
    MoveOnly& operator=(const MoveOnly&) = delete;
};

class NotCopyableOrMovable {
public:
    // Deleted.
    NotCopyableOrMovable(const NotCopyableOrMovable&) = delete;
    NotCopyableOrMovable& operator=(NotCopyableOrMovable&) = delete;

    // Deleted, disabled:
    NotCopyableOrMovable(NotCopyableOrMovable&&) = delete;
    NotCopyableOrMovable& operator=(NotCopyableOrMovable&&) = delete;
};
```


Deleted, disabled:

- Deleted private (deleted), deleted/default.
- Deleted, deleted. Deleted, deleted.
- Deleted, deleted, deleted. Deleted.

Deleted, deleted, deleted. Deleted, deleted. Deleted (default) deleted. Deleted.

Deleted, deleted (abstract) deleted. Deleted, deleted protected, deleted (pure virtual) deleted. Deleted (concrete class).

2.4.4 3.4. Deleted

 **Tip**

Deleted struct deleted. Deleted class.

C++ deleted class deleted. Deleted, deleted.

Deleted, deleted. Deleted. Deleted (invariant) deleted, deleted. Deleted, deleted.

Deleted, deleted, deleted. Deleted, deleted.

Deleted STL deleted, deleted (trait) deleted (template metafunction) deleted (functor) deleted.

Deleted, deleted deleted.

2.4.5 3.5. `std::pair` and `std::tuple`

 Tip

`std::pair`, `std::tuple` (pair) `std::tuple` (tuple).

`std::pair`, `std::tuple`, `std::pair` `.first`, `.second` `std::get<X>` `std::get<Type>`, `std::get`, `std::get`, `std::get`, `std::get`, `std::get`.

`std::pair`, `std::tuple`. `std::pair` API `std::tuple` API.

2.4.6 3.6. `std::is_base_of`

 Tip

`std::is_base_of`, `std::is_base_of` (composition) `std::is_base_of` (inheritance) `std::is_base_of` `public`.

`std::is_base_of`

`std::is_base_of`, `std::is_base_of`. `std::is_base_of` (interface inheritance) `std::is_base_of` (pure abstract base class, `std::is_base_of`) `std::is_base_of` (implementation inheritance).

`std::is_base_of`

`std::is_base_of`, `std::is_base_of`. `std::is_base_of`, `std::is_base_of`. `std::is_base_of` API. `std::is_base_of` API `std::is_base_of`, `std::is_base_of`.

`std::is_base_of`

`std::is_base_of`, `std::is_base_of`, `std::is_base_of`. `std::is_base_of` (override) `std::is_base_of`, `std::is_base_of`.

`std::is_base_of`, `std::is_base_of` (`std::is_base_of`, `std::is_base_of`). `std::is_base_of`, `std::is_base_of` (diamond inheritance) `std::is_base_of`, `std::is_base_of`.

`std::is_base_of`

`std::is_base_of` `public` `std::is_base_of`. `std::is_base_of`, `std::is_base_of`. `std::is_base_of`, `std::is_base_of` `final`.

`std::is_base_of`. `std::is_base_of` (composition) `std::is_base_of`. `std::is_base_of` “is-a”, YuleFox `std::is_base_of` “has-a” `std::is_base_of` `std::is_base_of`: `std::is_base_of` `std::is_base_of`.

`std::is_base_of` `protected`. `std::is_base_of`, `std::is_base_of`.

`std::is_base_of` `override` `std::is_base_of` `final` (`std::is_base_of`) `std::is_base_of`. `std::is_base_of` `virtual` `std::is_base_of`. `std::is_base_of` `std::is_base_of` `std::is_base_of`, `std::is_base_of`, `std::is_base_of`, `std::is_base_of`. `std::is_base_of`, `std::is_base_of`, `std::is_base_of`, `std::is_base_of`.

`std::is_base_of`, `std::is_base_of`.

2.4.7 3.7. 2022

Tip

2022 (overload). 2022 (user-defined literal).

2022

2022 operator 2022 2022, 2022. 2022 operator" 2022, 2022 (2022 operator bool()).

2022

2022, 2022. 2022 (2022 ==, <, =, <<). 2022, 2022.

2022

2022

- 2022. 2022.
- 2022.
- 2022.
- 2022, 2022.
- 2022, 2022 C++ 2022, 2022 grep 2022.
- 2022, 2022, 2022. 2022: foo < bar 2022 &foo < &bar 2022.
- 2022. 2022, 2022 & 2022, 2022. 2022 &&, || 2022, 2022. (2022: 2022, 2022 && 2022, 2022, 2022.)
- 2022, 2022. 2022 (link) 2022, 2022 (undefined behavior), 2022.
- 2022, 2022 std::string_view("Hello World") 2022 "Hello World"sv. 2022 C++ 2022. 2022, 2022.
- 2022, 2022 using 2022 (using-directive, 2022:ref:2022 <namespaces>) 2022 using 2022 (using-declaration, 2022, 2022). 2022, 2022.

2022

2022, 2022, 2022. 2022, 2022 | 2022/2022, 2022 shell 2022.

2022. 2022, 2022, .cc 2022. 2022, 2022, 2022. 2022, 2022. 2022, 2022.

2022. 2022, 2022, 2022. 2022 a + b 2022 b + a 2022, 2022.

2022 T, 2022 operator==, 2022 T 2022. 2022 T 2022, 2022 operator<=>, 2022 operator== 2022. 2022.

2022. 2022, 2022 ==, = 2022 << 2022 Equals(), CopyFrom() 2022 PrintTo(). 2022, 2022. 2022, 2022 std::set 2022, 2022 (comparator) 2022 <.

&&, ||, , unary) & . operator "", . (). = . (stream) << , . . , .

2.4.8 3.8.

Tip

(private), . (invariant) , (accessor) (const).

, Google Test .cc (test fixture class) (protected). .cc (.h), .

2.4.9 3.9.

Tip

(public) .

public: , protected:, private: . .

, , :

1. (typedef, using, enum, ,)
2. (,)
- 3.
4. (factory function)
- 5.
- 6.
7. (,)
8. ()

. , . .

2.4.10 (YuleFox)

1. ;
2. , , , ;
3. , explicit;
4. , private ;
5. struct;
6. > > > , virtual , ;

7. `protected, final, synchronized, volatile;`
8. `Interface final, synchronized, volatile, protected, final, final, final protected;`
9. `final, synchronized, final, synchronized;`
10. `final;`
11. `final: public -> protected -> private;`
12. `final, final, final;`

2.5 4. `final`

2.5.1 4.1. `final`

`final`

final keyword is used to declare final variables, methods, and classes.

`final`

C/C++ `final` keyword is used to declare final variables, methods, and classes.

`final` keyword is used to declare final variables, methods, and classes.

C/C++ `final` keyword is used to declare final variables, methods, and classes. `const` keyword is used to declare constant variables, methods, and classes. `std::optional` keyword is used to declare optional variables, methods, and classes. `const` keyword is used to declare constant variables, methods, and classes.

`const` keyword is used to declare constant variables, methods, and classes. `const` keyword is used to declare constant variables, methods, and classes. `const` keyword is used to declare constant variables, methods, and classes.

`const` keyword is used to declare constant variables, methods, and classes. `const` keyword is used to declare constant variables, methods, and classes. `const` keyword is used to declare constant variables, methods, and classes.

`const` keyword is used to declare constant variables, methods, and classes. `const` keyword is used to declare constant variables, methods, and classes. `const` keyword is used to declare constant variables, methods, and classes.

2.5.2 4.2. `final`

`final`

`final` keyword is used to declare final variables, methods, and classes.

`final`

`final` keyword is used to declare final variables, methods, and classes. `final` keyword is used to declare final variables, methods, and classes. `final` keyword is used to declare final variables, methods, and classes.

`final` keyword is used to declare final variables, methods, and classes. `final` keyword is used to declare final variables, methods, and classes. `final` keyword is used to declare final variables, methods, and classes. `bug` keyword is used to declare bug variables, methods, and classes.

`final` keyword is used to declare final variables, methods, and classes. `final` keyword is used to declare final variables, methods, and classes. `final` keyword is used to declare final variables, methods, and classes. `final` keyword is used to declare final variables, methods, and classes. `final` keyword is used to declare final variables, methods, and classes.

2.5.3 4.3. `string`

??

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

??

???????????????? const string& ??, ????????????????? const char* ??????????:

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

??

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

??

???????????????????? (acgtyrant ?????????????????), ????????? C++ ?????????, ?????????????????. ??, ?????????????????????, ?????????????????.

??

????????????????, ?????????????????????. ??, ? AppendString() ? AppendInt() ?, ????????????? Ap- pend(). ?????????????????????????????????????, ????????? std::vector ?????????????????????.

2.5.4 4.4. `string`

??

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

??

????????????????????, ?????????????????????. ?????????????????????, ???. ?????????, ?????????????????, ?????????????, ????????????? “????” ? “????”.

??

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

??, ???.

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

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

??

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

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

2.5.5 4.5. `constexpr`

??

constexpr (constexpr) constexpr.

??

C++ constexpr. constexpr. ??:

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

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

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

constexpr. ?? int constexpr, constexpr. constexpr, constexpr, constexpr, constexpr.

??

constexpr Lambda ?? constexpr. constexpr, constexpr Lambda constexpr, constexpr, constexpr.

constexpr, constexpr, constexpr, constexpr. ??:

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

constexpr:

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

??

constexpr, ?? C ? Java constexpr, constexpr.

constexpr, constexpr. constexpr. constexpr, constexpr, constexpr.

??

constexpr, constexpr, constexpr. constexpr (?? Lambda ??) constexpr. constexpr, constexpr, constexpr.

2.6 5. `constexpr` Google

Google `constexpr` / `constexpr` C++ constexpr, constexpr C++ constexpr.

2.6.1 5.1. `std::auto_ptr`

> `std::auto_ptr`

`std::auto_ptr<T>`, `std::auto_ptr<T>`.

> `std::auto_ptr`

`std::auto_ptr<T>`. `std::auto_ptr<T>`, `std::auto_ptr<T>`. `std::auto_ptr<T>`, `std::auto_ptr<T>`. `std::auto_ptr<T>`, `std::auto_ptr<T>`.

`std::auto_ptr<T>` * `ptr` -> `std::auto_ptr<T>`. `std::auto_ptr<T>`, `std::auto_ptr<T>`. `std::unique_ptr` `ptr` C++11 `std::auto_ptr<T>`, `std::auto_ptr<T>`; `std::unique_ptr` `ptr`, `std::auto_ptr<T>`. `std::unique_ptr` `ptr`, `std::auto_ptr<T>` `move(ptr)`. `std::shared_ptr` `ptr`, `std::auto_ptr<T>`, `std::auto_ptr<T>`; `std::auto_ptr<T>`, `std::auto_ptr<T>`, `std::auto_ptr<T>`.

> `std::auto_ptr`

- `std::auto_ptr<T>`, `std::auto_ptr<T>`.
- `std::auto_ptr<T>`, `std::auto_ptr<T>`, `std::auto_ptr<T>`.
- `std::auto_ptr<T>` "ptr" `std::auto_ptr<T>`, `std::auto_ptr<T>` `std::auto_ptr<T>`.
- `std::auto_ptr<T>`, `std::auto_ptr<T>`, `std::auto_ptr<T>`.
- `std::auto_ptr<T>` `ptr`, `std::auto_ptr<T>`, `std::auto_ptr<T>`.
- `std` `const ptr`, `std::auto_ptr<T>`, `std::auto_ptr<T>`.

> `std::auto_ptr`

- `std::auto_ptr<T>` `ptr`. `std::auto_ptr<T>`, `std::auto_ptr` API `std::auto_ptr<T>`, `std::auto_ptr`, `std::auto_ptr`, `std::auto_ptr<T>`.
- `std::auto_ptr<T>`, `std::auto_ptr<T>` `std::auto_ptr<T>`.
- `std` API `std::auto_ptr`, `std::auto_ptr<T>`.
- `std::auto_ptr`, `std::auto_ptr<T>`.
- `std::unique_ptr` `ptr` C++11 `std::auto_ptr` `ptr`, `std::auto_ptr`, `std::auto_ptr`.
- `std::auto_ptr<T>`, `std::auto_ptr<T>`, `std::auto_ptr<T>`.
- `std::auto_ptr<T>`, `std::auto_ptr`.
- `std::auto_ptr` (`ptr`), `std::auto_ptr<T>`.
- `std::auto_ptr<T>`.

> `std::auto_ptr`

`std::auto_ptr<T>`, `std::auto_ptr<T>`. `std::auto_ptr<T>`, `std::auto_ptr`, `std::auto_ptr<T>`. `std::auto_ptr` `std::unique_ptr` `ptr`, `std::auto_ptr`

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

`std::auto_ptr`, `std::auto_ptr`. `std::auto_ptr<T>`, `std::auto_ptr`, `std::auto_ptr<T>` `std::shared_ptr<const Foo>` `ptr`, `std::auto_ptr`. `std::auto_ptr<T>`, `std::auto_ptr` `std::shared_ptr`. `std::auto_ptr` `std::auto_ptr`, `std::auto_ptr` `std::unique_ptr` `ptr`.

2.6.2 5.2. Cpplint

>

cpplint.py

>

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

2.6.3 acgyrant

1. acgyrant, acgyrant.
2. Rust Ownership C++.
3. scoped_ptr auto_ptr. shared_ptr unique_ptr.
4. acgyrant, acgyrant, acgyrant, acgyrant.
5. Arch Linux, AUR cpplint.

2.7 6. C++

2.7.1 6.1.

 Tip

:

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

“&&”

:

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

```
std::unique_ptr, std::move
```

:

(C++11),

:

```
std::forward std::move
```

2.7.2 6.2. [C++](#)

 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()

2.7.3 6.3. [C++](#)

 Tip

??:

??

??:

function signature site acgtyrant

??:

acgtyrant .cc

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

2.7.4 6.4. `alloca()`

Tip

`alloca()`.

Tip:

`alloca()` is not thread-safe.

Tip:

`alloca()` is not C++ safe. See [this](#), [this](#), [this](#) bugs: “`alloca()`, `alloca()`”.

Tip:

`allocator_traits::vector` or `std::unique_ptr<T[]>`.

2.7.5 6.5. `Foo`

Tip

`Foo`.

`FooBuilder` is a `Foo` builder. `FooBuilder` is a `Foo` builder. `FooBuilder` is a `Foo` builder. `FooBuilder` is a `Foo` builder.

`FooBuilder` (public) `FooBuilder`. `FooBuilder` public, `FooBuilder`, `FooBuilder`. `FooBuilder`, `FooBuilder`.

2.7.6 6.6. `Foo`

Tip

`Foo` C++.

Tip:

- `failures` `acgyrant` error code, `int`
- `C++` `Python`, `Java` `C++`
- `C++`

- `acgtyrant` `factory function`, `C++` `Init()` `“”`
- `throw`

##:

- `throw` `g()`, `g()` `h()`, `h` `f` `g`
- `RAII`
- `RAII` `“”` `(` `)`
- `Windows`
- `Windows`

##:

`throw`, `RAII`, `Google` `C++`

`Google`, `Windows`

`Google`

`Windows`

(YuleFox `throw`, `C++`, `Google`)

2.7.7 6.7. RTTI

TODO

 Tip

RTTI.

##:

RTTI `C++` `typeid` `dynamic_cast`

##:

RTTI `(` `)`

RTTI `RTTI`

RTTI `RTTI`

C C++

- `static_cast` C, `static_cast` C++.
- `const_cast` `const`.
- `reinterpret_cast` `reinterpret_cast`.

`dynamic_cast` 6.7.

2.7.9 6.9.

Tip

:

`printf()` `scanf()`.

:

, `printf`, `scanf`. `printf` `scanf` (`gcc` `printf`).

:

`pread()` `printf` `printf` `printf` (%.*s) `printf` (%ls), `printf`.

:

`printf`, `printf`.

`printf`, `printf`.

:

`printf`, `printf`. `printf` (Only One Way): `printf` I/O, `printf` I/O. `printf`, `printf` + `read/write`.

`printf`, `printf`. `printf`. `printf`.

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

`cout << , ,`

`printf` `printf`, `printf`, `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.

2.7.10 6.10. ...

Tip

... (++) ...

...:

... (++i ... i++) ... (--i ... i--) ...

...:

... (++i) ... (i++) ... i ...

...:

... C ... for ... (i) ... ++ ...

...:

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

2.7.11 6.11. 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 ...

- `constexpr const`.

`constexpr const int * const * const x;` `constexpr const int** x;`

`mutable constexpr`, `constexpr`.

`constexpr`:

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

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

2.7.12 6.12. constexpr



Tip

C++11 `constexpr`

`constexpr`:

`constexpr constexpr` `constexpr`

`constexpr`:

`constexpr`

`constexpr`:

`constexpr constexpr`

`constexpr`:

<code>constexpr</code>	<code>constexpr</code>	C++	<code>constexpr</code>	<code>constexpr</code>
<code>constexpr</code>	<code>constexpr</code>		<code>constexpr</code>	<code>constexpr</code>
<code>constexpr</code>				

2.7.13 6.13.



Tip

C++ `constexpr`, `constexpr int`. `constexpr`, `constexpr <stdint.h>`, `constexpr`, `constexpr int16_t`, `constexpr 2^31 (2GiB)`, `constexpr 64` `constexpr int64_t`. `constexpr int`

`constexpr`:

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

`constexpr`:

`constexpr`.

??:

C++

??:

<stdint.h> int16_t, uint32_t, int64_t short, unsigned long long C, int. size_t ptrdiff_t. int, int. 32, 32, 64, int64_t uint64_t. uint32_t, int64_t, unsigned int. acgyrant integer promotions, int unsigned int unsigned int

??????:

???, ?????, ?????. ??, ? C ???, ????? bug ??:

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

!!!!!!! ?? gcc ??? bug ???, ??? bug ???????. ??? C ??????. ??, ??????

2.7.14 6.14. 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"
```

(continues on next page)

(continued from previous page)

```
#define PRIxS __PRIS_PREFIX "X"
#define PRIoS __PRIS_PREFIX "o"
```

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

?? PRI* ??????????????????. ?????????????????????, ??????????????????????. ?? PRI* ????????? % ??????????. ??, printf("x = %30PRIuS\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 ?????. (?????????, ?????????, ??????????)

2.7.15 6.15. ??????

 **Tip**

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

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

????????, C++ ?, ????? C ??????????. ?????????????????????, ??????????????????. ????????? const ??????. ?? "???" ??????????????????. ?????????????... ??, ?????????, ?????????? (#define ?????????????????????????????).

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

- ????????????????????????????????; ??????, ??????:
- ????? .h ?????????.
 - ??????????????? #define, ?????????? #undef.
 - ???????????????????#undef????????????????????????????
 - ??????????????????? C++ ??????????, ?????????????????????????????.
 - ????? ## ?????????????????????????

2.7.16 6.16. 0, nullptr, NULL

Tip

```

nullptr '\0' (0)
(nullptr) nullptr
'\0'

```

2.7.17 6.17. 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;
}

```

2.7.18 6.18. 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 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
```

2.7.19 6.19. ?????



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"};
```

```

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

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

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

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

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

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

```

```

std::initializer_list<T>

```

```

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

```

```

auto

```

Warning

```

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

```

```

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

```

```

9.7.

```

2.7.20 6.20. Lambda

Tip

lambda lambda

Tip

Lambda

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

```
int weight = 3;
int sum = 0;
// Captures `weight` by value and `sum` by reference.
std::for_each(v.begin(), v.end(), [weight, &sum](int x) {
    sum += weight * x;
});
```

lambda this

```
const std::vector<int> lookup_table = ...;
std::vector<int> indices = ...;
// Captures `lookup_table` by reference, sorts `indices` by the value
// of the associated element in `lookup_table`.
std::sort(indices.begin(), indices.end(), [&](int a, int b) {
    return lookup_table[a] < lookup_table[b];
});
```

```
std::unique_ptr<Foo> foo = ...;
[foo = std::move(foo)] () {
    ...
}
```

lambda

Lambda

```
[foo = std::vector<int>({1, 2, 3})] () {
    ...
}
```

Tip

- STL Lambda
-
- Lambda, std::functions, std::bind general purpose callback mechanism

Tip

- lambda `auto lambda`
- `this` `this`
- C++ `auto`
- `auto`
- lambda `lambda`

???

- `lambda`
- `lambda`

```
{
  Foo foo;
  ...
  executor->Schedule([&] { Froblicate(foo); });
  ...
}
// BAD! The fact that the lambda makes use of a reference to `foo` and
// possibly `this` (if `Froblicate` is a member function) may not be
// apparent on a cursory inspection. If the lambda is invoked after
// the function returns, that would be bad, because both `foo`
// and the enclosing object could have been destroyed.
```

?????

```
{
  Foo foo;
  ...
  executor->Schedule([&foo] { Froblicate(foo); })
  ...
}
// BETTER - The compile will fail if `Froblicate` is a member
// function, and it's clearer that `foo` is dangerously captured by
// reference.
```

- `lambda` `[&]`
- `this` `lambda` `this` `[&]` `lambda`
- `lambda`
- `lambda`

2.7.21 6.21. [\[?\]\[?\]\[?\]](#)

 Tip

[\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#)

[\[?\]\[?\]](#):

[\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#)c++ [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#)

[\[?\]\[?\]](#):

[\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#)Google Test, [std::tuple](#), [std::function](#) [\[?\]](#) [Boost.Spirit](#).
[\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#)


[\[?\]\[?\]](#):

- [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#)c++[\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]](#). [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]](#)debug [\[?\]](#) [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#)
- [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#): [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#).
[\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#)
- [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#)(Visual Assist X, Refactor for C++[\[?\]\[?\]](#))[\[?\]\[?\]\[?\]\[?\]\[?\]](#). [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#),
[\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#)AST [\[?\]](#).
[\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#).

[\[?\]\[?\]](#):

- [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#). [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#),
[\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#)
- [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#). [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#).[\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#).[\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#)c++
[\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#). [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#),
[\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#)
- [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#). [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#),
[\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#). [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#). [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#),
[\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#). [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#).
[\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#)

2.7.22 6.22. Boost [\[?\]](#)

 Tip

[\[?\]\[?\]](#) Boost [\[?\]\[?\]\[?\]\[?\]\[?\]](#).

[\[?\]\[?\]](#):

Boost [\[?\]](#) [\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]](#) C++ [\[?\]](#).

[\[?\]\[?\]](#):

Boost[\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]](#) C++ [\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]](#)

[\[?\]\[?\]](#):

[\[?\]](#) Boost [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]\[?\]](#), [\[?\]\[?\]\[?\]\[?\]\[?\]](#) “[\[?\]\[?\]\[?\]](#)” [\[?\]\[?\]\[?\]\[?\]\[?\]](#).

??:

Boost

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

Boost

C++ 11

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

2.7.23 6.23. C++11

 Tip

C++11 C++0x C++11

???

C++11

???

C++11 C++

???

C++11 1300 vs 800

6.22. Boost C++11

???

C++11 `auto foo() -> int` `int foo()`. `<ratio>`, `<cfenv>` `<fenv.h>` `lambda`

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

2.7.24 `acgtyrant`

1. `void a()` `void a(int b = 0)`, `int`
2. `friend` `.cc` `friend`
3. `friend` `.cc` `friend`
4. `friend` `.cc` `friend`
5. `friend` `.cc` `friend`
6. `friend` `.cc` `friend`
7. `friend` `.cc` `friend`
8. `friend` `.cc` `friend`
9. `friend` `.cc` `friend`
10. `friend` `.cc` `friend`
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" ..
```



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

2.8.4 7.4. Enumerations

Enum

Enum (enumeration) defines a set of named integer constants. Enumerations are used to define a set of related constants, such as `a_local_variable`, `a_struct_data_member`, `a_class_data_member`.

Enum

Enumerations

Enum:

```
string table_name; // 0 - 0000.
string tablename; // 0 - 000.

string tableName; // 0 - 00000
```

Enumerations

Enumerations, Enumerations, Enumerations.

```
class TableInfo {
    ...
private:
    string table_name_; // 0 - 00000.
    string tablename_; // 0.
    static Pool<TableInfo>* pool_; // 0.
};
```

Enumerations

Enumerations, Enumerations, Enumerations:

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

Enumerations, Enum vs. Enum.

2.8.5 7.5. [Constant Expressions](#)

??

constexpr const int, [Constant Expressions](#), ["k"](#) int, [int](#). ??:

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

??

[Constant Expressions](#) ([Constant Expressions](#), [int](#) [int](#)) [Constant Expressions](#). [Constant Expressions](#), [int](#), [Constant Expressions](#). [Constant Expressions](#), [Constant Expressions](#).

2.8.6 7.6. [Constant Expressions](#)

??

[Constant Expressions](#), [Constant Expressions](#): MyExcitingFunction(), MyExcitingMethod(), my_exciting_member_variable(), set_my_exciting_member_variable().

??

[Constant Expressions](#), [Constant Expressions](#) ("???" "???"), [Constant Expressions](#). [Constant Expressions](#), [Constant Expressions](#) (??, ?? StartRpc() ?? StartRPC()).

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

([Constant Expressions](#), [Constant Expressions](#) API [Constant Expressions](#), [Constant Expressions](#), [Constant Expressions](#).)

[Constant Expressions](#). [Constant Expressions](#), [Constant Expressions](#). int count() void set_count(int count).

2.8.7 7.7. [Constant Expressions](#)

??

[Constant Expressions](#). [Constant Expressions](#). [Constant Expressions](#).

[Constant Expressions](#). [Constant Expressions](#), [Constant Expressions](#).

?? [Constant Expressions](#) [Constant Expressions](#). [Constant Expressions](#), [Constant Expressions](#).

[Constant Expressions](#). [Constant Expressions](#), [Constant Expressions](#). int, [Constant Expressions](#) std [Constant Expressions](#). [Constant Expressions](#) (websearch::index, websearch::index_util) [Constant Expressions](#) (?? websearch::util).

?? internal [Constant Expressions](#), [Constant Expressions](#) internal [Constant Expressions](#) ([Constant Expressions](#), [Constant Expressions](#)). [Constant Expressions](#), [Constant Expressions](#) (??? frobber.h, ?? websearch::index::frobber_internal).

2.8.8 7.8. [unclear]

[unclear]

[unclear] [unclear] [unclear] [unclear]: kEnumName [unclear] ENUM_NAME.

[unclear]

[unclear] [unclear] [unclear] [unclear]. [unclear] [unclear] [unclear] [unclear]. [unclear] UrlTableErrors ([unclear] AlternateUrlTableErrors) [unclear], [unclear] [unclear] [unclear] [unclear].

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

2009 [unclear] 1 [unclear], [unclear] [unclear] [unclear] [unclear] [unclear]. [unclear] [unclear] [unclear] [unclear], [unclear] [unclear] [unclear]. [unclear], [unclear] [unclear] [unclear] [unclear] [unclear] [unclear] [unclear] [unclear]. [unclear] [unclear] [unclear] [unclear] [unclear] [unclear] [unclear] [unclear], [unclear] [unclear] [unclear] [unclear] [unclear] [unclear].

2.8.9 7.9. [unclear]

[unclear]

[unclear] [unclear] [unclear], [unclear] [unclear] [unclear], [unclear]: MY_MACRO_THAT_SCARES_SMALL_CHILDREN.

[unclear]

[unclear] [unclear]; [unclear] [unclear] [unclear]. [unclear] [unclear], [unclear] [unclear] [unclear] [unclear] [unclear] [unclear] [unclear] [unclear], [unclear]:

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

2.8.10 7.10. [unclear]

[unclear]

[unclear] [unclear] [unclear] [unclear] C/C++ [unclear], [unclear] [unclear] [unclear].

bigopen(): [unclear], [unclear] open() [unclear]












uint: typedef

bigpos: struct [unclear] class, [unclear] pos [unclear]









sparse_hash_map: STL [unclear]; [unclear] STL [unclear]





LONGLONG_MAX: [unclear], [unclear] INT_MAX

2.8.11 acgyrant

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




2.9 8.

, . . : , .
, .







, . , 

2.9.1 8.1.




 //  /* */ , .

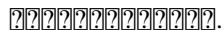



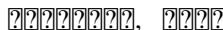




//  /* */ ;  //  . .

2.9.2 8.2.

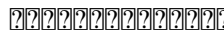
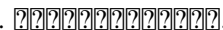



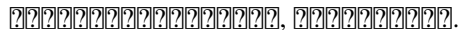


. , , , , .




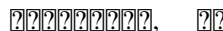
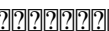








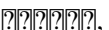


. .(, Apache 2.0, BSD, LGPL, GPL)





 .h , , . , , .

 .h  .cc , .

2.9.3 8.3. Iterator

??

Iterates over the contents of a GargantuanTable, returning a GargantuanTableIterator.

```
// 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 {
    ...
};
```

??

Iterates over the contents of a GargantuanTable, returning a GargantuanTableIterator. The iterator is responsible for deleting itself 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.

Iterates over the contents of a GargantuanTable, returning a GargantuanTableIterator.

Iterates over the contents of a GargantuanTable (GargantuanTable.h and GargantuanTable.cc), returning a GargantuanTableIterator, which iterates over the contents of the GargantuanTable.

2.9.4 8.4. Iterator

??

Iterates over the contents of a GargantuanTable; returns a GargantuanTableIterator.

??

Notes

Iterates over the contents of a GargantuanTable, returning a GargantuanTableIterator (GargantuanTable.h, GargantuanTable.cc). Iterates over the contents of a GargantuanTable (GargantuanTable.h, GargantuanTable.cc). Iterates over the contents of a GargantuanTable (GargantuanTable.h, GargantuanTable.cc). Iterates over the contents of a GargantuanTable (GargantuanTable.h, GargantuanTable.cc).

Notes:

- Iterates over the contents of a GargantuanTable.
- Iterates over the contents of a GargantuanTable; returns a GargantuanTableIterator, which iterates over the contents of the GargantuanTable.
- Iterates over the contents of a GargantuanTable.
- Iterates over the contents of a GargantuanTable.
- Iterates over the contents of a GargantuanTable.
- Iterates over the contents of a GargantuanTable, returning a GargantuanTableIterator?

Notes:

```
// 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.
//
```

(continues on next page)

(continued from previous page)

```
// 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;
```

returns true if the table cannot hold any more entries. Returns false, otherwise:

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

returns true if the table cannot hold any more entries. Returns false, otherwise. Returns true if the table cannot hold any more entries. Returns false, otherwise. Returns true if the table cannot hold any more entries. Returns false, otherwise.

2.9.5

returns true if the table cannot hold any more entries. Returns false, otherwise. Returns true if the table cannot hold any more entries. Returns false, otherwise.

returns true if the table cannot hold any more entries. Returns false, otherwise.

2.9.5 8.5

returns true if the table cannot hold any more entries. Returns false, otherwise.

returns true if the table cannot hold any more entries. Returns false, otherwise.

returns true if the table cannot hold any more entries. Returns false, otherwise.

2.9.5 8.5

returns true if the table cannot hold any more entries. Returns false, otherwise. Returns true if the table cannot hold any more entries. Returns false, otherwise.

returns true if the table cannot hold any more entries. Returns false, otherwise.

```
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_;
```

[REDACTED]

[REDACTED], [REDACTED], [REDACTED]. [REDACTED]:

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

2.9.6 8.6. [REDACTED]

[REDACTED]

[REDACTED], [REDACTED], [REDACTED], [REDACTED].

[REDACTED]

[REDACTED]

[REDACTED]. [REDACTED]:

```
// 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;
}
```

[REDACTED]

[REDACTED]. [REDACTED]. [REDACTED]:

```
// 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.
```

[REDACTED], [REDACTED], [REDACTED].

[REDACTED], [REDACTED]:

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

QUESTION

QUESTION, ANSWER:

- QUESTION, ANSWER, QUESTION, ANSWER, QUESTION.
- QUESTION, bool enum, QUESTION.
- QUESTION, QUESTION, QUESTION. QUESTION, QUESTION, QUESTION, QUESTION. QUESTION, QUESTION, QUESTION, QUESTION.
- QUESTION.
- QUESTION, QUESTION.

QUESTION:

```
// 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);
```

QUESTION.

QUESTION

QUESTION, ANSWER, QUESTION, ANSWER C++ QUESTION. QUESTION C++ QUESTION, QUESTION:

QUESTION, ANSWER, QUESTION.

QUESTION:

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

QUESTION:

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

QUESTION. QUESTION:

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

2.9.7 8.7. `/*`, `*/`

/*

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

/*

/*, */. /*, */. /*, */. /*, */. /*, */.

/*, */. /*, */. /*, */.

2.9.8 8.8. TODO

/*

/*, */, /* TODO

TODO /* TODO, /*, bug ID, /* 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
60
80
URL
#include

2.10.2 9.2. ASCII

ASCII
UTF-8
"\xEF\xBB\xBF"
u8 "\uFEFF"
Unicode
UTF-8 with BOM
u8
uXXXX
char16_t
char32_t
wchar_t
Windows API
wchar_t

2.10.3 9.3. `constexpr`

??

constexpr, constexpr 2 ???.

??

constexpr. constexpr. constexpr.

2.10.4 9.4. `constexpr`

??

constexpr, constexpr, constexpr, constexpr ??? ??

??

constexpr:

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

constexpr, constexpr:

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

constexpr:

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

constexpr:

- constexpr.
- constexpr, constexpr.
- constexpr, ??
- constexpr, ???.
- constexpr.
- constexpr.
- constexpr.
- constexpr, ???.
- constexpr, constexpr.

- `foo.foo()`.
- `foo.foo(1)`.
- `foo.foo(2, 3)`.
- `foo.foo(4, 5, 6, 7)`.

`foo.foo()`, `foo.foo(1)`, `foo.foo(2, 3)`:

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

`foo.foo(1)`, `foo.foo(2, 3)`:

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

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

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

```
// 2 - foo.foo(), foo.foo(1).
void Circle::Rotate(double) {}
```

`foo.foo()`, `foo.foo(1)`, `foo.foo(2, 3)`:

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

2.10.5 9.5. Lambda

??

Lambda `foo.foo()`; `foo.foo(1)`, `foo.foo(2, 3)`.

??

`foo.foo()`, `foo.foo(1)` & `foo.foo(2, 3)`.

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

?? lambda `foo.foo()`.

```
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();
}));
```

(continues on next page)

```
    },
    digits.end());
```

2.10.6 9.6. `constexpr`

`constexpr`

`constexpr` variables, `constexpr` functions, `constexpr` variables. `constexpr` variables, `constexpr` functions, `constexpr` variables.

`constexpr`

`constexpr`

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

`constexpr`, `constexpr`, `constexpr`, `constexpr`

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

`constexpr`, `constexpr`

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

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

`constexpr`, `constexpr`, `constexpr`, `constexpr`

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

`constexpr`, `constexpr`

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

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

`constexpr`, `constexpr`, `constexpr`

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

2.10.7 9.7. `return`

??

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

??

????????????????, ?????????, ?????????????????, {} ??????????. ??????, ?????????????.

```
// ??????????.
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`

??

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

??

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

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

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

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

```
if ( condition ) { // IF - DO
    ... // 2 DO.
} else { // ELSE DO IF.
    ...
}
```

IF-DO, ELSE-DO, IF-DO-ELSE-DO:

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

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

IF-DO, ELSE-DO, IF-DO-ELSE-DO else DO:

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

ELSE-DO:

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

IF, ELSE-DO, IF-DO-ELSE-DO; IF-DO-ELSE-DO-DO. IF-DO-ELSE-DO-DO:

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

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

IF-ELSE-DO, ELSE-DO-DO:

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

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

```
// DO-DO-DO-DO-DO, DO-DO-DO-DO-DO.
if (condition) {
    foo;
} else {
    bar;
}
```

2.10.9 9.9. switch

??

switch (var) { cases ... } continue.

??

switch (var) case ...

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

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

(continues on next page)

(continued from previous page)

```
BackToNormal();
}
```

```
// 2 - 2000
if (lopsided_score) {
    #if DISASTER_PENDING // 2 - "#if" 20000000
    DropEverything();
    #endif // 2 - "#endif" 2000
    BackToNormal();
}
```

2.10.15 9.15. 222

22

222222222222222222 public:,protected:,private:, 22222 1 222.

22

222 (222222222222, 22 222) 22222222:

```
class MyClass : public OtherClass {
public: // 222222222222
    MyClass(); // 2222222222
    explicit MyClass(int var);
    ~MyClass() {}

    void SomeFunction();
    void SomeFunctionThatDoesNothing() {
    }

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

private:
    bool SomeInternalFunction();

    int some_var_;
    int some_other_var_;
};
```

22222:

- 22222222 80 22222222222222222222.
- 222 public:,protected:,private: 222 1 222.
- 22222222 (222 public) 2, 222222222222. 2222222222222222.
- 2222222222222222.
- public 222222, 222 protected, 222 private.
- 2222222222222222 2222 22.

2.10.16 9.16.

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

// 
// 
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 {

// , .
```

(continues on next page)

(continued from previous page)

```

void foo() {
    ...
}

} // namespace

```

namespace foo, namespace bar.

```

namespace foo {
namespace bar {

```

2.10.18 9.18. namespace

namespace

namespace foo, namespace bar.

namespace

namespace

```

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

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

```

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

namespace

```

if (b) { // if namespace.
} else { // else namespace.
}

while (test) {} // namespace.
switch (i) {
for (int i = 0; i < 5; ++i) {
switch ( i ) { // namespace.
if ( test ) { // namespace. namespace. namespace.

```

(continues on next page)

(continued from previous page)

```
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;
y = static_cast<char*>(x);

// , .
vector<char *> x;
```

2.10.19 9.19. ????

??

????????.

??

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

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

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

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

2.10.20 [???](#) (YuleFox) [???](#)

1. [XXXXXXXXXX, ??, XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX](#);
2. [XXXXXXXXXX 80 ?](#), [? 22 XXXXXXXXXXXXX, XXXXXXXX](#);
3. [XXXXXXXXXX ASCII ??, XXXXXXXX, ?? UTF-8 ?? \(??? UNIX/Linux ?, Windows XXXXXXXXXXXXX\), XXXXXXXXXXXXXXXXXXXXXXX, XXXXXXXXXXXXX, XXXXXXXXXXXXXXX](#);
4. UNIX/Linux [XXXXXXXXXXXX](#), MSVC [XXXXX](#) Tab [XXXXXXXX](#);
5. [XXXXX, XXXXX, XXXXXX: XXXXXXXXXXXXXXXXXXXXXXX, XXXXXXXXXXXXXXX](#);
6. [XXXXXXXXXXXXXXXXXXXXXXXXXXXX, XXXXX/?/XXX/XXXX, XXXXXXXXXXXXXXXXXXXXXXX, XXXXXXXXXXXXXXX](#);
7. [./-> XXXXXXXXXXXXX, *& XXXXXXX, XXXXX, XXXXXXXXXXXXXXX](#);
8. [XXXXXXXX/XXXXXXXXXXXXXXXXXXXX, ?/XXX/?/?/XXXXXXXX](#);
9. [XXXXX = ?? \(\) XXXXXX, XXXXX](#);
10. `return XXXX ();`
11. [??/XXXXXXXXXXXX, XXXXXXXX](#).
12. [?? UNIX/Linux XXXXXXXXXXXXXXXXXXXXXXX \(.cc XXXXXXXXXXXXX, XXXXXXXXXXXXX\), XXXXXXXXXXXXXXXXXXXXXXX, XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX, XXXXXXXXXXXXXXXXXXXXXXX](#); Windows [XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX](#).

2.10.21 [????acgtyrant????](#)

1. [80 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX, XXXXXXXXXXXXX, XXXXXXXXXXXXXXX](#).
2. [Linux XXXXXX Locale XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX, XXXXXX Windows](#).
3. [Google XXXXXX if-else ?, XXXXXXXXXXXXX, XXXXXXXX. Apple ?? XXXXXX](#) .
4. [XXXXXXXXXXXXXXXXXXXXXXXXXXXX, int* a, bvs int *a, b,XXXXXXXXXXXX b ? int * ??, XXXXXXXXXXXXX, XXXXX](#).
5. [XXXXXXXXXXXXXXXXXXXX C++ XXXXXXXXXXXX Alternative operator representations, XXXXXXXX](#).
6. [XXXXXXXXXXXXXXXXXXXXConstructor Initializer ListXXXXXXXXXXXXXXXXXXXXInitializer ListXXXXXX, XXXXXXXXXXXXXXXXXXXXXXX](#).
7. [XXXX, XXXXXXXXXXXXXXXXXXXXXXX, XXX. XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX, XXXXXXXX; XXXXXXXXXXXXXXXXXXXXXXX, ?? if \(true\) XXXXXXX true.](#)
8. [XXXXXXXXXXXXXXXXXXXX void XXXXXXXX return ??, ??? Google XXXXX leveldb XXXXX; XXXX Is a blank return statement at the end of a function whos return type is void necessary? ??, return; ? return ; XXXXXXXXXXXXXXX cpplint XXXXXXX, XXXXXXXXXXXXXXXXXXXXXXX, XXXXXXXXXXXXXXXXXXXXXXX](#).

2.11 10. [?????](#)

[XXXXXXXXXXXXXXXXXXXXXXXXXXXX. XXXXXXXXXXXXXXXXXXXXXXX, XXXXXXXXXXXXXXX.](#)

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.

OBJECTIVE-C [?] - [?]

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 [?]vim[?]HTML[?]
"ewangke at gmail.com" 2011.03.27

Yang.Y

[?] Objective-C [?] C/C++ [?]

- [?] 2.36 [?]
- [?]

3.1.2 [REDACTED]

Objective-C [REDACTED] C [REDACTED] Mac OS X [REDACTED] iPhone [REDACTED]

Cocoa [REDACTED] Mac OS X [REDACTED] Objective-C [REDACTED] Mac OS X [REDACTED]

[REDACTED] Objective-C [REDACTED] Google [REDACTED] C++ [REDACTED] Objective-C [REDACTED] Google [REDACTED]

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

Note

[REDACTED] Google [REDACTED] C++ [REDACTED] Objective-C++ [REDACTED]

[REDACTED] Mac OS X [REDACTED] Google [REDACTED]

Google [REDACTED] Google Toolbox for Mac project [REDACTED] GTM [REDACTED]

[REDACTED] Objective-C [REDACTED] Objective-C [REDACTED] Objective-C [REDACTED] The Objective-C Programming Language [REDACTED]

3.1.3 [REDACTED]

[REDACTED]

[REDACTED] @interface [REDACTED]

```

// 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];
}

- (NSString *)bar {
    return bar_;
}

}
```

(continues on next page)

(continued from previous page)

```

- (void)setBar:(NSString *)bar {
    [bar_ autorelease];
    bar_ = [bar copy];
}

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

@end

```

@interface @implementation @end @interface }
 @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

- +


```
[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_;
}
- (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 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++ `objc++`

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

```
// file: cross_platform_header.h

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

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

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

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

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

3.3.3

Tip

GTMSendMessage

3.3.4

Tip

NSString GTMNSString+Parsing.h GTMStringParsingAdditions gtm_myCategoryMethodOnAString: Objective-C

3.3.5 Objective-C

Tip

convertPoint:fromRect: replaceCharactersInRange:withString: Apple's Guide to Naming Methods

get

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

Objective-C C++ C++

3.3.6

Tip

myLocalVariable myInstanceVariable Objective-C 2.0 @property KVO/KVC

int

-

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

-

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

Objective-C

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


Objective-C

k kInvalidHandle kWritePerm

3.4 Objective-C

Objective-C
 Objective-C
 Objective-C C++ Objective-C

3.4.1 Objective-C

 Tip

Objective-C

Objective-C

Objective-C

- Objective-C
- Objective-C
- Objective-C 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 re-
tained __strong __weak CoreFoundation Objective-C
```

_____ __weak _____ clang _____ C++

Objective-C _____ C++ _____ ? ?

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

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

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

_____ - _____ retained_____ - _____ retained_____

3.5 Cocoa Objective-C

3.5.1 @private



Tip

_____ @private

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

3.5.2



Tip

3.5.3

Tip

```
init... 
```

bug

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 - 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`

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

```
@implementation
```

“middle truncation” `NSString` `NSString` 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 #in-  
clude
```

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

3.5.9

Tip

```
#import
```

Cocoa Foundation Objective-C

```
#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 @interface `dealloc`

3.5.14 setter NSStrings

Tip

NSString setter copy

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

int main(int argc, char *argv[]) {
    GMAutoreleasePool pool;
    @try {
        foo();
    }
    @catch(NSEException *ex) {
        NSLog(@"exception raised");
    }
    return 0;
}
```

smartptr shared_ptr linked_ptr STL

Objective-C++ C++ Objective-C @try @catch @finally C++

3.5.16 nil

Tip

nil

nil Objective-C nil OS X Apple's documentation

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

3.5.17 BOOL

Tip

BOOL YES

Objective-C BOOL YES ``(1) ``NO``(0) ``BOOL NO 256 256 512 ...

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

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 {
    return [self isValid] && [self isBold];
}

```

YES/NO BOOL

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

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

3.5.18 Property



Tip

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

@synthesize

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

@implementation MyClass
@synthesize name = name_;
@end
```

@implementation @interface @implementation

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

@implementation MyClass
@synthesize name = name_;
- (id)init {
```

(continues on next page)

(continued from previous page)

```
...
}
@end
```

`copy` `Attribute`

```
copy attribute NSString property
NSString setter copy retain
```

`property`

```
property synthesize setter getter get set
property nonatomic
```

`Objective-C 2.0`

```
Objective-C 2.0 set/get

```

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

`NSArray`

```
NSArray *array = [[NSArray arrayWithObject:@"hello"] retain];
NSUInteger numberOfItems = array.count; // not a property
array.release; // not a property
```

3.5.19

 Tip

`@interface`

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

`@interface`

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

3.5.20 `synthesize`

 Tip

iOS `synthesize`

```
synthesize @synthesize var = var_; self.var = blah; var =
blah;
```

```
synthesize CType@dynamic CType@dynamic CTyperetain
releasegettersetter
@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 “”

PYTHON 开发指南 - 第四卷

4.1 致谢

感谢

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

感谢所有为本书做出贡献的 Git 用户。

感谢

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

感谢

- [Google Style Guide](#) (感谢)
- [Google 开发指南](#) - 感谢

感谢

Python 代码遵循 Apache License 2.0，文档遵循 CC-BY 3.0。

4.2 附录

Python 开发指南 Python 开发指南
感谢 Vim 和 Emacs
感谢 Black 和 Pyink

4.3 Python

4.3.1 Lint

 Tip

`pylintrc` `pylint`, `pylint`.

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

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

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

`pylint`.

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

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

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

`pylint`

`pylint`, `pylint`.

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

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

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

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

(`Viking`.)

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

4.3.2

Tip

```
import sys, os, random
```

Tip:

```
sys.stdout.write('')
```

Tip:

```
sys.stdout.write('x. Obj %s Obj %s x %s')
```

Tip:

```
sys.stdout.write('')
```

Tip:

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

`sound.effects.echo:`

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

```
sys.stdout.write('')
```

Tip:

```
sys.stdout.write('')
```

1. `typing`
 1. `typing`
 2. `collections.abc`
 3. `typing_extensions`
2. `six.moves`

4.3.3

Tip

```
sys.stdout.write('')
```

Tip:

```
sys.stdout.write('')
```

Tip:

```
sys.stdout.write('')
```

??:

```
#####.
```

#####:

??:

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

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

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

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

??: (##### jodie.py ##### doctor/who/ ?)

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

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

4.3.4 ??

💡 Tip

#####, #####.

??:

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

??:

```
#####. #####, #####.   ?, #####N#####, #####.
```

??:

```
#####. #####.
```

??:

```
#####:
```

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

??:

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

(continues on next page)

(continued from previous page)

```

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

    Parameters:
        minimum: The minimum port number to try.

    Returns:
        The next open port number.

    Raises:
        ValueError: If the minimum port number is less than 1024.
        ConnectionError: If the port is not open.
    """
    if minimum < 1024:
        # ValueError: minimum must be at least 1024
        # ValueError: minimum must be at least 1024
        raise ValueError(f'minimum must be at least 1024 (got {minimum}).')
    port = self._find_next_open_port(minimum)
    if port is None:
        raise ConnectionError(f'port {minimum} is not open.')
    assert port >= minimum, (
        f'port {port} is less than minimum {minimum}.')
    return port

```

```

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

    Parameters:
        minimum: The minimum port number to try.

    Returns:
        The next open port number.

    Raises:
        ValueError: If the minimum port number is less than 1024.
    """
    assert minimum >= 1024, 'minimum must be at least 1024.'
    port = self._find_next_open_port(minimum)
    assert port is not None
    return port

```

2. `raise ValueError('minimum must be at least 1024 (got foo).')` Error `ValueError`, `ValueError('foo foo. FooError')`.
3. `except: ValueError, Exception` `StandardError`, `except: ValueError`.
 1. `ValueError`.
 2. `ValueError`, `ValueError`, `ValueError`. `ValueError`, `ValueError`.

`Python` `except: ValueError, ValueError sys.exit()` `Ctrl+C` `ValueError`.
4. `try/except ValueError`. `try ValueError`, `ValueError`. `try/except ValueError`.
5. `finally ValueError`. `ValueError`, `ValueError`.

4.3.5

 Tip

:
 (class attribute).

:
 .


- :
 .
1. : . , , ().
 2. , .

:
 .

, , _ . , .

`_MAX_HOLY_HANDGRENADE_COUNT = 3
 SIR_LANCELOTS_FAVORITE_COLOR = "blue"`

4.3.6

 Tip

:
 . . (nonlocal)

:
 . .

:
 . , .

:
 . self cls . . _ , .

4.3.7 Comprehension expression (generator expression)

Tip

Comprehensions.

Comprehensions are a compact way to create lists, dictionaries, sets, and generators. They are similar to the `map()` and `filter()` functions, but they allow you to combine them in a single expression. (Comprehensions are also used to create dictionaries, sets, and generators.)

Comprehensions are used to create lists, dictionaries, sets, and generators. They are similar to the `map()` and `filter()` functions, but they allow you to combine them in a single expression.

Comprehensions are used to create lists, dictionaries, sets, and generators.

Comprehensions are used to create lists, dictionaries, sets, and generators. They are similar to the `map()` and `filter()` functions, but they allow you to combine them in a single expression.

```

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)
]

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

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

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

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

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

```

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

(continues on next page)

(continued from previous page)

```

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).

:
 . , . . , .

:
 ().

:
 (), , , ,

:

```

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

```

:

```

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

```

4.3.9

Tip

:
 yield , . , .

?:
 , , . , .

?:
 , .

?:
 . "Yields:" "Returns:".
 (:)
 , .
 PEP-0533.

4.3.10 Lambda

Tip

map()/filter() lambda

?:
 lambda , .

?:
 .

?:
 . 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')

```

Tip:

```

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

```

4.3.12



Tip

Python 3.6.

Tip:

```

def foo(a, b = 0):
    foo(a, b)

```

Tip:

```

def foo(a, b=None):
    if b is None:
        b = []

```

Tip:

```

def foo(a, b: Sequence = ()):
    # ...

```

Tip:

```

def foo(a, b=None):
    if b is None:
        b = []

```

Tip:

```

def foo(a, b=None):
    if b is None:
        b = []
def foo(a, b: Optional[Sequence] = None):
    if b is None:
        b = []
def foo(a, b: Sequence = ()): # ...

```

Tip:

```

from absl import flags
_FOO = flags.DEFINE_string(...)

def foo(a, b=[]):

```

(continues on next page)

(continued from previous page)

```

...
def foo(a, b=time.time()): #
...
def foo(a, b=_FOO.value): # sys.argv...
...
def foo(a, b: Mapping = {}): # (unchecked)
...

```

4.3.13 (properties)

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

Tip

(attribute) :

:

- :
1. (getter) (setter).
 - 2.
 - 3.
 - 4.

- :
1. (operator overload).
 - 2.

:

, , , . , .

, : , (). .

@property (decorator) .

. (override) .

4.3.14 True/False

Tip

“”.

:

Python False. , , , 0, None, [], {}, "" False.

??:

Python

??:

C/C++

??:

if foo: if foo != []: .

1. if foo is None: (foo is not None) None ., None None
2. == False. if not x: . False None, if not x and x is not None:.
3. (foo, [],) if not seq: if len(seq): , if not seq: if not len(seq): .
4. , False (None) (len()).

??:

```

if not users:
    print('')

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

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

??:

```

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

if not i % 10:
    self.handle_multiple_of_ten()

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

5. '0' True.
6. Numpy .size np.array (if not users. size).

4.3.15 (Lexical Scoping,)

Tip

??:

Python, .


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

Tip: Decorators are functions that take other functions as arguments, and they modify (decorate) them. An invariant is a property that remains true before and after some operation.

Tip: The decorator pattern is a design pattern that allows behavior to be attached to objects at run-time without permanently changing them. It is a simple way to extend the functionality of a class without changing its code.

Tip: The `pydoc` module provides a simple interface to the Python documentation. It can be used to generate HTML or text documentation for modules, classes, and functions. The `pydoc` module also provides a command-line interface for viewing documentation.

4.3.17

Tip
The `__hash__` and `__eq__` methods are used to define the hash and equality of objects. The `__hash__` method returns an integer value that is used to hash the object. The `__eq__` method returns a boolean value that indicates whether two objects are equal.

Python provides several built-in data structures, including lists, tuples, dictionaries, and sets. The `queue` module provides a `Queue` class for thread-safe queue operations. The `threading` module provides a `Condition` class for thread synchronization.

4.3.18

Tip
The `__del__` method is used to define the cleanup behavior of an object. It is called when the object is garbage collected.

Python provides several built-in data structures, including lists, tuples, dictionaries, and sets. The `import` statement is used to load modules and sub-modules. The `getattr()` function is used to get an attribute of an object. The `__del__` method is used to define the cleanup behavior of an object.

Tip: The `__del__` method is used to define the cleanup behavior of an object. It is called when the object is garbage collected.

Tip: The `__del__` method is used to define the cleanup behavior of an object. It is called when the object is garbage collected.

??:

???????

??, ?? abc.ABCMeta, dataclasses ? enum.

4.3.19 Python: from __future__ imports

 Tip

?????? __future__ ?, ??????????????????, ??????????????.

??:

???? from __future__ import ??????????, ?????????? Python ??

??:

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

??:

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

??:

from __future__ imports

???? from __future__ import ??. ??, ????????????????????????? Python ??. ?????????????????, ?????????????????.

???????????? 3.5 ??, ????????? >=3.7, ???:

```
from __future__ import generator_stop
```

???? Python future ?? ???.

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

????????????, ????????? from __future__ ???.

4.3.20 Python 3.6

 Tip

?????? PEP-484 ?? python3 ?????????,?????? pytype ?????????????????????.

????????????????????,?????? pyi ?. ??????????????. ?????????????????, ??? pyi ??.

??:

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

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

?????? PEP-526 ?????????????????:

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

Tip: `python --help`. `python --help`, `python --help`.

Tip: `python --help`. `python --help`. `python --help`.

Tip: `python --help`. `python --help` API, `python --help`, `python --help`(build system) `python --help` pytype. `python --help`python, `python --help`(`python --help`)`python --help`. `python --help`, `python --help` BUILD `python --help` TODO `python --help`, `python --help`.
(Tip: `python --help`IDE `python --help`vim `python --help`)

4.4 Python

4.4.1

Tip
`python --help`, `python --help`.

4.4.2

Tip
`python --help` 80 `python --help`.

- 1. `python --help` (import) `python --help`.
- 2. `python --help` URL `python --help` (flag).
- 3. `python --help` URL `python --help`.
- 4. Pylint `python --help`. (Tip: `python --help` # pylint: disable=invalid-name)

`python --help` `python --help` (explicit line continuation).

`python --help` Python `python --help`, `python --help` (implicit line joining) . `python --help`, `python --help`.

Tip:

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

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

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

with (
    very_long_first_expression_function() as spam,
```

(continues on next page)

(continued from previous page)

```

very_long_second_expression_function() as beans,
third_thing() as eggs,
):
    place_order(eggs, beans, spam, beans)

```

??:

```

if width == 0 and height == 0 and \
    color == '?' and emphasis == '??':

bridge_questions.clarification_on \
    .average_airspeed_of.unladen_swallow = '?????????'

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

```

???????? (literal) ????, ?????????????????:

```

x = ('?????????????????????'
     '?????????????????????')

```

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

??:

```

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

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

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

```

??:

```

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

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

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

```

???, ?????? URL ????????

??:

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

:

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

; 80 , Black Pyink , 80 . .

4.4.3

 Tip
.

(tuple) , . , .

:

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

:

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



```

"""blaze golden files.

blaze. google3. blaze,
2/blaze.

blaze:

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

```

blaze

blaze. google3. blaze, blaze, blaze.

```

"""blaze golden files.

blaze, google3. blaze
`blaze run //foo/bar:foo_test -- --update_golden_files`
"""

```

blaze.

```

"""foo.bar """

```

blaze

blaze (generator) blaze (property).

blaze:

1. blaze API
2. blaze
3. blaze

blaze, blaze. blaze, blaze, blaze, blaze, blaze, blaze, blaze, blaze, blaze.

blaze ("Fetches rows from a Bigtable.") blaze ("Fetch rows from a Bigtable."), blaze. @property blaze (data descriptor), blaze (attribute) blaze blaze ("Bigtable blaze." blaze "Bigtable blaze.").

blaze (override) blaze (base class) blaze, blaze, blaze. """blaze. """. blaze. blaze, blaze, blaze (blaze), blaze.

blaze. blaze, blaze. blaze, blaze2004000 (blaze) blaze. blaze (signature) blaze, blaze, blaze.

Args: (blaze):

blaze. blaze, blaze, blaze. blaze 80 blaze, blaze2004000 (blaze) blaze. blaze, blaze. blaze *foo (blaze) blaze **bar (blaze) blaze, blaze *foo **bar blaze.


```

class CheeseShopAddress:
    """
    ...
    """

class OutOfCheeseError(Exception):
    """
  
```

Code Review

Code review is a process where a developer reviews the code written by another developer. It helps to catch errors and improve code quality.

```

# ...
# ...
if i & (i-1) == 0: # ...
  ...

```

Code review is a process where a developer reviews the code written by another developer. It helps to catch errors and improve code quality.

Code review is a process where a developer reviews the code written by another developer. It helps to catch errors and improve code quality.

```

# ...
# ...

```

4.4.8

Tip

Code review is a process where a developer reviews the code written by another developer. It helps to catch errors and improve code quality.

Code review is a process where a developer reviews the code written by another developer. It helps to catch errors and improve code quality.

4.4.9

Tip

Code review is a process where a developer reviews the code written by another developer. It helps to catch errors and improve code quality.

Code review is a process where a developer reviews the code written by another developer. It helps to catch errors and improve code quality.

```

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

```

??:

```
x = first + ', ' + second
x = '??: ' + name + '; ??: ' + str(n)
```

???????? + ? += ??????????. ??????????????????????. ?? CPython ?????????, ??????????. ?????????????????????, ??????????????????. ??????, ?????????????????, ??????????? '.join ?????. ????????????????? io.StringIO ?????. ????????????????? (amortized) ?????.

??:

```
items = ['<table>']
for last_name, first_name in employee_list:
    items.append('<tr><td>%s, %s</td></tr>' % (last_name, first_name))
items.append('</table>')
employee_table = ''.join(items)
```

??:

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

?????????????????????????????. ?? ' ?? " ??????????. ?????????????????????, ?????????????.

??:

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

(???: ?? "I'm" ??)

??:

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

????????????? """ ?? '?. ?????????? ' ?????????????, ?????????????????????????????? '?. ????, ?????????????? """.

?????????????????????????????. ?????????????????????, ?????????????????, ??? textwrap.dedent() ?????????????.

??:

```
long_string = """?????.
?????.
"""
```

??:

```
long_string = """????????????????,
?????."""
long_string = ("????????????????,\n" +
               "?????.")
```

(continues on next page)

(continued from previous page)

```
long_string = ("XXXXXXXXXXXXXXXXX,\n"
              "XXXXXX.")
```

```
import textwrap

long_string = textwrap.dedent("""\
    XXXX, XX textwrap.dedent()
    XXXXXXXXXXXXXXXXXXXX.""")
```

XX, XXXXXXXXXXXXXXXX XXXXXXXX. XX, XXXXXXXXXXXXXXXX (literal) X XXXXXXXX.

XX

XXXXXXXXXXXXXXXXXXXX (XX % XXX) XXXXX: XXXXXXXXXXXXXXX (XX f-string!) XXXXXXXX, XXXXXXXXXXXXXXXXXXXX. XXXXXXXXXXXXXXXXXXXXXXXXXXXX, XXXXXXXXXXXX. XXXXXXXXXXXXXXXXXXXXXXXXXXXX.

XXX

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

```
import os
from absl import logging

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

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

XX:

```
import os
from absl import logging

logging.info('XXX $PAGER X:')
logging.info(os.getenv('PAGER', default=''))

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

XXXX

XXXX (XX: XX ValueError XXXXXXXXXXXXXXXXXXXX) XXXXXXXXXXXXX:

1. XXXXXXXXXXXXXXXXXXXX.
2. XXXXXXXXXXXXXXXXXXXX.
3. XXXXXXXXXXXXX (XXXXXXXX, XXX grepping).

XX:

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

(continues on next page)

(continued from previous page)

```
try:
    os.rmdir(workdir)
except OSError as error:
    logging.warning('Failed to remove directory (%s: %r): %r',
                    error, workdir)
```

Tip:

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

try:
    os.rmdir(workdir)
except OSError:
    # Error: OSError
    # OSError, OSError.
    logging.warning('Failed to remove directory: %s', workdir)

try:
    os.rmdir(workdir)
except OSError:
    # Error: OSError, OSError `workdir` error.
    # OSError, OSError workdir = '%s'. OSError:
    # "OSError."
    logging.warning('Failed to remove directory.', workdir)
```

4.4.10 `socket` (socket) `socket`

Tip

For more information, see the `socket` module documentation (Python 3.9) `socket`.
 The `socket` module provides `mmap` `h5py` `matplotlib.pyplot` `matplotlib`.

For more information, see the `socket` module documentation.

1. `socket` module documentation, `socket`. `socket` module documentation (Python 3.9), `socket`.
2. `socket` module documentation, `socket` (unmont) `socket`.
3. `socket` module documentation, `socket` module documentation. `socket` module documentation, `socket`, `socket`.

Tip, `socket` (Python 3.9) `socket` (destruct) `socket`, `socket` module documentation.

1. `socket` (runtime) `socket` `__del__` `socket`. `socket` Python `socket` module documentation (Python 3.9), `socket`, `socket`.
2. `socket` (runtime) (Python 3.9) `socket`, `socket` (exception tracebacks) `socket`.

`socket` (finalizer) `socket` module documentation. `socket` module documentation (Python 3.9) `socket` `socket`.

Tip “with” `socket` `socket`:

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

with urllib.urlopen("https://www.python.org/") as front_page:

```
import contextlib

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

(context) contextlib.closing(), urllib.urlopen()

4.4.11 TODO (??) ??

Tip

TODO (??) ??.

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

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

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

TODO "????", "20091111" "???? XML ??, ???", ?????.

4.4.12 ?? (import) ????

Tip

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

?:

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

?:

```
import os, sys
```

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

1. ?? Python ? __future__. ??:

```
from __future__ import annotations
```

???? __future__ ??:

2. Python:

```
import sys
```

3. TensorFlow:

```
import tensorflow as tf
```

4. OtherProject.AI:

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

5. MyProject.Backend.Hgwells:

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

Python, TensorFlow, OtherProject.AI, MyProject.Backend.Hgwells.

from path import ... path)

```
import collections
import queue
import sys

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

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

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

4.4.13

Tip

try / except, try except

if else

:

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


??:

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

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

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

4.4.14 ???? (getter) ???? (setter)

 **Tip**
 ??????????, ?????????? (??????? accessor ?????? mutator) ??????????????????, ??????.

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


???????????????????????????????????? (attribute), ??????????????????. ????, ?????????????????????????????, ??????????. ?????????????????????????????????????, ?????????????????????????????????????, ????????? (property) ??.

(??? : ?????????????????????????????????????, ?????????????: ?????????????????????! ??, ?????????????, ??????Pythonic????)

????????????????????????, ?? get_foo() ? set_foo().

????????????????????????, ?????????????????/?????????????. ?????????????????????????????????????, ?????????????????????????.

4.4.15 ??

 **Tip**
 ????: module_name; ???: package_name; ????: ClassName; ????: method_name; ????: Exception-Name; ????: function_name, query_proper_noun_for_thing, send_acronym_via_https; ??????: GLOBAL_CONSTANT_NAME; ??????: global_var_name; ????: instance_var_name; ??????: function_parameter_name; ??????: local_var_name.

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

??? .py ??????????. ?????????.

??????????

1. ?????????????, ?????????????????:
 1. ????????? (??, i, j, k, v ??).
 2. ? try/except ????????????? e.
 3. ? with ????????????????? f.

4. `constrain` (type variable, `_T = TypeVar("_T"), _P = ParamSpec("_P")`).
2. `__double_leading_and_trailing_underscore__` (Python).
4. `id_to_name_dict`.

Python

1. “(Internal)”
2. `__`
3. `dunder` (name mangling).
4. Java
5. `CapWords`, `lower_with_under.py`. `CapWords.py`
6. PEP 8, `test_<>_<>`. `CapWords`, `test`

Python

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

Python Guido

Table 1:

4.4.17

 Tip

4.4.18 (type annotation)

1. PEP-484 .
2. self cls

```
@classmethod
def create(cls: Type[_T]) -> _T:
    return cls()
```

3. __init__ (None).
4. Any.
5.
 1. API.
 2. , ,
 3. ().
 4. .
 5. .

(signature) . ,

```
def my_method(
    self,
    first_var: int,
    second_var: Foo,
    third_var: Bar | None,
) -> int:
    ...
```

, . , ,

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

Annotations, Annotations, Annotations, Annotations, Annotations def

:

```
def my_method(
    self,
    other_arg: MyLongType | None,
) -> tuple[MyLongType1, MyLongType1]:
    ...
```

.

:

```
def my_method(
    self,
    first_var: int,
    second_var: int) -> dict[OtherLongType, MyLongType]:
    ...
```

pylint Annotations, Annotations, Annotations.

:

```
def my_method(self,
    other_arg: MyLongType | None,
) -> dict[OtherLongType, MyLongType]:
    ...
```

Annotations, Annotations. Annotations. Annotations.

```
def my_method(
    self,
    first_var: tuple[list[MyLongType1],
                    list[MyLongType2]],
    second_var: list[dict[
        MyLongType3, MyLongType4]],
) -> None:
    ...
```

Annotations, Annotations (alias) Annotations. Annotations.

:

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

:

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

(forward declaration)

from __future__ import annotations

Example:

```
from __future__ import annotations

class MyClass:
    def __init__(self, stack: Sequence[MyClass], item: OtherClass) -> None:

class OtherClass:
    ...
```

```
class MyClass:
    def __init__(self, stack: Sequence['MyClass'], item: 'OtherClass') -> None:

class OtherClass:
    ...
```

Example:

PEP-008, =

Example:

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

Example:

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

NoneType

Python, NoneType, None, Union, Optional, Union

X | None, PEP 484, a: str = None, a: str | None = None,

Example:

```
#
def modern_or_union(a: str | int | None, b: str | None = None) -> str:
    ...
# Union / Optional.
def union_optional(a: Union[str, int, None], b: Optional[str] = None) -> str:
    ...
```

Example:

```
# Union Optional.
def nullable_union(a: Union[None, str]) -> str:
    ...
# Optional.
def implicit_optional(a: str = None) -> str:
    ...
```

alias

Python 3.10 introduced the `typing.TypeAlias` type. It is used to define a type alias for an existing type. For example, you can define a type alias for a tuple of TensorFlow tensors:

`typing.TypeAlias` Python 3.10

```
from typing import TypeAlias

_LossAndGradient: TypeAlias = tuple[tf.Tensor, tf.Tensor]
ComplexTFMap: TypeAlias = Mapping[str, _LossAndGradient]
```

pytype

`pytype` # type: ignore

pytype (disable=attribute-error)

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

SomeUndecoratedFunction

`SomeUndecoratedFunction`

Python 3.6 introduced the `typing.TypeAlias` type. It is used to define a type alias for an existing type. For example, you can define a type alias for a function:

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

Python 3.6

Python 3.6 introduced the `typing.TypeAlias` type. It is used to define a type alias for an existing type. For example, you can define a type alias for a function:

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

list

Python 3.5 introduced the `typing.List` type. It is used to define a type alias for a list. For example, you can define a type alias for a list of integers:

(Python 3.5 introduced the `typing.List` type, `python`, `list`, `tuple`, `list`, `tuple`)

```
a: list[int] = [1, 2, 3]
b: tuple[int, ...] = (1, 2, 3)
c: tuple[int, str, float] = (1, "2", 3.5)
```

type variable

Python 3.5 introduced the `typing.TypeVar` type. It is used to define a type variable. For example, you can define a type variable for a function:

`TypeVar`

```
from collections.abc import Callable
from typing import ParamSpec, TypeVar

_P = ParamSpec("_P")
_T = TypeVar("_T")
...
def next(l: list[_T]) -> _T:
    return l.pop()

def print_when_called(f: Callable[_P, _T]) -> Callable[_P, _T]:
    def inner(*args: _P.args, **kwargs: _P.kwargs) -> R:
        print('called')
```

(continues on next page)

(continued from previous page)

```

    return f(*args, **kwargs)
return inner

```

TypeVar

```

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

```

AnyStr typing bytes str

```

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

```

(x bytes str.)

:

- 1.
- 2.

:

```

_T = TypeVar("_T")
_P = ParamSpec("_P")
AddableType = TypeVar("AddableType", int, float, str)
AnyFunction = TypeVar("AnyFunction", bound=Callable)

```

:

```

T = TypeVar("T")
P = ParamSpec("P")
_T = TypeVar("_T", int, float, str)
_F = TypeVar("_F", bound=Callable)

```

typing.Text. Python 2/3

str bytes

```

#
def deals_with_text_data(x: str) -> str:
    ...
#
def deals_with_binary_data(x: bytes) -> bytes:
    ...

```

AnyStr.

typing collections.abc, typing collections.abc

(continued from previous page)

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

Generics

Example, `Sequence`. `Any`.

Example:

```
def get_names(employee_ids: Sequence[int]) -> Mapping[int, str]:
    ...
```

Example:

```
# Example get_names(employee_ids: Sequence[Any]) -> Mapping[Any, Any]
def get_names(employee_ids: Sequence) -> Mapping:
    ...
```

Example `Any`, `Sequence`, `TypeVar`.

Example:

```
def get_names(employee_ids: Sequence[Any]) -> Mapping[Any, str]:
    """Example ID"""
```

Example:

```
_T = TypeVar('_T')
def get_names(employee_ids: Sequence[_T]) -> Mapping[_T, str]:
    """Example ID"""
```

4.5

Example.

Example, `Sequence`. `Any`, `Sequence`, `TypeVar`. `Any`, `Sequence`, `TypeVar`.

Example `Any`, `Sequence`, `TypeVar`. `Any`, `Sequence`, `TypeVar`. `Any`, `Sequence`, `TypeVar`.

SHELL [?] - [?]

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
- \${PHPESTATUS} 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

STDPERR

```
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'

#####
# 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

here document

```
# DO use 'here document's
cat <<END;
I am an exceptionally long
string.
END

# Embedded newlines are ok too
long_string="I am an exceptionally
long string."
```

5.6.3

 Tip

'&&'

```
# All fits on one line
command1 | command2

# Long commands
command1 \
| command2 \
| command3 \
| command4
```

5.6.4

Tip

```
do,; then while,for,if
```

```
shell ; do,; then if/for/while else
```

```
for
```

```
for dir in ${dirs_to_cleanup}; do
  if [[ -d "${dir}/${ORACLE_SID}" ]]; then
    log_date "Cleaning up old files in ${dir}/${ORACLE_SID}"
    rm "${dir}/${ORACLE_SID}/*"
    if [[ "$?" -ne 0 ]]; then
      error_message
    fi
  else
    mkdir -p "${dir}/${ORACLE_SID}"
    if [[ "$?" -ne 0 ]]; then
      error_message
    fi
  fi
done
```

5.6.5 case

Tip

-
- ; ;
- ; ;


```
case esac ;& ;&
```

```
case "${expression}" in
  a)
    variable="..."
    some_command "${variable}" "${other_expr}" ...
    ;;
  absolute)
    actions="relative"
    another_command "${actions}" "${other_expr}" ...
    ;;
  *)
    error "Unexpected expression '${expression}'"
    ;;
esac
```

```
;;
;;
```

```
verbose='false'
aflag=''
bflag=''
files=''
while getopts 'abf:v' flag; do
  case "${flag}" in
    a) aflag='true' ;;
    b) bflag='true' ;;
    f) files="${OPTARG}" ;;
    v) verbose='true' ;;
    *) error "Unexpected option ${flag}" ;;
  esac
done
```

5.6.6

 Tip

`var=${var}`

1. `var=${var}`
2. `var=${var}`
3. `var=${var}`

```
# 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
```

(continues on next page)

(continued from previous page)

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

5.7

5.7.1

 Tip

```
$(command)
```

```
$(command)
```

```
# This is preferred:
var="$(command "$(command1) ")"
```

```
# This is not:
var="\`command \`${command1}\`"
```

5.7.2 test

 Tip

```
[[ ... ]] [ ,test, /usr/bin/[
```

```
[[ ]] [ ... ] [ ... ] [ ... ]
```

```
# This ensures the string on the left is made up of characters in the
# alnum character class followed by the string name.
```

(continues on next page)

(continued from previous page)

```
# 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 `rm`

Tip

`rm -rf /`

`rm -rf *`

```
# 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 ./ *
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

```
while read line; do
  your_command
done
```

```
while read line; do
  your_command
done
```

```
last_line='NULL'
your_command | while read line; do
  last_line="${line}"
done
# This will output 'NULL'
echo "${last_line}"
```

```
for value in $(command); do
  your_command
done
```

```
total=0
# Only do this if there are no spaces in return values.
for value in $(command); do
  total+="${value}"
done
```

```
while read count filename; do
  your_command
done <<(your_command | uniq -c)
```

```
total=0
last_file=
while read count filename; do
  total+="${count}"
  last_file="${filename}"
done <<(your_command | uniq -c)
# This will output the second field of the last line of output from
# the command.
echo "Total = ${total}"
echo "Last one = ${last_file}"
```

```
while read src dest type opts rest; do
  if [[ ${type} == "nfs" ]]; then
    echo "NFS ${dest} maps to ${src}"
  fi
done
```

```
# 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
[redacted]

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"
```

(continues on next page)

(continued from previous page)

```
# 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

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 PIPESTA-
TUS [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++

JAVASCRIPT [?] - [?]

6.1 0. [?]

[?] [?] Git [?].

[?]

- Google [?] - [?]

[?]

- Google Style Guide

6.2 [?]

[?]Google[?]JavaScript[?]JavaScript[?]

6.3 Javascript[?]

6.3.1 var[?]

[?] var [?]

[?]

[?] var [?]var[?]docu
var [?]

6.3.2

- NAMES_LIKE_THIS
- @const
- IE const

CONSTANT_VALUE_CASE

number string boolean

@const const

IE const

@const

@const CONSTANT_VALUE_CASE

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

1260 @const

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

6.3.3

```

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

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

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

// 2. IE Firefox.
//.
[normalVersion, ffVersion][isIE]();

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

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


1. js4242
2. “no sush property in undefined” x[normalVersion, ffVersion][isIE]()
3. die resultOfOperation() NaN THINGS_TO_EAT die()

js”)”}”]”

??????

??

```

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

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

```

6.3.4

????

??

6.3.5

??????

??????

```

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

```

??ECMAScript????ECMA-262 1314????????????EcmaScript??ECM

```

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

```

6.3.6

??????

??

6.3.7

??????

??

6.3.8

string.charAt(3) string[3]

6.3.9

```
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';
```

6.3.10

JavaScript class B class D

Closure goog.inherits()

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

D.prototype.method =function() {
    ...
};
```

6.3.11

```
/** */ function SomeConstructor() { this.someProperty = 1; } Foo.prototype.  
someMethod = function() { ... };
```

“new”

```
Foo.prototype.bar = function() {  
  /* ... */  
};
```

```
/** @constructor */  
function Foo() {  
  this.bar = value;  
}
```

JavaScript “”

6.3.12

this.foo = null

```
o.prototype.dispose = function() {  
  this.property_ = null;  
};
```

```
Foo.prototype.dispose = function() {  
  delete this.property_;  
};
```

JavaScript (key in obj) if

6.3.13

JavaScript

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 */ }
}
```

6.3.14 eval()

RPC

eval() eval() eval()

```
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', 'https://chat.google.com/isUserOnline?user=' + user, false);
xmlhttp.send('');
// 
// userOnline = true;
if (xmlhttp.status == 200) {
    eval(xmlhttp.responseText);
}
// userOnline true
```



```
function printArray(arr) {
  var l = arr.length;
  for (var i = 0; i < l; i++) {
    print(arr[i]);
  }
}
```

6.3.18

..... JS Date RegExp String

6.3.19

```
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.';
```

6.3.20

```
// 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.
```

(continues on next page)

6.4 Javascript

6.4.1

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

-
-

opt_

var_args var_args arguments

@param

getter/setter

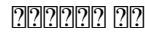
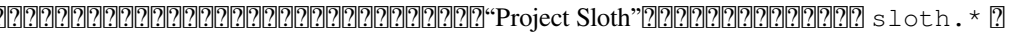
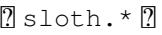

EcmaScript 5 getter/setter

```
/**
 *--.
 */
var foo = { get next() { return this.nextId++; } };
};
```


getter/setter getFoo() setFoo(value)
isFoo()

JavaScript

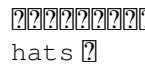
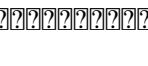
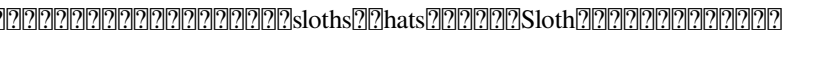


JavaScript

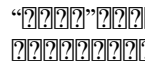
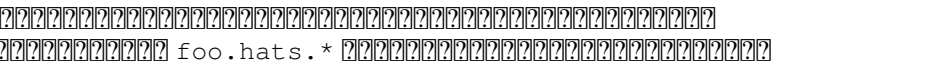
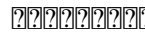
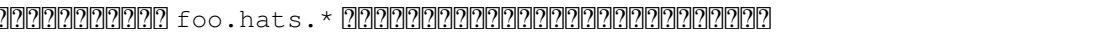
 “Project Sloth” sloth.* 

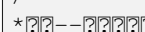
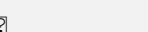
```
var sloth = {};  
  
sloth.sleep = function() {  
  ...  
};
```

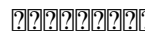
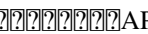
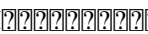
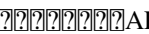
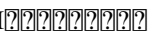
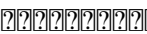
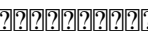
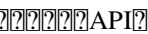
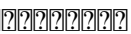
JavaScript the Closure Library  Dojo toolkit 

```
goog.provide('sloth');  
  
sloth.sleep = function() {  
  ...  
};
```


slothshatsSloth sloth.
hats 

“” foo.hats.*
 foo.hats.* 

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

APIAPIAPIAPIAPIAPIAPIAPI

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

```

??


```

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

// [REDACTED]
this.rows_ = [
  "Slartibartfast" <fjordmaster@magrathea.com>',
  "Zaphod Beeblebrox" <theprez@universe.gov>',
  "Ford Prefect" <ford@theguide.com>',
  "Arthur Dent" <has.no.tea@gmail.com>',
  "Marvin the Paranoid Android" <marv@googlemail.com>',
  'the.mice@magrathea.com'
];

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

```

[REDACTED]

```

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

```

[REDACTED]

```

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

```

[REDACTED]

[REDACTED]80 [REDACTED]80 [REDACTED]

```

// [REDACTED]80 [REDACTED]
// [REDACTED]
goog.foo.bar.doThingThatIsVeryDifficultToExplain = function(
  veryDescriptiveArgumentNumberOne, veryDescriptiveArgumentTwo,
  tableModelEventHandlerProxy, artichokeDescriptorAdapterIterator) {
  // ...
};

// [REDACTED]
// [REDACTED]
goog.foo.bar.doThingThatIsVeryDifficultToExplain = function(

```

(continues on next page)

goog.scope

goog.scope the Closure Library

goog.scope

```
goog.scope(function() {
  goog.provide('goog.ui.Button');
  goog.require('goog.ui.Button');
  // goog.scope
});
```

C++ goog.scope

```
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.getElementById('my-button'));
  };
  ...
}); // goog.scope
```



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

thisIsAVeryLongVariableName =
  hereIsAnEvenLongerOtherFunctionNameThatWillNotFitOnPrevLine();

thisIsAVeryLongVariableName = 'expressionPartOne' + someMethodThatIsLong() +
  thisIsAnEvenLongerOtherFunctionNameThatCannotBeIndentedMore();
```

(continues on next page)

(continued from previous page)

```

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.
var z = a ?
    moreComplicatedB :
    moreComplicatedC;

```

??????????????

```

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

```

6.4.6

```

delete type of void return throw case in new

```

6.4.7

```

' ' "
HTML

```

```

var msg = 'This is some HTML';

```

6.4.8

```

@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_();
}
AA_init_();

```

```

@private
@protected
C++ JAVA private protected C++

```

```

// File 1.
/** @constructor */

```

(continues on next page)

(continued from previous page)

```

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();
};
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

6.4.9 JavaScript

JSDoc EcmaScript 4

JavaScript

ES4 JavaScript JSDoc

ES4

Code	Language	Annotation	Output
<code>{null} {undefined} {boolean} {number} {string}</code>	JavaScript		
<code>{Object} {Function} {EventTarget} EventTarget</code>		@constructor	
<code>{goog.events.EventType} goog.events.EventType</code>		JSDoc @interface JS- Doc ES4 @enum	
<code>{Array.<string>} {Object.<string, number>}</code>		Java	
<code>{(number boolean)}</code>		A?B? { (number, boolean) }	
<code>{?number}</code>		syntactic sugar	{ number? }
<code>{!Object}</code>		null	{ Object! }
<code>{myNum: number, myObject}</code>		myNum number myObject length	
<code>{function(string, boolean)}</code>			
<code>{function(): number}</code>			
<code>{function(this:goog.ui.Menu, string)}</code>			
<code>{function(new:goog.ui.Menu, string)}</code>			
<code>{function(string, ... [number]): number}</code>			
<code>@param {...number}</code>			
<code>@param var_args</code>			
<code>{function(?string=, number=)}</code>			
<code>@param {number=}</code>			

JavaScript

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

???

??

```
/** @type {number} */ (x)
```

????????????????

??Javascript??

????????????????????????????????

```
/**
 * ?????????
 * @param {Object} value???
 * @constructor
 */
function MyClass(value) {
  /**
   * Some value.
   * @type {Object}
   * @private
   */
  this.myValue_ = value;
}
```

???? myValue_ ??????null???? myValue_ ??????null, ???????:

```
/**
 * ???null????????
 * @param {!Object} value???
 * @constructor
 */
function MyClass(value) {
  /**
   * Some value.
   * @type {!Object}
   * @private
   */
  this.myValue_ = value;
}
```

???????????????????? MyClass ??????null????????????

????????????????????????undefined????????????????????

```
/**
 * ?????????????
 * @param {Object=} opt_value????????
 * @constructor
 */
function MyClass(opt_value) {
  /**
   * Some value.
   * @type {Object|undefined}
   * @private
   */
}
```

(continues on next page)

(continued from previous page)

```

    this.myValue_ = opt_value;
}

```

myValue_ null undefined
 : opt_value {Object=} {Object|undefined} unde-
 fined

```

/**
 * @param {!Object} nonNull null
 * @param {Object} maybeNull null
 * @param {!Object=} opt_nonNull null
 * @param {Object=} opt_maybeNull null
 */
function strangeButTrue(nonNull, maybeNull, opt_nonNull, opt_maybeNull) {
    // ...
};

```

```

/**
 * @param {string} tagName
 * @param {(string|Element|Text|Array.<Element>|Array.<Text>)} contents
 * @return {!Element}
 */
goog.createElement = function(tagName, contents) {
    ...
};

```

@typedef

```

/** @typedef {(string|Element|Text|Array.<Element>|Array.<Text>)} */
goog.ElementContent;

/**
 * @param {string} tagName
 * @param {goog.ElementContent} contents
 * @return {!Element}
 */
goog.createElement = function(tagName, contents) {
    ...
};

```



```

/**
 * This is NOT the preferred indentation method.
 * @param {string} foo This is a param with a description too long to fit in
 *                   one line.
 * @return {number} This returns something that has a description too long to
 *                   fit in one line.
 */
project.MyClass.prototype.method = function(foo) {
  return 5;
};

```

JSDoc HTML

JavaDoc, JSDoc HTML <code> <pre> <tt> <a>

JSDoc

```

/**
 * Computes weight based on three factors:
 *   items sent
 *   items received
 *   last timestamp
 */

```

Computes weight based on three factors: items sent items received items received last timestamp

```

/**
 * Computes weight based on three factors:
 * <ul>
 * <li>items sent
 * <li>items received
 * <li>last timestamp
 * </ul>
 */

```

JavaDoc doc

```

/**
 * @fileoverview Description of file, its uses and information
 * about its dependencies.
 */

```

Class

Example 1: A class that inherits from a base class and has a constructor.

```
/**
 * Class making something fun and easy.
 * @param {string} arg1 An argument that makes this more interesting.
 * @param {Array.<number>} arg2 List of numbers to be processed.
 * @constructor
 * @extends {goog.Disposable}
 */
project.MyClass = function(arg1, arg2) {
  // ...
};
goog.inherits(project.MyClass, goog.Disposable);
```

Example 2

Example 2: A method that operates on an instance of a class.

```
/**
 * Operates on an instance of MyClass and returns something.
 * @param {project.MyClass} obj Instance of MyClass which leads to a long
 *   comment that needs to be wrapped to two lines.
 * @return {boolean} Whether something occurred.
 */
function PR_someMethod(obj) {
  // ...
}
```

Example 3

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

JSDoc

??	??????	??
<p>@author</p>	<p>@author username@google.com (first last) ???</p> <pre>/** * @fileoverview * Utilities for * handling textareas. * @author kuth@google. * com (Uthur Pendragon) */</pre>	<p>???????????????????? @fileoverview ????</p>
<p>@code</p>	<pre>{@code ...} ???</pre> <div style="border: 1px solid gray; padding: 5px; margin: 5px 0;"> <pre>/**</pre> </div> <ul style="list-style-type: none"> • Moves to the next position in the selection. • Throws <code>{@code goog.iter.StopIteration}</code> when it • passes the end of the range. • @return <code>{Node}</code> The node at the next position. <pre>*/ goog.dom.RangeIterator.prc = function() { // ... };</pre>	<p>????????????????????????????????</p>

continues on next page

Table 1 – continued from previous page

??	??????	??
@const	<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>
@constructor	<pre> @constructor ??? /** * A rectangle. * @constructor */ function GM_Rect() { ... } </pre>	<pre> ???????????????????????????????? </pre>
@define	<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' ?? ?? true ? </pre>

continues on next page

Table 1 – continued from previous page

??	??????	??
<p>@deprecated</p>	<p>@deprecated Description ??? /** * 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) { // ... };</p>	<p>??</p>
<p>@dict</p>	<p>@dict Description ??? /** * @constructor * @dict */ function Foo(x) { this['x'] = x; } var obj = new Foo(123); var num = obj.x; // ↪warning (** @dict */ { x: 1 } ↪).x = 123; // ↪warning</p>	<p>?????? (?????Foo)????? @dict ?????????????????????? Foo ??</p>
<p>@enum</p>	<p>@enum {Type} ??? /** * Enum for tri-state ↪values. * @enum {number} */ project.TriState = { TRUE: 1, FALSE: -1, MAYBE: 0 };</p>	

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

??	??????	??
@interface	<pre> @interface ??? /** * A shape. * @interface */ function Shape() {}; Shape.prototype.draw = ↳function() {}; /** * A polygon. * @interface * @extends {Shape} */ function Polygon() {}; Polygon.prototype. ↳getSides = ↳function() {}; </pre>	????????????????????
@lends	<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>
@license or @preserve	<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>

continues on next page

Table 1 – continued from previous page

??	??????	??
@param	<pre> @param {Type} varname De- scription /** * 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>	<pre> ???????????????????????????????????? ????? ????????????????????????????????? </pre>
@private	<pre> @private @private {type} /** * Handlers that are * listening to this * logger. * @private {!Array. * <Function>} */ this.handlers_ = []; </pre>	<pre> ???????????????????????????????????? ????? ????????? @private ????????????????????????????? </pre>

continues on next page

Table 1 – continued from previous page

@protected	<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>	<pre> `" <https://google-styleguide. googlecode.com/svn/trunk/javascriptguide.xml# Visibility__private_and_protected_fields_>`_ </pre>
@return	<pre> @return {Type} Description /** * @return {string} The * ↪hex ID of the last * ↪item. */ goog.Baz.prototype. ↪getLastId = ↪function() { // ... return id; }; </pre>	<pre> @return </pre>
@see	<pre> @see Link /** * Adds a single item, * ↪recklessly. * @see #addSafely * @see goog.Collect * @see goog. * ↪RecklessAdder#add * ... </pre>	<pre> </pre>

continues on next page

Table 1 – continued from previous page

??	??????	??
@struct	<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>	<pre> ???????????????? Foo ????? @struct ????????????????- Foo?????????????Foo?? </pre>
@supported	<p>@supported Description</p> <pre> /** * @fileoverview Event↳ ↳Manager * Provides an↳ ↳abstracted interface↳ ↳to the * browsers' event↳ ↳systems. * @supported So far↳ ↳tested in IE6 and↳ ↳FF1.5 */ </pre>	<pre> ?? </pre>
@suppress	<p>@suppress {warning1 warning2}</p> <pre> /** * @suppress ↳{deprecated} */ function f() { ↳ ↳deprecatedVersionOfF(); ↳ } </pre>	<pre> ?? </pre>

continues on next page

Table 1 – continued from previous page

??	??????	??	
@template	<pre> @template ???</pre>	<pre> ???</pre>	
@this	<pre> @this Type @this {Type} ???</pre>	<pre> 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>
@type	<pre> @type Type @type {Type} ???</pre>	<pre> /** • The message hex ID. • @type {string} */ var hexId = hexId;</pre>	

continues on next page

Table 1 – continued from previous page

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

JSDoc JSDoc Toolkit

- @arguments
- @argument
- @borrows
- @class
- @constant
- @constructs
- @default
- @event
- @example
- @field
- @function
- @ignore
- @inner
- @link
- @memberOf
- @name
- @namespace
- @property
- @public
- @requires
- @returns
- @since
- @static
- @version

6.4.11 goog.provide

Example 1

Code snippet for Example 1

Example 1

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

Example 2

```
goog.provide('namespace.MyClass');
goog.provide('namespace.MyClass.Enum');
goog.provide('namespace.MyClass.InnerClass');
goog.provide('namespace.MyClass.TypeDef');
goog.provide('namespace.MyClass.CONSTANT');
goog.provide('namespace.MyClass.staticMethod');
```

Example 3

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

6.4.12

Example 1

Code snippet for Example 1

6.4.13

JavaScript

True/False

Examples of false

- null
- undefined
- ""
- 0

Examples of true

- "0"
- []
- {}

Example 2

&& ||

??

“||” ????? ‘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^2)

```
var paragraphs = document.getElementsByTagName('p');
for (var i = 0; i < paragraphs.length; i++) {
  doSomething(paragraphs[i]);
}
```

??????

```
var paragraphs = document.getElementsByTagName('p');
for (var i = 0, paragraph; paragraph = paragraphs[i]; i++) {
  doSomething(paragraph);
}
```

????????(????????false) ?????

????????firstChild?nextSibling??????????

```
var parentNode = document.getElementById('foo');
for (var child = parentNode.firstChild; child; child = child.nextSibling) {
  doSomething(child);
}
```


TYPESCRIPT `???`

7.1 0. `??`

`???`

`????????????????????` Git `??`.

`??`

- Google `??????????` - `???`

`?????`

- Google Style Guide

7.2 `??`

`????????????????` TypeScript `??` TypeScript
`??`

`??`

7.2.1 `????`

`????? ???? ???` RFC 2119 `??????????` `???? ? ??` `????? ?? ?` `????????????????????????????????` `??`
`????????????????????????????????`

7.2.2 `????`

`?????????? ???? ???`

7.2.3



2024 年 02 月 29 日



- Frank Li

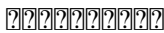


Google TypeScript Style Guide



 TypeScript 


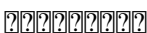
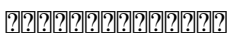
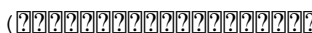
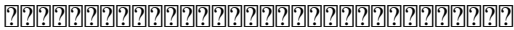






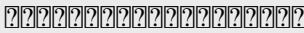


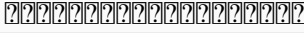


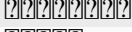



- 2021 年 09 月 02 日  Frank Li 
- 2024 年 02 月 29 日  Frank Li 

7.3

7.3.1



 TypeScript  ASCII  ( `(\\)\w`)


	
 UpperCamelCase 	
 lowerCamelCase 	
 CONSTANT_CASE 	
 #ident 	

??

loadHttpRequest

\$\$\$ \$

Observable

???

Array<T> UpperCamelCase

???

Closure testSuites xUnit testX_whenY_doesZ()

_????

```
const [a, , b] = [1, 5, 10]; // a <- 1, b <- 10
```

???

lowerCamelCase snake_case

```
import * as fooBar from './foo_bar';
```

- jQuery \$
- three.js THREE

??

CONSTANT_CASE deep frozen

```
const UNIT_SUFFIXES = {
  'milliseconds': 'ms',
  'seconds': 's',
};
// UNIT_SUFFIXES
//
//
```

```
class Foo {
  private static readonly MY_SPECIAL_NUMBER = 5;

  bar() {
    return 2 * Foo.MY_SPECIAL_NUMBER;
  }
}
```

??

??

??

??

?? const
????????????????? readonly ?????????????????????

```
const {Foo} = SomeType;
const CAPACITY = 5;

class Teapot {
  readonly BrewStateEnum = BrewStateEnum;
  readonly CAPACITY = CAPACITY;
}
```

?????

TypeScript ??? Testing Blog??

????????????????

- ?????????????????????????? _ ?????????
- ?????????????? opt_ ???
 - ?????????????????????? 1.6. #include ????????? ?
- ?? IMyInterface ?? MyFooInter-
face ?? TodoItem ?????????? JSON
??? TodoItemStorage ??
- ?? Observable ?????????????????????? \$???

7.3.1

API i j

7.3.2

UTF-8

ASCII Unicode ∞ Unicode \u221e

```
// const units = 'µs';
// const output = '\uffeff' + content; // Byte Order MarkBOM
```

```
// const units = '\u03bcs'; // Greek letter mu, 's'
// const output = '\uffeff' + content;
```

7.3.3

JSDoc

Typescript JSDoc /** ... */

- /** JSDoc */
- /**

JSDoc

JSDoc

JSDoc JavaScript JavaScript


```
/** JSDoc */
/** JSDoc */
```

TypeScript

@param @return implements enum private @implements @enum @private

@override

TypeScript @override @override


```
//
/** @param fooBarService Foo Bar */
```

@param @return

```
/**
 * POST
 * @param amountLitres
 */
brew(amountLitres: number, logger: Logger) {
  // ...
}
```



```
class Foo {
  constructor(private readonly bar: Bar) { }
}
```

Foo Bar bar

JSDoc @param

```
/**
class ParamProps {
  /**
   * @param percolator
   * @param beans
   */
  constructor(
    private readonly percolator: Percolator,
    private readonly beans: CoffeeBean[]) {}
}
```

```
/**
class OrdinaryClass {
  /** brew() */
```

(continues on next page)

(continued from previous page)

```

nextBean: CoffeeBean;

constructor(initialBean: CoffeeBean) {
    this.nextBean = initialBean;
}
}

```

Annotations

Annotations are used to provide additional information about code elements.

```

// Example of an annotation
new Percolator().brew(/* amountLitres= */ 5);

// Example of an annotation with a parameter
new Percolator().brew({amountLitres: 5});

```

```

/** @link CoffeeBrewer */
export class Percolator implements CoffeeBrewer {
    /**
     * @param amountLitres amount of coffee to brew
     */
    brew(amountLitres: number) {
        // TODO(b/12345): Implement brewing
    }
}

```

Annotations

Annotations are used to provide additional information about code elements.

Annotations are used to provide additional information about code elements.

```

// Example of an annotation
@Component({
    selector: 'foo',
    template: 'bar',
})
export class FooComponent {}

```

Annotations

```

/** @link "bar" */
@Component({
    selector: 'foo',
    template: 'bar',
})
export class FooComponent {}

```

7.4

7.4.1

Example code for 7.4.1

- Example 1
- Example 2
- TypeScript `public` vs `readonly` vs `public`

```
class Foo {
  public bar = new Bar(); // public
  constructor(public readonly baz: Baz) {} // readonly baz
  →public public public
}
```

```
class Foo {
  bar = new Bar(); // public
  constructor(public baz: Baz) {} // public
}
```

Example code for 7.4.1

7.4.2

Example code for 7.4.2

```
// 
const x = new Foo();

// 
const x = new Foo();
```

ES2015

```
// 
class UnnecessaryConstructor {
  constructor() {}
}
```

```
// 
class UnnecessaryConstructorOverride extends Base {
  constructor(value: number) {
    super(value);
  }
}
```

```
// 
class DefaultConstructor {
}
```

(continues on next page)

(continued from previous page)

```
// 
class ParameterProperties {
    constructor(private myService) {}
}

// 
class ParameterDecorators {
    constructor(@SideEffectDecorator myService) {}
}

// 
class NoInstantiation {
    private constructor() {}
}
```

7.4.3

#private

#private

```
// 
class Clazz {
    #ident = 1;
}
```

TypeScript

```
// 
class Clazz {
    private ident = 1;
}
```

TypeScript JavaScript ES2015 TypeScript ES2015

readonly

readonly

TypeScript

```
// 
class Foo {
    private readonly barService: BarService;

    constructor(barService: BarService) {
        this.barService = barService;
    }
}
```

(continues on next page)

(continued from previous page)

```
}
}
```

```
// 
class Foo {
  constructor(private readonly barService: BarService) {}
}
```

JSDoc @param

```
// 
class Foo {
  private readonly userList: string[];
  constructor() {
    this.userList = [];
  }
}
```

```
// 
class Foo {
  private readonly userList: string[] = [];
}
```

template AngularJS controller private

public protected Angular Polymer public AngularJS protected

TypeScript obj['foo']

private

obj['foo'] TypeScript

7.4.1 Accessors

Accessors are a way to get and set the value of a property. They are used to control how data is accessed and modified.

```
class Foo {
  constructor(private readonly someService: SomeService) {}

  get someMember(): string {
    return this.someService.someVariable;
  }

  set someMember(newValue: string) {
    this.someService.someVariable = newValue;
  }
}
```

Accessors can be used to wrap an internal property. The wrapped property is marked as `internal` and the accessor is marked as `public` and `readonly`.

```
class Foo {
  private wrappedBar = '';
  get bar() {
    return this.wrappedBar || 'bar';
  }

  set bar(wrapped: string) {
    this.wrappedBar = wrapped.trim();
  }
}
```

```
class Bar {
  private barInternal = '';
  // Accessor for barInternal, bar is public
  get bar() {
    return this.barInternal;
  }

  set bar(value: string) {
    this.barInternal = value;
  }
}
```

7.4.2 Primitive Wrappers

TypeScript provides wrapper classes for the primitive types `String`, `Boolean`, and `Number`. These wrappers allow you to create objects from primitive values. For example, `new Boolean(false)` creates a `Boolean` object with the value `false`.

```
// Primitive wrappers
const s = new String('hello');
const b = new Boolean(false);
const n = new Number(5);
```

```
// Primitive values
const s = 'hello';
const b = false;
const n = 5;
```

7.4.5

TypeScript Array() new

```
// 
const a = new Array(2); // 2 [undefined, undefined]
const b = new Array(2, 3); // 2, 3 [2, 3]
```

from

```
const a = [2];
const b = [2, 3];

// Array(2)
const c = [];
c.length = 2;

// [0, 0, 0, 0, 0]
Array.from<number>({length: 5}).fill(0);
```

7.4.6

TypeScript String() Boolean() new !!

```
const bool = Boolean(false);
const str = String(aNumber);
const bool2 = !!str;
const str2 = `result: ${bool2}`;
```

string

Number() NaN

Tip

Number('') Number(' ') Number('\t') 0 NaN Number('Infinity') Number('-Infinity') Infinity -Infinity

```
const aNumber = Number('123');
if (isNaN(aNumber)) throw new Error(...); // NaN
assertFinite(aNumber, ...); //
```

+ +

```
// 
const x = +y;
```

parseInt parseFloat 12 dwarves 12

```
const n = parseInt(someString, 10); // 
const f = parseFloat(someString); //
```

parseInt

```
if (!/^[a-zA-F0-9]+$/.test(someString)) throw new Error(...);
// 16
// tslint:disable-next-line:ban
const n = parseInt(someString, 16); // parseInt
```

Number() Math.floor Math.trunc

```
let f = Number(someString);
if (isNaN(f)) handleError();
f = Math.floor(f);
```

if for while boolean

```
//
const foo: MyInterface|null = ...;
if (!!foo) {...}
while (!!foo) {...}
```

```
//
const foo: MyInterface|null = ...;
if (foo) {...}
while (foo) {...}
```

```
// 0
if (arr.length > 0) {...}

//
if (arr.length) {...}
```

7.4.7

const let const var

```
const foo = otherValue; // foo const
let bar = someValue; // bar let
```

const let var JavaScript bug TypeScript var

```
//
var foo = someValue;
```

7.4.8

new Error() Error() new

```
// 
throw new Error('Foo is not a valid bar.');
```

```
// 
throw Error('Foo is not a valid bar.');
```

7.4.9

for (... in ...) for (... in ...)

```
// 
for (const x in someObj) {
  // x someObj 
}
```

if for (... of Object.keys(...))

```
// 
for (const x in someObj) {
  if (!someObj.hasOwnProperty(x)) continue;
  // x someObj 
}
```

```
// 
for (const x of Object.keys(someObj)) { // for _of_ 
  // x someObj 
}
```

```
// 
for (const [key, value] of Object.entries(someObj)) { // for _of_ 
  // key someObj 
}
```

7.4.10

for (... in ...) string

```
// 
for (const x in someArray) {
  // x (string) 
}
```

for (... of someArr) for

```
// 
for (const x of someArr) {
  // x
}

```

```
// 
for (let i = 0; i < someArr.length; i++) {
  // for/of
  const x = someArr[i];
  // ...
}

```

```
// 
for (const [i, x] of someArr.entries()) {
  // 
}

```

Array.prototype.forEach Set.prototype.forEach Map.prototype.forEach

```
// 
someArr.forEach((item, index) => {
  someFn(item, index);
});

```

```
let x: string|null = 'abc';
myArray.forEach(() => { x.charAt(0); });

```

x null .
forEach() () => { x.charAt(0); } x
null for-of

7.4.11

[...foo]{...bar}

```
const foo = {
  num: 1,
};

const foo2 = {
  ...foo,
  num: 5,
};

const foo3 = {
  num: 5,
  ...foo,
}

```

(continues on next page)

(continued from previous page)

```
// foo2
foo2.num === 5;

// foo3
foo3.num === 1;
```

undefined

```
// 
const foo = {num: 7};
const bar = {num: 5, ...(shouldUseFoo && foo)}; // undefined
```

```
// fooStrings length {0: 'a', 1: 'b', 2: 'c'}
const fooStrings = ['a', 'b', 'c'];
const ids = {...fooStrings};
```

```
// 
const foo = shouldUseFoo ? {num: 7} : {};
const bar = {num: 5, ...foo};

// 
const fooStrings = ['a', 'b', 'c'];
const ids = [...fooStrings, 'd', 'e'];
```

7.4.12 for / if

for

```
// 
for (let i = 0; i < x; i++) {
  doSomethingWith(i);
  andSomeMore();
}

if (x) {
  doSomethingWithALongMethodName(x);
}
```

```
// 
if (x)
  x.doFoo();
for (let i = 0; i < x; i++)
  doSomethingWithALongMethodName(i);
```

if

```
// 
if (x) x.doFoo();
```

7.4.13 switch

switch default

```
// 
switch (x) {
  case Y:
    doSomethingElse();
    break;
  default:
    // 
}
```

case ...

```
// 
switch (x) {
  case X:
    doSomething();
    // 
  case Y:
    // ...
}
```

```
// 
switch (x) {
  case X:
  case Y:
    doSomething();
    break;
  default: // 
}
```

7.4.14

JavaScript JavaScript

```
// 
if (foo == 'bar' || baz != bam) {
  // 
}
```

```
// 
if (foo === 'bar' || baz !== bam) {
  // 
}
```

null == ? != null ? undefined

```
// 
if (foo == null) {
  // foo null undefined
}
```

7.4.15

function foo() { ... }

const x = function() {...}; TypeScript
const

this

```
//  
function foo() { ... }
```

```
//  
//  
foo = () => 3; //  
  
//  
const foo = function() { ... }
```

function foo() {} doSomethingWith(function() {});

```
interface SearchFunction {  
  (source: string, subString: string): boolean;  
}  
  
const fooSearch: SearchFunction = (source, subString) => { ... };
```

7.4.16

ES6 function

```
//  
bar(() => { this.doSomething(); })
```

```
//  
bar(function() { ... })
```

this function this
this


(continued from previous page)

```
// 
const setTextFn = (e: HTMLElement) => { e.textContent = 'hello'; };
document.body.onclick = setTextFn.bind(null, document.body);
```

QUESTION

```

this 
this
const handler = (x) => { this.listener(x); }; 
const handler = this.listener; handler(x);
```

 Tip

```
// 
class DelayHandler {
  constructor() {
    // 
    // 
    setTimeout(this.patienceTracker, 5000);
  }
  private patienceTracker() {
    this.waitedPatiently = true;
  }
}
```

```
// 
class DelayHandler {
  constructor() {
    // 
    setTimeout(this.patienceTracker, 5000);
  }
  private patienceTracker = () => {
    this.waitedPatiently = true;
  }
}
```

```
// 
class DelayHandler {
  constructor() {
    // 
    setTimeout(() => {
      this.patienceTracker();
    }, 5000);
  }
  private patienceTracker() {
    this.waitedPatiently = true;
  }
}
```


7.4.17

ASI Bug clang-format

7.4.18 @ts-ignore

@ts-ignore any

7.4.19

x as SomeType y!

```
// (x as Foo).foo();
y!.bar();
```

```
//
// Foo
if (x instanceof Foo) {
  x.foo();
}

if (y) {
  y.bar();
}
```

```
//
// x Foo.....
(x as Foo).foo();

// y null.....
y!.bar();
```

??????

???????? as ???

```
// ??????
const x = (<Foo>z).length;
const y = <Foo>z.length;
```

```
// ??????
const x = (z as Foo).length;
```

????????????

???????? : Foo ????????? as Foo ??? Bug?

```
interface Foo {
  bar: number;
  baz?: string; // ??????????????"bam"?????????"baz"
}

const foo = {
  bar: 123,
  bam: 'abc', // ?????????????????????????????????????
} as Foo;

function func() {
  return {
    bar: 123,
    bam: 'abc', // ?????????????????????????????????????
  } as Foo;
}
```

7.4.20 ????????

???????????????? ; ?????????????

```
// ??????
interface Foo {
  memberA: string;
  memberB: number;
}
```

?? , ??????

```
// ??????
interface Foo {
  memberA: string,
  memberB: number,
}
```

?? , ??????

```
// 
type SomeTypeAlias = {
  memberA: string,
  memberB: number,
};

let someProperty: {memberC: string, memberD: number};
```

Example 7-10

Example 7-10: Using type aliases

```
// 
// 
console.log(x['someField']);
console.log(x.someField);
```

Example 7-11: Using type aliases

```
// 
declare interface ServerInfoJson {
  appVersion: string;
  user: UserJson;
}

const data = JSON.parse(serverResponse) as ServerInfoJson;
console.log(data.appVersion); // 
```

Example 7-12

Example 7-12: Using type aliases

```
// 
import {method1, method2} from 'utils';
class A {
  readonly utils = {method1, method2};
}
```

```
// 
import * as utils from 'utils';
class A {
  readonly utils = utils;
}
```

7.4.20

Web

7.4.21

enum TypeScript const enum JavaScript

7.4.22 debugger

debugger

```
//
function debugMe() {
  debugger;
}
```

7.4.23

@MyDecorator

- Angular @Component @NgModule
- Polymer @property

TC39 Bug

```
/** JSDoc */
@Component({...}) //
class MyComp {
  @Input() myField: string; // .....
  @Input()
  myOtherField: string; // .....
}
```

7.5

7.5.1

TypeScript . . root/path/to/file

./foo path/to/foo

../../../../

```
import {Symbol1} from 'google3/path/from/root';
import {Symbol2} from '../parent/file';
import {Symbol3} from './sibling';
```

Typescript

TypeScript namespace module

TypeScript ES6 import {foo} from 'bar'

namespace Foo { ... }

require import x = require('...'); ES6

```
// 
namespace Rocket {
  function launch() { ... }
}

// <reference> 
/// <reference path="..." />

// require() 
import x = require('mydep');
```

Tip

TypeScript module module Foo { ... } ES6

7.5.2

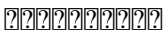
```
// Use named exports:
export class Foo { ... }
```

```
// 
export default class Foo { ... }
```

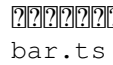
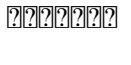
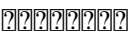
```
// 
import Foo from './bar'; // 
import Bar from './bar'; // 
```

foo.ts

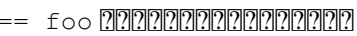
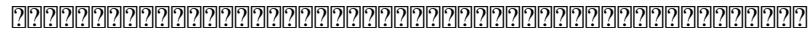
```
// foo.ts
const foo = 'blah';
export default foo;
```

foo bar.ts 

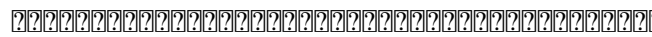
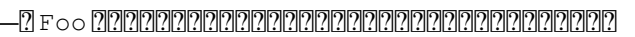
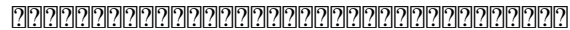
```
// bar.ts
import {fizz} from './foo';
```

 error TS2614: Module './foo' has no exported member 'fizz' 
bar.ts 

```
// foo.ts
import fizz from './foo';
```

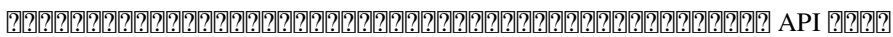

foo fizz === foo 


```
// foo.ts
export default class Foo {
  static SOME_CONSTANT = ...
  static someHelpfulFunction() { ... }
  ...
}
```

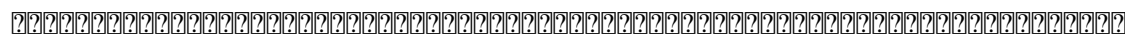

 —  Foo 

```
// foo.ts
export const SOME_CONSTANT = ...
export function someHelpfulFunction()
export class Foo {
  // foo Foo foo
}
```

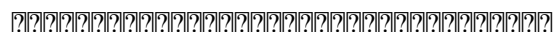
ES6

TypeScript  API 

ES6

 export
let 

```
// foo.ts
export let foo = 3;
// ES6 foo foo foo
// TypeScript foo foo
window.setTimeout(() => {
  foo = 4;
}, 1000 /* ms */);
```



```
// [REDACTED]
let foo = 3;
window.setTimeout(() => {
  foo = 4;
}, 1000 /* ms */);
// [REDACTED]
export function getFoo() { return foo; };
```

[REDACTED]

```
function pickApi() {
  if (useOtherApi()) return OtherApi;
  return RegularApi;
}
export const SomeApi = pickApi();
```

[REDACTED]

[REDACTED]

```
// [REDACTED]
export class Container {
  static FOO = 1;
  static bar() { return 1; }
}
```

[REDACTED]

```
// [REDACTED]
export const FOO = 1;
export function bar() { return 1; }
```

7.5.3 [REDACTED]

[REDACTED] ES6 [REDACTED] TypeScript [REDACTED]

[REDACTED] [REDACTED]	[REDACTED]
[REDACTED] import * as foo from '...';	TypeScript [REDACTED]
[REDACTED] import {Something} from '...';	TypeScript [REDACTED]
[REDACTED] import Something from '...';	[REDACTED]
[REDACTED] import '...';	[REDACTED]

```
// [REDACTED]
import * as ng from '@angular/core';
import {Foo} from './foo';

// [REDACTED]
import Button from 'Button';

// [REDACTED]
import 'jasmine';
import '@polymer/paper-button';
```

Imports

Imports are used to bring in code from other files.

Imports are used to bring in code from other files. **API**

Imports are used to bring in code from other files. **Jasmine** describe it

```
// Imports are used to bring in code from other files.
import {TableViewItem, TableViewHeader, TableViewRow, TableViewModel,
TableViewRenderer} from './tableview';
let item: TableViewItem = ...;
```

```
// Imports are used to bring in code from other files.
import * as tableview from './tableview';
let item: tableview.Item = ...;
```

```
import * as testing from './testing';

// Imports are used to bring in code from other files.
// Imports are used to bring in code from other files.
// Imports are used to bring in code from other files.
testing.describe('foo', () => {
testing.it('bar', () => {
    testing.expect(...);
    testing.expect(...);
});
});
```

```
// Imports are used to bring in code from other files.
import {describe, it, expect} from './testing';

describe('foo', () => {
it('bar', () => {
    expect(...);
    expect(...);
});
});
...
```

Imports

Imports are used to bring in code from other files. **import {Something as SomeOtherThing}**

Imports are used to bring in code from other files.

- Imports are used to bring in code from other files.
- Imports are used to bring in code from other files.
- Imports are used to bring in code from other files. **RxJS** from **observableFrom**

import type export type

import type ... from export type ... from

Tip

export type Foo = ...;

```
// 
import type {Foo} from './foo';
export type {Bar} from './bar';
```

```
// 
import {Foo} from './foo';
export {Bar} from './bar';
```

TypeScript
import type import import type
import '...'
export type API import type
export type API
UserService AjaxUserService

7.5.4

products checkout backend views models controllers

7.6

7.6.1

TypeScript google3 any

```
const x = 15; // x
```

string number boolean RegExp new

```
// boolean 
const x: boolean = true;
```

```
// Set 
const x: Set<string> = new Set();
```

```
// TypeScript
const x = new Set<string>();
```

TypeScript

-
-

7.6.2 Null Undefined

TypeScript null undefined string | null undefined null undefined

TypeScript undefined null JavaScript API undefined Map.get DOM Google API null Element.getAttribute null undefined

| null | undefined

```
// undefined
type CoffeeResponse = Latte|Americano|undefined;

class CoffeeService {
  getLatte(): CoffeeResponse { ... };
}
```

```
// undefined
type CoffeeResponse = Latte|Americano;

class CoffeeService {
  getLatte(): CoffeeResponse|undefined { ... };
}
```

```
//
type CoffeeResponse = Latte|Americano;

class CoffeeService {
  getLatte(): CoffeeResponse {
    return assert(fetchResponse(), 'Coffee maker is broken, file a ticket');
  };
}
```

undefined

TypeScript ?

```
interface CoffeeOrder {
  sugarCubes: number;
  milk?: Whole|LowFat|HalfHalf;
}

function pourCoffee(volume?: Milliliter) { ... }
```

undefined {sugarCubes: 1} CoffeeOrder milk

undefined

```
class MyClass {
  field = '';
}
```

7.6.3

TypeScript

Mock

```
// 
const foo: Foo = {
  a: 123,
  b: 'abc',
}
```

```
// 
const badFoo = {
  a: 123,
  b: 'abc',
}
```

badFoo badFoo

badFoo Foo

```
interface Animal {
  sound: string;
  name: string;
}

function makeSound(animal: Animal) {}

/**
```

(continues on next page)

(continued from previous page)

```

* 'cat' {sound: string}'
*/
const cat = {
  sound: 'meow',
};

/**
 * 'cat'
 * TypeScript
 * 'cat'
 */
makeSound(cat);

/**
 * Horse
 * 'horse' 'Animal'
 */
const horse: Animal = {
  sound: 'niegh',
};

const dog: Animal = {
  sound: 'bark',
  name: 'MrPickles',
};

makeSound(dog);
makeSound(horse);

```

7.6.4

TypeScript

```

//
interface User {
  firstName: string;
  lastName: string;
}

```

```

//
type User = {
  firstName: string,
  lastName: string,
}

```

TypeScript

7.6.5 Array<T>

Array<T> . T[] ReadonlyArray<T>
Array<T>
readonly T[] ReadonlyArray<T>

```
//  
const a: string[];  
const b: readonly string[];  
const c: ns.MyObj[];  
const d: Array<string|number>;  
const e: ReadonlyArray<string|number>;
```

```
//  
const f: Array<string>; //  
const g: ReadonlyArray<string>;  
const h: {n: number, s: string}[]; //  
const i: (string|number)[];  
const j: readonly (string|number)[];
```

7.6.6 {[key: string]: number}

JavaScript

```
const fileSizes: {[fileName: string]: number} = {};  
fileSizes['readme.txt'] = 541;
```

TypeScript

```
//  
const users: {[key: string]: number} = ...;
```

```
//  
const users: {[userName: string]: number} = ...;
```

TypeScript ES6 Map Set JavaScript ES6 Map Set string

TypeScript Record<Keys, ValueType>

7.6.7

TypeScript Record Partial Readonly

TypeScript

- TypeScript
- TypeScript
- TypeScript/

- IDE “” “” “” Pick<T, Keys>

-
-
-

TypeScript Pick<T, Keys> T

```
interface User {
  shoeSize: number;
  favoriteIcecream: string;
  favoriteChocolate: string;
}

// FoodPreferences favoriteIcecream favoriteChocolate shoeSize
type FoodPreferences = Pick<User, 'favoriteIcecream'|'favoriteChocolate'>;
```

FoodPreferences

```
interface FoodPreferences {
  favoriteIcecream: string;
  favoriteChocolate: string;
}
```

User FoodPreferences User FoodPreferences

```
interface FoodPreferences { /* */ }

interface User extends FoodPreferences {
  shoeSize: number;
  // User FoodPreferences
}
```

IDE

7.6.8 any

TypeScript any

any

-
- unknown any
- Lint any

Generics

Generic Interfaces and Functions

```
// Example of a generic interface
declare interface MyUserJson {
    name: string;
    email: string;
}

// Example of a generic type
type MyType = number|string;

// Example of a generic function
function getTwoThings(): {something: number, other: string} {
    // ...
    return {something, other};
}

// Example of a generic function with a constraint
function nicestElement<T>(items: T[]): T {
    // ...
    // T extends HTMLElement
}

```

unknown and any

any is a catch-all type that can be assigned any value. unknown is a type that can only be assigned values that have been explicitly checked for their type.

```
// Example of unknown
const val: unknown = value;
```

```
// Example of any
const danger: any = value /* ... */;
danger.whoops();
```

Linting with any

Linting with any can be disabled using ts-lint rules.

```
// Example of linting with any
const mockBookService = ({get() { return mockBook; }} as any) as BookService;

// Example of linting with any and a rule
const component = new MyComponent(mockBookService, /* unused ShoppingCart */ null as any);
```

7.6.9

Pair

```
// 
interface Pair {
  first: string;
  second: string;
}

function splitInHalf(input: string): Pair {
  // ...
  return {first: x, second: y};
}
```

```
// 
function splitInHalf(input: string): [string, string] {
  // ...
  return [x, y];
}

// :
const [leftHalf, rightHalf] = splitInHalf('my string');
```

```
function splitHostPort(address: string): {host: string, port: number} {
  // ...
}

// :
const address = splitHostPort(userAddress);
use(address.port);

// :
const {host, port} = splitHostPort(userAddress);
```

7.6.10

JavaScript

- String Boolean Number string boolean number
- Object {} object {} null undefined " " symbol bigint

7.6.11

API API

7.7

7.7.1

- 1.

- any
- TypeScript
- .
-
- private

- 2.

JavaScript

-
- x as T
- Array<[number, number]>

- 3.

TypeScript

-
- Closure TS
-
- google3

- 4.

Bug

HTML/CSS [?] - [?]

8.1 0. [?]

[?] [?] Git [?].

[?]

- Google [?] - [?]

[?]

- Google Style Guide

8.2 [?]

[?]HTML/CSS[?]HTML[?]CSS[?]GSS[?]

8.3 [?]

8.3.1 [?]

[?]

[?]URL[?]http:[?]https:[?]URL[?]

```
<!-- [?] -->
<script src="https://www.google.com/js/gweb/analytics/autotrack.js"></script>

<!-- [?] -->
<script src="//www.google.com/js/gweb/analytics/autotrack.js"></script>
```

```
/* [?] */
.example {
  background: url(https://www.google.com/images/example);
}

/* [?] */
.example {
```

(continues on next page)

(continued from previous page)

```
background: url(//www.google.com/images/example);
}
```

8.4

8.4.1

TABTAB

```
<ul>
  <li>Fantastic
  <li>Great
</ul>
```

```
.example {
  color: blue;
}
```

8.4.2

HTMLtext/CDATA CSS

```
<!-- -->
<A HREF="/">Home</A>

<!-- -->

```

```
/* */
color: #E5E5E5;

/* */
color: #e5e5e5;
```

8.4.3

```
<!-- -->
<p>What?_

<!-- -->
<p>Yes please.
```

8.5

8.5.1

UTF-8 BOM

UTF-8

HTML `<meta charset="utf-8">` UTF-8

Handling character encodings in HTML and CSS

8.5.2

HTML/CSS

8.5.3

TODO

TODO @

TODO

TODO

```
{# TODO(john.doe): #}
<center>Test</center>

<!-- TODO: -->
<ul>
  <li>Apples</li>
  <li>Oranges</li>
</ul>
```

8.6 HTML

8.6.1

HTML5

HTML `<!DOCTYPE html>`

HTML `text/html` XHTML `application/xhtml+xml` HTML `
`

8.6.2 HTML

HTML

HTML

W3C HTML validator

HTML

```

<!-- -->
<title>Test</title>
<article>This is only a test.

<!-- -->
<!DOCTYPE html>
<meta charset="utf-8">
<title>Test</title>
<article>This is only a test.</article>

```

8.6.3

HTML

“” p a

HTML

```

<!-- -->
<div onclick="goToRecommendations();">All recommendations</div>

<!-- -->
<a href="recommendations/">All recommendations</a>

```

8.6.4

canvas alt

alt alt alt

CSS alt=""

```

<!-- -->


<!-- -->


```

8.6.5

HTML
 HTML
 HTML
 HTML

```

<!-- -->
<!DOCTYPE html>
<title>HTML sucks</title>
<link rel="stylesheet" href="base.css" media="screen">
<link rel="stylesheet" href="grid.css" media="screen">
<link rel="stylesheet" href="print.css" media="print">
<h1 style="font-size: 1em;">HTML sucks</h1>
<p>I've read about this on a few sites but now I'm sure:
  <u>HTML is stupid!!1</u>
<center>I can't believe there's no way to control the styling of
  my website without doing everything all over again!</center>

<!-- -->
<!DOCTYPE html>
<title>My first CSS-only redesign</title>
<link rel="stylesheet" href="default.css">
<h1>My first CSS-only redesign</h1>
<p>I've read about this on a few sites but today I'm actually
  doing it: separating concerns and avoiding anything in the HTML of
  my website that is presentational.
<p>It's awesome!
    
```

8.6.6

UTF-8
 HTML

```

<!-- -->
The currency symbol for the Euro is &ldquo;&eur;&rdquo;.

<!-- -->
The currency symbol for the Euro is "€".
    
```

8.6.7

HTML5 Web

```
<!-- -->
<!DOCTYPE html>
<html>
  <head>
    <title>Spending money, spending bytes</title>
  </head>
  <body>
    <p>Sic.</p>
  </body>
</html>

<!-- -->
<!DOCTYPE html>
<title>Saving money, saving bytes</title>
<p>Qed.</p>
```

8.6.8 type

type CSS JavaScript type HTML5 text/css text/javascript type

```
<!-- -->
<link rel="stylesheet" href="//www.google.com/css/maia.css" type="text/css">

<!-- -->
<link rel="stylesheet" href="//www.google.com/css/maia.css">

<!-- -->
<script src="//www.google.com/js/gweb/analytics/autotrack.js" type="text/javascript">
</script>

<!-- -->
<script src="//www.google.com/js/gweb/analytics/autotrack.js"></script>
```

8.7 HTML

8.7.1

CSS display li Linter

```
<blockquote>
  <p><em>Space</em>, the final frontier.</p>
</blockquote>

<ul>
  <li>Moe
  <li>Larry
  <li>Curly
</ul>

<table>
  <thead>
    <tr>
      <th scope="col">Income
      <th scope="col">Taxes
    </tr>
  <tbody>
    <tr>
      <td>$ 5.00
      <td>$ 4.50
    </tr>
  </tbody>
</table>
```

8.7.2 HTML

[HTML](#)
[HTML](#)

```
<!-- HTML -->
<a class='maia-button maia-button-secondary'>Sign in</a>

<!-- HTML -->
<a class="maia-button maia-button-secondary">Sign in</a>
```

8.8 css

8.8.1 CSS

[CSS](#)
[CSS](#) [css](#) [bug](#)
[W3C CSS validator](#)
[CSS](#) [CSS](#) [CSS](#)

8.8.2 id class

```

id class
id class
id class
id class
id class

```

```

/*  */
#yee-1901 {}

/*  */
.button-green {}
.clear {}

/*  */
#gallery {}
#login {}
.video {}

/*  */
.aux {}
.alt {}

```

8.8.3 id class

```

ID class
id class
id class

```

```

/*  */
#navigation {}
.atr {}

/*  */
#nav {}
.author {}

```

8.8.4

```

id class
id class
id class

```

```

/*  */
ul#example {}
div.error {}

```

(continues on next page)

(continued from previous page)

```
/* ... */
#example {}
.error {}
```

8.8.5

...

CSS font

...

```
/* ... */
border-top-style: none;
font-family: palatino, georgia, serif;
font-size: 100%;
line-height: 1.6;
padding-bottom: 2em;
padding-left: 1em;
padding-right: 1em;
padding-top: 0;

/* ... */
border-top: 0;
font: 100%/1.6 palatino, georgia, serif;
padding: 0 1em 2em;
```

8.8.6

“0”

...

```
margin: 0;
padding: 0;
```

8.8.7

“0”

...

```
font-size: .8em;
```

8.8.8

3
3

```
/*  */
color: #eebbcc;

/*  */
color: #ebc;
```

8.8.9

idclass

```
.adw-help {} /* AdWords */
#maia-note {} /* Maia */
```

8.8.10 idclass

ID

```
/* : */
.demoimage {}

/*  */
.error_status {}

/*  */
#video-id {}
.ads-sample {}
```

8.8.11 Hacks

CSS“hacks”
CSShacks

8.9 CSS

8.9.1

css

CSS-moz-webkit

```
background: fuchsia;
border: 1px solid;
-moz-border-radius: 4px;
-webkit-border-radius: 4px;
border-radius: 4px;
color: black;
text-align: center;
text-indent: 2em;
```

8.9.2

```
@media screen, projection {
    html {
        background: #fff;
        color: #444;
    }
}
```

8.9.3

```
/* */
.test {
    display: block;
    height: 100px;
}

/* */
.test {
    display: block;
    height: 100px;
}
```

8.9.4 CSS

```

/* */
h3 {
  font-weight:bold;
}

/* */
h3 {
  font-weight: bold;
}

```

8.9.5

```

/* */
#video{
  margin-top: 1em;
}

/* */
#video
{
  margin-top: 1em;
}

/* */
#video {
  margin-top: 1em;
}

```

8.9.6

```

/* */
a:focus, a:active {
  position: relative; top: 1px;
}

/* */
h1,
h2,
h3 {

```

(continues on next page)

(continued from previous page)

```
font-weight: normal;
line-height: 1.2;
}
```

8.9.7 CSS

```
html {
  background: #fff;
}

body {
  margin: auto;
  width: 50%;
}
```

8.9.8 CSS

```
/* */
@import url("//www.google.com/css/maia.css");

html {
  font-family: "open sans", arial, sans-serif;
}

/* */
@import url(//www.google.com/css/maia.css);

html {
  font-family: 'open sans', arial, sans-serif;
}
```

8.10 CSS

8.10.1

```
/* Header */  
#adw-header {}  
/* Footer */  
#adw-footer {}  
/* Gallery */  
.adw-gallery {}
```

8.11

9.2.2 1.2.

Google

9.3 2.

9.3.1 2.1.

.java

9.3.2 2.2. UTF-8

UTF-8

9.3.3 2.3.

2.3.1.

ASCII


- 1.
- 2. Tab

2.3.2

Unicode \b \t \n \f \r \" \' ` \ \ \ 012Unicode \u000a

2.3.3. ASCII

ASCII Unicode ∞ Unicode \u221e Unicode

 Tip

Unicode Unicode

String unitAbbrev = "μs";	
String unitAbbrev = "\u03bcs"; // "μs"	
String unitAbbrev = "\u03bcs"; // Greek letter mu, "s"	
String unitAbbrev = "\u03bcs";	
return '\uffeff' + content; // byte order mark	

Tip

ASCII

9.4 3.

- 1.
- 2.
- 3. Import statements
- 4.

9.4.1 3.1.

9.4.2 3.2.

4.4 Column limit: 100

9.4.3 3.3.

3.3.1.

3.3.2.

4.4 Column limit: 100

3.3.3.

-
-

```

ASCII
import packageA.ClassA; import packageA.
ClassA.; ASCII

```



```

return () -> {
    while (condition()) {
        method();
    }
};

return new MyClass() {
    @Override public void method() {
        if (condition()) {
            try {
                something();
            } catch (ProblemException e) {
                recover();
            }
        } else if (otherCondition()) {
            somethingElse();
        } else {
            lastThing();
        }
        {
            int x = foo();
            frob(x);
        }
    }
};

```

4.8.1,

4.1.3.

K & R 4.1.2 K & R if/else try/catch/finally

```

// 
void doNothing() {}

// 
void doNothingElse() {
}

```

```

// 
try {
    doSomething();
} catch (Exception e) {}

```


9.5.2 4.2. `String`

4.1.2 `String` & `String`

9.5.3 4.3. `String`

9.5.4 4.4. `String`100


Java `String`100 “” `Unicode`4.5


 Tip

`Unicode`

- `Javadoc` `URL` `JSNI`
- 3.2 3.3
- `shell`
- `google-java-format`

9.5.5 4.5. `String`

 Tip

 Tip


4.5.1. [Annotations](#)

Annotations are used to provide metadata about the code.

- 1. [@Google](#) [@C++](#) [@JavaScript](#)
 - `@Foo`
 * `@Bar`
 * `@Baz`
 * `@T extends Foo & Bar`
 * `catch (FooException | BarException e)`
- 2. [for](#) [foreach](#)
 - `for` `foreach`
- 3. [@](#) (`@`
- 4. `@`, `@`
- 5. `lambda` `lambda`

```
MyLambda<String, Long, Object> lambda =
    (String label, Long value, Object obj) -> {
        ...
    };

Predicate<String> predicate = str ->
    longExpressionInvolving(str);
```

 **Tip**
 Annotations are used to provide metadata about the code.

4.5.2. [Annotations](#)

Annotations are used to provide metadata about the code.

Annotations are used to provide metadata about the code.

Annotations are used to provide metadata about the code.

9.5.6 4.6. [Annotations](#)

4.6.1. [Annotations](#)

Annotations are used to provide metadata about the code.

- 1. [Annotations](#)
 - `Annotations`
 - `Annotations` [4.8.1](#) `Annotations`
- 2. [Annotations](#) [3.3](#) `Annotations` `Annotations`

.....
.....

4.6.2. Annotations

Javadoc ASCII

- 1. `if` for `catch` ()
- 2. `else` `catch` }
- 3. {
 - `@SomeAnnotation({a, b})`
 - `String[][] x = {"foo"};` { }
- 4. " "
 - `<T extends Foo & Bar>`
 - `catch` `catch` (`IOException` | `BarException` `e`)
 - `for` "foreach" :
 - `lambda` (`String` `str`) -> `str.length()`
- 5. , : ;)
- 6. //
- 7. //
- 8. `List<String>` `list`
- 9.
 - `new int[] {5, 6}` and `new int[] { 5, 6 }`
- 10. [] ...

.....

4.6.3. Annotations

.....

.....Google.....

.....

```
private int x; //
private Color color; //

private int x; //
private Color color; //
```

Tip

conflicts

9.5.7 4.7.

Java

9.5.8 4.8.

4.8.1.

```
private enum Answer {
    YES {
        @Override public String toString() {
            return "yes";
        }
    },
    NO,
    MAYBE
}
```

4.8.3.1

```
private enum Suit { CLUBS, HEARTS, SPADES, DIAMONDS }
```

4.8.2.

4.8.2.1.

int a, b;

for

4.8.2.2.

4.8.3. ??

4.8.3.1. ?????"?"?

?????"?"

```

new int[] {
    0, 1, 2, 3
}

new int[] {
    0, 1,
    2, 3
}

new int[] {
    0,
    1,
    2,
    3
}

new int[]
{0, 1, 2, 3}

```

4.8.3.2. ????C?????

String[] args String args[]

4.8.4. switch??

switch case FOO: default:

4.8.4.1. ??

switch

4.8.4.2. ?????

switch break continue return // fall through switch

```

switch (input) {
    case 1:
    case 2:
        prepareOneOrTwo();
        // fall through
    case 3:

```

(continues on next page)

(continued from previous page)

```

        handleOneTwoOrThree ();
        break;
    default:
        handleLargeNumber (input);
    }

```

switch case 1

4.8.4.3. default

switch default

enum switch default

4.8.5.

4.8.5.1.

@Target (ElementType.TYPE_USE)

```

final @Nullable String name;

public @Nullable Person getPersonByName (String name)

```

4.8.5.2.

4.5

```

@Deprecated
@CheckReturnValue
public final class Frozzler { ... }

```

4.8.5.3.

```

@Deprecated
@Override
public String getNameIfPresent () { ... }

```

4.8.5.4. `@Mock`

Mockito

```
@Partial @Mock DataLoader loader;
```

4.8.5.5. `@Mocked`

Mockito

4.8.6. `/**`

JUnit5 Javadoc

JUnit5

4.8.6.1. `/**`

JUnit5 `/** ... */` `/** ... */` `/** ... */`

```
/*
 * This is           // And so           /* Or you can
 * okay.             // is this.          * even do this. */
 */
```

JUnit5

Tip
JUnit5 `/** ... */` `/** ... */`

4.8.7. `enum`

Java

```
public protected private abstract default static final transient volatile_
↳synchronized native strictfp
```

4.8.8. `long`

```
long L 1 3000000000L
3000000000L
```

9.6 5. `Google`

9.6.1 5.1. `Google`

```
ASCII \w+
Google name_ mName s_name kName
```

9.6.2 5.2. `com.example.deepspace`

5.2.1. `com.example.deepspace`

```
com.example.deepspace com.example.deepspace com.example.deep_space
```

5.2.2. `UpperCamelCase`

```
UpperCamelCase
Character ImmutableList List
Readable
Test HashIntegrationTest Test HashImplTest
```

5.2.3. `lowerCamelCase`

```
lowerCamelCase
sendMessage stop
JUnit transfer-
Money_deductsFromSource
```

5.2.4. `UPPER_SNAKE_CASE`

```
UPPER_SNAKE_CASE
static final
null
```


9.7.2 6.2. try-with-resources

try-with-resources Assertion-Error

try catch

```
try {
    int i = Integer.parseInt(response);
    return handleNumericResponse(i);
} catch (NumberFormatException ok) {
    // it's not numeric; that's fine, just continue
}
return handleTextResponse(response);
```

try expected

```
try {
    emptyStack.pop();
    fail();
} catch (NoSuchElementException expected) {
}
```

9.7.3 6.3. try-with-resources

try-with-resources

```
Foo aFoo = ...;
Foo.aStaticMethod(); //
aFoo.aStaticMethod(); //
somethingThatYieldsAFoo().aStaticMethod(); //
```

9.7.4 6.4. try-with-resources

Object.finalize

Tip

Effective Java

9.8 7. Javadoc

9.8.1 7.1. Javadoc

7.1.1. Javadoc

Javadoc

```
/**
 * Javadoc,
 * ...
 */
public int method(String p1) { ... }
```

```
/** Javadoc */
```

Javadoc @return

7.1.2.

HTML `` `<table>` `<p>`

7.1.3.

@param @return @throws @deprecated

9.8.2 7.2.

Javadoc

A {@code Foo} is a... This method returns... Save the record.

Tip

Javadoc `/** @return the customer ID */` `/** Returns the customer ID. */`

9.8.3 7.3. Javadoc

public public protected Javadoc

7.3.4 Javadoc Javadoc

7.3.1. `getFoo()`

```
getFoo() /** Returns the foo */
```

Tip

```
getCanonicalName  
canonical name /** Returns the canonical  
name. */
```

7.3.2. `getFoo()`

```
/**
```

7.3.4. `getFoo()` Javadoc `7.3.3`

```
/**
```

```
/**
```

```
7.1.1 7.1.2 7.1.3 7.2
```