

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

Google ????????

Mar 14, 2024

CONTENTS

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

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

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

Contents

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

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

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

2024/02/18

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

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 2020 2020

2020 2020

2020 2020 2020 2020. 2020 2020 2020 2020. 2020 2020, 2020 2020, 2020 2020 2020 2020 2020 2020. 2020 2020 2020 2020 2020 2020 (2020 2020), 2020 2020 2020 2020 2020 2020.

2020 2020 2020 2020:

2020 2020

2020 2020 2020 2020, 2020 2020 2020. 2020 2020 2020 2020 2020 2020, 2020 2020 2020, 2020 2020 2020 2020 2020. 2020, goto 2020 2020 2020, 2020 2020 2020 2020.

2020, 2020 2020

2020 2020 (2020 2020) 2020 2020. 2020, 2020 2020 2020 2020 2020 2020. 2020 2020 2020. “2020 2020” 2020 2020 2020 (2020 2020), 2020 2020 2020 2020 2020 (2020 2020 std::unique_ptr 2020 2020 2020).

2020 2020

2020 2020 2020 2020, 2020 2020 (2020) 2020. 2020 2020 2020: 2020 2020 2020 #include 2020, 2020 2020 2020 2020 2020. 2020, 2020 “2020” 2020 2020 “2020 2020”; 2020 2020, 2020 2020 2020 2020 2020. 2020, 2020 2020 2020. 2020 2020 2020 2020 2020 2020. 2020 2020 2020 (2020, 2020 2020 2020). 2020 2020 2020 2020, 2020 2020. 2020 2020 2020 2020 2020 2020.

2020 C++ 2020

2020 2020 2020 2020, 2020 2020 2020 2020. 2020 C++ 2020 2020 2020, 2020 2020 2020, 2020 2020 2020 2020. 2020, 2020 2020 2020 2020 2020 2020. 2020 (2020 2020) 2020 2020 2020 2020. 2020, 2020 C++ 2020, 2020 2020 2020 2020. 2020 2020 2020 2020 2020 2020.

2020 2020 2020 2020

2020 C++ 2020 2020 2020. 2020 2020 2020 2020 2020 2020. 2020 2020 2020 2020 2020.

2020 2020 2020 C++ 2020 2020 2020 2020 2020

2020 C++ 2020 2020 2020, 2020 2020 2020. 2020 2020 2020, 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020. 2020, 2020 2020 2020 2020. 2020 2020 2020 2020, 2020 2020 2020 2020: 2020 2020 2020 2020 2020 2020.

2020 2020

2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020. 2020, 2020 2020 2020 (global namespace): 2020 2020 2020 2020 2020 2020 (name collision), 2020 2020.

2020 2020

2020 2020 2020 2020 2020 2020, 2020 2020 2020 2020.

2020 2020 2020 2020 2020 2020. 2020, 2020 2020 2020. 2020 2020 2020 C++ 2020 2020, 2020 2020 2020 2020. 2020 2020 2020 2020 2020 2020: 2020 “2020 2020”. 2020 2020. 2020, 2020, 2020 2020 2020.

2.1.4 C++

C++20, C++23. C++ (C) .
.
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 ., , (pre-requisite). .inc ., .

2.2.2 1.2. #define

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

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

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

2.2.3 1.3. `foo.h`

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

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

2.2.4 1.4. `foo.h`

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

`foo.h`

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

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

`foo.h`

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

`foo.h`

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

```
// 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 "b.h"`, `test()` `f(void*)`.

- `#include "b.h"`.
- `foo` (`foo`) `foo`.

???

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

2.2.5 1.5. ????

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

??:

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

??:

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

??:

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

??:

???????????????????????? 10 ????. ??????????. ?????????????????????, ?????????????????????????????????! ??????????????: ????????????? switch ????????????????? (???????????? switch ?????????????). ?????, ?????????????????????, ?????????????????; ??, ?????????????????????????????. ????????????????????????????? (YuleFox ?): ?????????????????????????????, ?????????????????????????????, ????????????????????????????????? (class) ?????????, ??????????????????????????. ?????????????????????????.

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

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

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

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

? dir/foo.cc ? dir/foo_test.cc ????????????? dir2/foo2.h ?????????, ?????????????????:

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


```
// .h
namespace mynamespace {

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

} // namespace mynamespace
```

```
// .cc
namespace mynamespace {

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

} // namespace mynamespace
```

.cc, (flag) using .

```
#include "a.h"

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

namespace mynamespace {

using ::foo::Bar;

... // .....

} // namespace mynamespace
```

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

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

- , (namespace alias). API:

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

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

(continues on next page)

(continued from previous page)

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

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

- [REDACTED].
- [REDACTED] “internal”, [REDACTED] API.

```
// Absl [REDACTED].
using ::absl::container_internal::ImplementationDetail;
```

- [REDACTED], [REDACTED].

[REDACTED]: [REDACTED]

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

2.3.2 2.2. [REDACTED]

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

[REDACTED]:

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

[REDACTED]:

[REDACTED] .cc [REDACTED]. [REDACTED] .h [REDACTED].

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

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

2.3.3 2.3. `constexpr`

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

Tip:

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

Tip:

`constexpr`, `constexpr`.

Tip:

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

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

2.3.4 2.4. `constexpr`

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

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

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

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

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

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

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

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

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

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

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

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

Example:

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

2.3.5 2.5. Example

Tip: `static storage duration` `trivially destructible`, `destructor` `base`. `virtual destructor`, `dynamic initialization`.

Example: `constexpr`.

Example:

`object` `lifetime` `storage duration`. `static specifier`. `dynamic` `non-trivial`. `named constants` `translation unit` `flag`.

`dynamic` `non-trivial`. `named constants` `translation unit` `flag`.

Example:

`named constants` `translation unit` `flag`.

Example:

`constexpr` `constexpr`. `constexpr` `constexpr`.

Example:

`constexpr` `constexpr`. `constexpr` `constexpr`.

```
const int kNum = 10; // Example
struct X { int n; };
```

(continues on next page)

Example:

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

Example: `explicit` keyword

Example

- Example: `explicit` keyword
- Example: `string_view` and `std::string`
- Example: `explicit` keyword

Example

- Example: `explicit` keyword

Example

Example: `explicit` keyword

Example: `explicit` keyword

Example: `explicit` keyword

2.4.3 3.3. Example

Example

Example: `explicit` keyword

Example

Example: `explicit` keyword

Example: `std::unique_ptr` and `std::string`

Example: `std::string` and `std::string`

Example: `std::string` and `std::string`

Example

Example: `API` and `API`

protected, private, public, friend, etc.

protected, private, public, friend, etc.

2.4.4 3.4. struct VS. class

??

struct, class.

??

C++ struct class

struct, class, struct, class

class, class.

STL, traits, class, struct.

???

3.5. pair VS. tuple

??

struct, pair, tuple.

??

pair, tuple, .first, .second, std::get<X>, C++14 std::get<Type>, tuple

pair, tuple, API

2.4.5 3.6. ??

??

(YuleFox, GoF <<Design Patterns>>)

??

“” ()

??

API, API, ?

??

?, ?

(,)

??

public, final

“” (“is-a”, YuleFox: “has-a”)

template<typename T>, operator== 重载 T 的 == 操作符。T t1, t2 类型, operator<=>, operator== 重载。重载 == 操作符。

重载 ==, !=, <, >, <=, >=, << 重载 Equals(), CopyFrom() 重载 PrintTo(). 重载, 重载 operator<=>, operator==, operator<, operator>, std::set 重载, 重载 <.

重载 &&, ||, , 重载 &. 重载 operator", 重载, 重载 (重载).

重载 == 操作符. = 重载 重载. 重载 << 重载. 重载 重载, 重载.

2.4.7 3.8. 面试题

面试题

面试题 private, 面试题 (面试题). 面试题, 面试题 const 面试题.

面试题, 面试题 Google Test 面试题 .cc 面试题 protected. 面试题 .cc 面试题, 面试题 .h 面试题, 面试题.

2.4.8 3.9. 面试题

面试题

面试题, 面试题 public 面试题.

面试题

面试题 public: 面试题, 面试题 protected:, 面试题 private:. 面试题.

面试题, 面试题, 面试题:

- 1. 面试题 (面试题 typedef, using, enum, 面试题)
- 2. (面试题, 面试题 struct) 面试题
- 3. 面试题
- 4. 面试题
- 5. 面试题
- 6. 面试题
- 7. 面试题 (面试题, 面试题)
- 8. 面试题 (面试题)

面试题. 面试题, 面试题, 面试题. 面试题 面试题.

2.4.9 (YuleFox)

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

2.5 4.

2.5.1 4.1.

C++ `std::optional` `std::optional`. `std::optional` `std::optional` `std::optional`.

C/C++ `std::optional`, `std::optional`, `std::optional`. `std::optional` `const` `std::optional`, `std::optional` `std::optional`. `std::optional` `std::optional` `std::optional` `std::optional` `std::optional` `std::optional`.

`std::optional` `const` `std::optional` `std::optional` `std::optional` `std::optional` `std::optional` `std::optional`.

`std::optional` `std::optional`. `std::optional`, `std::optional`, `std::optional`, `std::optional`.

`std::optional`. `std::optional` (`std::optional`) `std::optional`. `std::optional`, `std::optional`, `std::optional`.

2.5.2 4.2. [C++](#)

??

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

??

????????????????, ??????????????. ?????? 40 ?, ?????????????????????????????????????.

????????????????????, ?????????, ?????????, ?????????? bug. ??????, ?????????????.

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

2.5.3 4.3. [C++](#)

??

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

??

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

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

??

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

??

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

??

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

2.5.4 4.4. [C++](#)

??

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

??

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

??

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

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

... ..

... ..

??

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

... ..

2.5.5 4.5. ???? ???? ??

??

... ..

??

C++

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

C++11 auto

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

... .. int

??

... .. Lambda

... ..

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

... ..

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

??

... .. C ? Java

... ..

??

... .. (Lambda)

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

2.6.2 5.2. Cpplint

>
cpplint.py
>
cpplint.py
cpplint.py

2.6.3 acgyrant

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

2.7 6. C++

2.7.1 6.1.

Tip:

void f(string&& s);

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

C++11


```

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

```

2.7.4 6.4. alloca()

Tip: alloca().

:

alloca(). alloca() .

:

alloca() C++ . , bugs: " , "

:

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

2.7.5 6.5.

Tip: .

, . FooBuilder Foo , FooBuilder Foo , . , .

() . , public, , .

2.7.6 6.6.

Tip: C++ .

:

- failures acgyrant error code, int
- C++ & Python, Java C++
- C++
- acgyrant factory function, C++ Init() , " " "
-

:

- throw f () f () g (), g () h (), h f g, g
-
- RAII. . . . , , , “ ” . (). , , .
-
-

:

Google C++ , .

Google , .

Google , .

Windows , .

(YuleFox : , , C++ , Google , , ,)

2.7.7 6.7.

TODO

Tip: RTTI.

:

RTTI C++ . typeid dynamic_cast .

:

RTTI () . , , .

RTTI . , . RTTI .

RTTI . :

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

:

RTTI. RTTI switch. RTTI.

RTTI:

RTTI RTTI. RTTI RTTI. RTTI, RTTI RTTI. RTTI RTTI:

RTTI.

RTTI, RTTI, RTTI. RTTI.

RTTI dynamic_cast. RTTI, RTTI dynamic_cast RTTI.

RTTI, RTTI. RTTI:

```

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

RTTI, RTTI. RTTI, RTTI.

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

2.7.8 6.8. RTTI

Tip: C++ RTTI, static_cast<>(). RTTI int y = (int)x RTTI int y = int(x) RTTI;

RTTI:

C++ RTTI C RTTI, RTTI.

RTTI:

C RTTI; RTTI (int)3.5, RTTI (int)"hello". RTTI, C++ RTTI.

RTTI:

RTTI.

RTTI:

RTTI C RTTI. RTTI C++ RTTI.

- static_cast RTTI C RTTI, RTTI.
- const_cast RTTI const RTTI.
- reinterpret_cast RTTI. RTTI.

RTTI dynamic_cast RTTI 6.7. RTTI.

2.7.10 6.10. `++i` and `--i`

Tip: `++i` and `--i` are `++i` and `--i`.

Tip:

`++i` and `--i` are `++i` and `--i`.

Tip:

`++i` and `--i` are `++i` and `--i`.

Tip:

`++i` and `--i` are `++i` and `--i`.

Tip:

`++i` and `--i` are `++i` and `--i`.

2.7.11 6.11. `const`

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

Tip:

`const` is `const`.

Tip:

`const` is `const`.

Tip:

`const` is `const`.

Tip:

- `const` is `const`.
- `const` is `const`.
- `const` is `const`.

`const` is `const`.

`mutable` is `mutable`.

`const` is `const`.

```

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

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

```

2.7.12 6.12. constexpr

Tip: C++11 constexpr

:

```

constexpr constexpr,
constexpr

```

:

```

constexpr

```

:

```

constexprconstexprconstexpr

```

:

```

constexpr C++ constexpr
constexprconstexprconstexpr
constexpr

```

2.7.13 6.13.

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

:

```

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

```

:

```

.

```

:

```

C++

```

:

```

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

```



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

- sizeof(void *) != sizeof(int). intptr_t.
- int64_t/uint64_t (Yang.Y : -). 64, 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

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

```
auto,
```


std::initializer_list<T>

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

std::initializer_list<T>

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

auto

Warning:

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

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

9.7.

2.7.20 6.20. Lambda

Tip: lambda lambda

Lambda

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

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

2.7.22 6.22. Boost

Tip: Boost

:

Boost, C++

:

Boost, C++

:

Boost

:

Boost

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

Boost

C++ 11

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

2.7.23 6.23. C++11

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

C++11

C++11 C++ Boost C++

C++11 1300 vs 800

6.22. Boost C++11

C++11 C++11

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

2.7.24 acgyrant

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

2.8 7. `using`

`using` 声明。 `using` 声明： `using`, `using`, `using`, `using`, `using`, `using`, `using`。

`using`, `using`, `using`, `using`, `using`。

2.8.1 7.1. `using`

`using`

`using`, `using`, `using`; `using`。

`using`

`using`, `using`, `using`。 `using`, `using`。

```
int price_count_reader; // 价格计数
int num_errors; // "num" 错误数
int num_dns_connections; // 连接 "DNS" 数
```

```
int n; // 数量。
int nerr; // 错误数。
int n_comp_conns; // 连接数。
int wgc_connections; // 连接数。
int pc_reader; // "pc" 价格计数。
int cstmr_id; // 客户 ID。
```

`using`, `using`, `using` i `using` T `using`。

`using`: `using` `using`, `using` `using`。

2.8.2 7.2. `using`

`using`

`using`, `using` (`_`) `using` (`-`), `using`。 `using`, `using` “`_`” `using`。

`using`

`using`:

- `my_useful_class.cc`
- `my-useful-class.cc`
- `myusefulclass.cc`
- `myusefulclass_test.cc // _unittest` `_regtest` `using`。

C++ `using` `.cc` `using`, `using` `.h` `using`。 `using` `.inc` `using`, `using` `using`。

`using` `/usr/include` `using` (Yang.Y `using`), `using` `db.h`。

`using` `http_server_logs.h` `using` `logs.h` `using`。 `using`, `using` `foo_bar.h` `using` `foo_bar.cc`, `using` `FooBar`。

`using` `.h` `using`。 `using`, `using` `.h` `using`。

2.8.3 7.3. `std::unordered_map`

Example:

`MyExcitingClass`, `MyExcitingEnum`.

Example:

Example: `MyExcitingClass` (typedef), `MyExcitingEnum` (enum), `MyExcitingMap` (unordered_map), `MyExcitingProperties` (struct).

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

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

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

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

2.8.4 7.4. `std::unordered_set`

Example:

Example: `MyExcitingClass`, `MyExcitingEnum`. Example: `MyExcitingSet`, `MyExcitingProperties`, `MyExcitingMap`: `a_local_variable`, `a_struct_data_member`, `a_class_data_member`.

Example:

Example:

Example:

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

string tableName; // Example - Example
```

Example:

Example: `MyExcitingClass`, `MyExcitingEnum`, `MyExcitingSet`.

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

2.8.5

2.8.5.1, 2.8.5.2, 2.8.5.3:

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

2.8.5.1, 2.8.5.2 vs. 2.8.5.3.

2.8.5 7.5. 2.8.5.1

2.8.5.1

2.8.5.1 constexpr const int, 2.8.5.1.1, 2.8.5.1.2 “k”, 2.8.5.1.3:

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

2.8.5.1

2.8.5.1.1 (2.8.5.1.1.1, 2.8.5.1.1.2) 2.8.5.1.1.3. 2.8.5.1.1.4, 2.8.5.1.1.5, 2.8.5.1.1.6. 2.8.5.1.1.7, 2.8.5.1.1.8.

2.8.6 7.6. 2.8.6.1

2.8.6.1

2.8.6.1.1, 2.8.6.1.2: MyExcitingFunction(), MyExcitingMethod(), my_exciting_member_variable(), set_my_exciting_member_variable().

2.8.6.1

2.8.6.1.3, 2.8.6.1.4 (2.8.6.1.4.1 “2.8.6.1.4.1”), 2.8.6.1.5. 2.8.6.1.6, 2.8.6.1.7 (2.8.6.1.7.1 StartRpc() 2.8.6.1.7.2 StartRPC()).

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

(2.8.6.1.8, 2.8.6.1.9 API 2.8.6.1.10, 2.8.6.1.11, 2.8.6.1.12.)

2.8.6.1.13. 2.8.6.1.14, 2.8.6.1.15. 2.8.6.1.16 int count() void set_count(int count).

2.9.3 8.3. Iterator

Iterator. (C++, Apache 2.0, BSD, LGPL, GPL)
Iterator.

Iterator

Iterator.h Iterator, Iterator, Iterator. Iterator, Iterator, Iterator.

Iterator.h.cc Iterator, Iterator.

2.9.3 8.3. Iterator

Iterator

Iterator, Iterator, Iterator.

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

Iterator

Iterator, Iterator, Iterator. Iterator, Iterator.
Iterator, Iterator.

Iterator, Iterator.

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

2.9.4 8.4. Iterator

Iterator

Iterator; Iterator.

Iterator

File

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

File:

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

File:

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

File, File. File “File false”, File:

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

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

File

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

File.h File. File, File.

???

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

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

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

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

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

???????

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

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

??????????:

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

?

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

????????.

Return type:

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

Parameters:

- Parameter 1.
- Parameter 2, Parameter 3.
- Parameter 4, 5.
- Parameter 6, 7.
- Parameter 8.
- Parameter 9.
- Parameter 10.
- Parameter 11.
- Parameter 12, 13.
- Parameter 14, 15.
- Parameter 16.
- Parameter 17.
- Parameter 18 2 19.
- Parameter 20 4 21.

Class definition:

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

Class hierarchy:

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

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

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

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

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

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

2.10.5 9.5. Lambda

??

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

??

???????, ?????? & ???????.

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

? lambda ?????????????.

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

2.10.6 9.6.

??

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

??

????????????????

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

??????????, ?????, ?????????????????????, ?????????????????????????

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

????????????, ??????

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

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

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

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

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

// 3x3 widget.

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

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

2.10.7 9.7. `return`

`return`

`return` statement, `return` statement.

`return`

`return` statement, `return` statement, `return` statement, `return` statement.

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

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

2.10.8 9.8. if-else

if-else

if-else statement. if-else statement.

if-else

if-else statement. if-else statement.

if-else statement. if-else statement. if-else statement. if-else statement. if-else statement. if-else statement.

```
if (condition) { // do something.
    ... // 2 lines.
} else if (...) { // else if statement.
    ...
} else {
    ...
}
```

if-else statement:

```
if ( condition ) { // do something - do
    ... // 2 lines.
} else { // else if statement.
    ...
}
```

if-else if-else statement. if-else if-else statement:

```
if(condition) // do - IF statement.
if (condition){ // do - { statement.
if(condition){ // do statement.
```

```
if (condition) { // do - IF statement { statement.
```

if-else statement, if-else statement. if-else statement else statement:

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

if-else statement:

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

if-else statement, if-else statement; if-else statement. if-else statement if-else statement:

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

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

if-else if-else statement, if-else if-else statement:

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

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

```
// IF, ELSE
if (condition) {
    foo;
} else {
    bar;
}
```

2.10.9 9.9. SWITCH

??

switch (var), cases { } continue.

??

switch case { }.

case, switch default (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; // - contunue
```

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

2.10.12 9.12.

```
return expr;
x = expr; return expr;
```

```
return result; // ,
// (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`

`if`

`if (lopsided_score), DropEverything();`

`if`

`if (lopsided_score), DropEverything();`

```
// 1 - 1
if (lopsided_score) {
#ifdef DISASTER_PENDING // 1 - 1
    DropEverything();
#endif
    NotifyClient();
}
#endif
BackToNormal();
}
```

```
// 1 - 1
if (lopsided_score) {
#ifdef DISASTER_PENDING // 1 - "#if" 1
    DropEverything();
#endif // 1 - "#endif" 1
    BackToNormal();
}
```

2.10.15 9.15. `class`

`class`

`class MyClass : public OtherClass {`

`class`

`class (MyClass, 1) MyClass:`

```
class MyClass : public OtherClass {
public: // 1
    MyClass(); // 1
    explicit MyClass(int var);
    ~MyClass() {}

    void SomeFunction();
    void SomeFunctionThatDoesNothing() {
    }

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

private:
    bool SomeInternalFunction();

    int some_var_;
    int some_other_var_;
};
```

~~~~:

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

## 2.10.16 9.16. ~~~~~

??

~~~~~.

??

~~~~~:

```

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

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

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

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

## 2.10.17 9.17.

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

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

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

## 2.10.18 9.18.

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

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

(continues on next page)

(continued from previous page)

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

. (Yang.Y : IDE)

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

```
//
x = 0;

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

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





- `#pragma __declspec`. `__declspec(dllimport)`  
`__declspec(dllexport)`, `DLLIMPORT` `DLEXPOR`,  
`_____`.

Windows \_\_\_\_\_:

- `COM` `ATL/WTL`. `COM` `ATL/WTL`,  
`_____`.
- `ATL` `STL` Visual C++ `STL`. `ATL`, `_____`  
`_ATL_NO_EXCEPTIONS`. `STL`, `_____`, `_____`. (`_____`  
`STL`, `_____`).
- `StdAfx.h` `precompile.h`. `_____`,  
`_____` (`precompile.cc`), `/FI`.
- `resource.h`, `_____`.

## 2.12 11. ???

\_\_\_\_\_, \_\_\_\_.

\_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_. `if`, \_\_\_\_\_. (\*) \_\_\_\_\_,  
 \_\_\_\_\_.

\_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_,  
 \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_.

\_\_\_\_\_, \_\_\_\_\_: \_\_\_\_\_.



## 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

Objective-C C Mac OS X iPhone

Cocoa Mac OS X Objective-C Mac OS X

Objective-C Google C++ Objective-C Google

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

Note: Google C++ Objective-C++

Mac OS X Google

Google Google Toolbox for Mac project GTM

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

### 3.1.3

@interface

```

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

#import <Foundation/Foundation.h>

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

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

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

```

(continues on next page)

(continued from previous page)

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

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

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

@end
```

Implementation getters setters init dealloc

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

#import "Foo.h"

@implementation Foo

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

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

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

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

- (NSString *)bar {
    return bar_;
}

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

(continues on next page)

(continued from previous page)

```

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

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

@end

```

@interface @implementation @end @interface }

### 3.2

#### 3.2.1 vs.

---

Tip:

---

#### 3.2.2

80

Objective-C 80

80

Xcode > Preferences > Text Editing > Show page guide

#### 3.2.3

---

Tip:

- / +
- 

```

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

```

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

4

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

### 3.2.4

Tip:

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

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

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

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

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

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

### 3.2.5 @public @private

Tip: @public @private

C++ public, private protected

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

### 3.2.6

Tip: @catch

Objective-C

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

### 3.2.7

Tip:

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

### 3.2.8 `block`

Tip: `block` target/selector `block` 4

`block`

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

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

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

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

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

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

(continues on next page)

```

    }
    });

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

```

### 3.3

Objective-C Objective-C naming rules C++ Google C++ Objective-C Objective-C URL TIFF EXIF Objective-C++ C++ API Objective-C Cocoa C++ Cocoa Cocoa @implementation Objective-C C++ C++

#### 3.3.1

---

Tip: -

---

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

GTMNSString+Utils.h ~GTMNSTextView+Autocomplete.h`

#### 3.3.2 Objective-C++

---

Tip: Objective-C++

---

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

```

// file: cross_platform_header.h

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

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

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

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

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

```

### 3.3.3

Tip:

GTMSendMessage

### 3.3.4

Tip:

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

### 3.3.5 Objective-C

Tip:

convertPoint:fromRect: 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

usernameTextField\_ Objective-C  
 2.0 KVO/KVC KVO=Key Value Observing KVC=Key Value Observing  
 Coding Objective-C  
 2.0 @property @synthesize

### Objective-C

k kInvalidHandle kWritePerm

## 3.4

Objective-C

Objective-C

C++

### 3.4.1

Tip:

### Objective-C

Objective-C

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

Objective-C

### 3.4.2

Tip:

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

“”

### 3.4.3

Tip: |

count

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

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

### 3.4.4

Tip: Objective-C

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

CoreFoundation C++ Objective-C retained \_\_strong \_\_weak CoreFoundation Objective-C \_\_weak clang C++

Objective-C C++

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

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

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

- retained - retained

## 3.5 Cocoa Objective-C

### 3.5.1 @private

Tip: @private

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

### 3.5.2

Tip:

### 3.5.3

Tip: init... bug

bug

### 3.5.4 NSObject

Tip: NSObject @implementation

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

### 3.5.5

Tip: `init` returns `0` if `nil`

`0` is a `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`

Objective-C 2.0

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

`@implementation`

“middle truncation” Bug

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

### 3.5.8 #import and #include

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

```

#import #include

```

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

```

Objective-C #define #import Objective-C Objective-C
#import

```

```

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

```

```

Mac C C++ #define Mac
#import #include #include

```

```

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

```

### 3.5.9

Tip: #import

```

Cocoa Foundation
#import #include Objective-C

```

```

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

```

### 3.5.10 autorelease

Tip: autorelease release

```

release return

```

```

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

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

```

### 3.5.11 autorelease retain

Tip: autorelease `` `` retain

“” “autorelease retain” autorelease

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

### 3.5.12 init dealloc

Tip: init dealloc

init dealloc ivals

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

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

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

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

### 3.5.13

Tip: dealloc @interface

```
dealloc retained
dealloc retained @interface dealloc
```

### 3.5.14 setter NSStrings

Tip: NSString setter copy

```
retain NSString
NSMutableString
```

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

### 3.5.15

Tip: @throw Objective-C OS

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

```
NS_DURING NS_HANDLER NS_ENDHANDLER NS_VALUEReturn NS_VOIDRETURN
Mac OS X 10.2
```

Objective-C Objective-C++

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

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

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

(continues on next page)

(continued from previous page)

```

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

try/catch

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 YES NO 256 256 512 ...

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

BOOL &| ! BOOL

isBold isValid

```

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

isBold

```

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

(continues on next page)

(continued from previous page)

```

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

```

YES/NO BOOL

```

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

```

```

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

```

### 3.5.18 Property

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

@synthesize

```

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

@implementation MyClass
@synthesize name = name_;
@end

```

??

@implementation @interface  
 ?? @implementation

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

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

copy @Attribute?

copy @attribute NSString @property  
 NSString ? setter copy retain

???

property synthesize ? setter ? getter get ? set  
 property nonatomic

???

Objective-C 2.0 set/get

???

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

???

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

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

### 3.5.19

Tip:

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

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

### 3.5.20

Tip: iOS

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

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

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

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

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

## 3.6 Cocoa

### 3.6.1

Tip: retain

1. delegate\_
2. delegate setDelegate:

3. delegate\_ retain

### 3.6.2 MVC

---

Tip: API @protocol

---

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

## 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 开发者资源](#) - 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





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

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

doctor/who/ jodie.py

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

sys.path, import jodie

### 4.3.4

Tip:

:

:

:

:

1. ValueError (API). assert, assert, assert, raise.

:

```
def connect_to_next_port(self, minimum: int) -> int:
    """
    :param minimum: 1024
    :return:
    :raises: ConnectionError
    """
    if minimum < 1024:
        # ValueError API
        #
        raise ValueError(f'1024 {minimum}.')
    port = self._find_next_open_port(minimum)
```

(continues on next page)



??:

```

    _MAX_HOLY_HANDGRENADE_COUNT = 3
    SIR_LANCELOTS_FAVORITE_COLOR = "blue"

```

### 4.3.6

Tip:

??:

```

    (nonlocal)

```

??:

```


```

??:

```


```

??:

```

    self
    _

```

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

Tip:

??:

```

    map(), filter(), lambda

```

??:

```


```

??:

```


```

??:

```

    for

```

??:

```

result = [mapping_expr for value in iterable if filter_expr]

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

result = [complicated_transform(x)

```

(continues on next page)

(continued from previous page)

```

        for x in iterable if predicate(x)]

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

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

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

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

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

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

```

??:

```

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

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

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

```

### 4.3.8

Tip: ,

?:

?:

?:

?:

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

Python:

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

Python:

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

### 4.3.9

Tip: Python.

Python:

Python yield Python. Python, Python, Python.

Python:

Python, Python. Python, Python.

Python:

Python, Python.

Python:

Python. Python"Yields:"Python>Returns:".

(Python: Python)

Python, Python.

Python PEP-0533.

### 4.3.10 Lambda

Tip: Python. Python map()/filter() lambda Python.

Python:

lambda Python, Python.

Python:

Python.

Python:

Python. Python. Python lambda Python, Python.

Python:

Python. Python60-80Python, Python.

Python, Python operator PythonlambdaPython. Python, Python operator.mul Python lambda x, y: x \* y.

### 4.3.11 Ternary Operator

Tip: Ternary Operator.

Example 1: `x = 1 if cond else 2.`

Example 2: `if cond, x.`

Example 3: `if cond: x else: y.`

Example 4: `cond and x or y.`

```

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

```

bad_line_breaking = ('yes' if predicate(value) else
                    'no') # 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 Default Argument Values

Tip: Default Argument Values.

Example 1: `def foo(a, b = 0):`

Example 2: `Python`

Example 3: `Python`

Example 4: `Python`

Example 5: `Python`

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

:

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

def foo(a, b=[]):
    ...
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“(attribute)”, “(property)”).

Tip: . (attribute) : .

:

.

:

1. , (getter) (setter).
2. .
3. .
4. , .

:

1. , (operator overload).
2. .

:

. , , , . , .

, : , (). .

@property (decorator) .

. (override) .



### 4.3.15 Lexical Scoping

Tip: [Python](#).

Tip:

Python's lexical scoping is implemented by the CPython interpreter. It is implemented in Python, and it is implemented in Python. It is implemented in Python.

Python:

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

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

Tip:

Python, Lisp, Scheme, Haskell, ML, etc.

Tip:

Python bug, PEP-0227:

```
i = 4
def foo(x: Iterable[int]):
    def bar():
        print(i, end='')
        # ...
        # Python
        # ...
    for i in x: # i, i Foo, bar
        print(i, end='')
    bar()
```

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

(Python: `x` for `x`, `i` for `i`, `foo` for `foo`, `bar` for `bar`. C++: `C++`.)

Tip:

Python.

### 4.3.16

Tip: `staticmethod`, `classmethod`.

Tip:

`@property`, `@staticmethod`, `@classmethod`, `my_decorator`.



Tip: `from __future__ import annotations` is available in Python 3.7 and later. It allows you to use type annotations in Python 2.7 and 3.5-3.6.2.

Tip: `from __future__ import annotations` is available in Python 3.7 and later. It allows you to use type annotations in Python 2.7 and 3.5-3.6.2.

### 4.3.19 `python: from __future__ imports`

Tip: `from __future__ import annotations` is available in Python 3.7 and later. It allows you to use type annotations in Python 2.7 and 3.5-3.6.2.

Tip: `from __future__ import annotations` is available in Python 3.7 and later. It allows you to use type annotations in Python 2.7 and 3.5-3.6.2.

Tip: `from __future__ import annotations` is available in Python 3.7 and later. It allows you to use type annotations in Python 2.7 and 3.5-3.6.2.

Tip: `from __future__ import annotations` is available in Python 3.7 and later. It allows you to use type annotations in Python 2.7 and 3.5-3.6.2.

Tip: `from __future__ import annotations` is available in Python 3.7 and later. It allows you to use type annotations in Python 2.7 and 3.5-3.6.2.

Tip: `from __future__ import annotations` is available in Python 3.7 and later. It allows you to use type annotations in Python 2.7 and 3.5-3.6.2.

```
from __future__ import generator_stop
```

Tip: `from __future__ import annotations` is available in Python 3.7 and later. It allows you to use type annotations in Python 2.7 and 3.5-3.6.2.

### 4.3.20 `python: type annotations`

Tip: `from __future__ import annotations` is available in Python 3.7 and later. It allows you to use type annotations in Python 2.7 and 3.5-3.6.2.

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

Tip: `from __future__ import annotations` is available in Python 3.7 and later. It allows you to use type annotations in Python 2.7 and 3.5-3.6.2.

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

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

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

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

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

## 4.4 Python?????

### 4.4.1 ??

---

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

---

### 4.4.2 ??

---

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

---

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

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

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

??:

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

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

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

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

(continues on next page)

(continued from previous page)

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

??:

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

bridge_questions.clarification_on \
    .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 ████████.

??:

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

??:

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

80 80, Black Pyink , 80 .

### 4.4.3

Tip: .

(tuple) , .

??:

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

??:

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

### 4.4.4

Tip: .

, ( , ) 4 ( , ) .

??:

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

meal = (spam,
        beans)

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

# 4█████████████; ████████
foo = long_function_name(
    var_one, var_two, var_three,
    var_four)
meal = (
    spam,
    beans)

# 4█████████████; ████████
# ██████████.
foo = long_function_name(
    var_one, var_two, var_three,
    var_four
)
meal = (
    spam,
    beans,
)

# ██████████████████.
foo = {
    'long_dictionary_key':
        long_dictionary_value,
    ...
}
```

██:

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

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

# ██████████████.
foo = {
    'long_dictionary_key':
        long_dictionary_value,
    ...
}
```

#### 4.4.5

Tip: `[, ), }` ... Python ...

#### 4.4.6 Shebang

Tip: `.py` ... `#!/usr/bin/env python3` (virtualenv) `#!/usr/bin/python3`.

(`Shebang` (Hashbang) `#!/usr/bin/env python3`, `#!/usr/bin/python3`, `#!/bin/sh` ...)

Python ... Python ...

#### 4.4.7 (docstring)

Tip: ...

Python

Python ... `__doc__` ... `pydoc` ... (PEP-257) ...

Tip

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

...

```

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

Python

...

...

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

.....

```
"""foo.bar """
```

### .....

..... (generator) ..... (property).

.....:

1. API
2. ....
3. ....

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

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

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

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

#### Args: (.....):

..... ....., ..... ....., ..... 80 .., ..... (.....) ....., ..... \*foo (.....) ..... \*\*bar (.....) .., ..... \*foo ..... \*\*bar .....

#### Returns: (".....")

..... "Yields:" (".....")

..... None, ..... Returns (.....) ..... Yields (.....) ..... Bigtable .., ..... Numpy ..... (.....). ....., .....: ".....: ..... (mat\_a, mat\_b), ..... mat\_a .., .....". .....

#### Raises: (.....):

..... Args (.....) .....+.....+...../....., ..... API ..... (..... API ..... API ....., .....

```
def fetch_smalltable_rows(
    table_handle: smalltable.Table,
```

(continues on next page)

(continued from previous page)

```

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

    table_handle Table object. UTF-8.

    :
    table_handle: smalltable.Table.
    keys: UTF-8.
    require_all_keys: True,

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

    keys, (require_all_keys false).

    :
    IOError: smalltable.
    """

```

Args ( ):

```

def fetch_smalltable_rows(
    table_handle: smalltable.Table,
    keys: Sequence[bytes | str],
    require_all_keys: bool = False,
) -> Mapping[bytes, tuple[str, ...]]:
    """Smalltable object.

    table_handle Table object. UTF-8.

    :
    table_handle:
        smalltable.Table.
    keys:
        UTF-8.
    require_all_keys:
        True,

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

    keys, (require_all_keys false).

```

(continues on next page)

(continued from previous page)

```

class IOError:
    """IOError: errno filename reason"""

```

**class**

Python classes are objects. They have attributes, methods, and a namespace. They are created using the `class` statement. The `class` statement is used to create a new class. The class is a blueprint for creating objects. The class has attributes and methods. The class is created in the current namespace. The class is a first-class object. The class can be assigned to a variable. The class can be passed as an argument to a function. The class can be returned from a function. The class can be used as a return value. The class can be used as a key in a dictionary. The class can be used as a value in a list. The class can be used as a value in a tuple. The class can be used as a value in a set. The class can be used as a value in a dictionary. The class can be used as a value in a list. The class can be used as a value in a tuple. The class can be used as a value in a set. The class can be used as a value in a dictionary.

```

class SampleClass(object):
    """Sample class"""

    # Attributes
    # ...

    def __init__(self, likes_spam = False):
        """Initialize SampleClass"""
        self.likes_spam = likes_spam
        self.eggs = 0

    def public_method(self):
        """Public method"""

```

Python classes are objects. They have attributes, methods, and a namespace. They are created using the `class` statement. The `class` statement is used to create a new class. The class is a blueprint for creating objects. The class has attributes and methods. The class is created in the current namespace. The class is a first-class object. The class can be assigned to a variable. The class can be passed as an argument to a function. The class can be returned from a function. The class can be used as a return value. The class can be used as a key in a dictionary. The class can be used as a value in a list. The class can be used as a value in a tuple. The class can be used as a value in a set. The class can be used as a value in a dictionary. The class can be used as a value in a list. The class can be used as a value in a tuple. The class can be used as a value in a set. The class can be used as a value in a dictionary.

`class`:

```

class CheeseShopAddress:
    """CheeseShopAddress"""

    ...

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

```

`class`:

```

class CheeseShopAddress:
    """CheeseShopAddress"""

    ...

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

```

**Python**

Python classes are objects. They have attributes, methods, and a namespace. They are created using the `class` statement. The `class` statement is used to create a new class. The class is a blueprint for creating objects. The class has attributes and methods. The class is created in the current namespace. The class is a first-class object. The class can be assigned to a variable. The class can be passed as an argument to a function. The class can be returned from a function. The class can be used as a return value. The class can be used as a key in a dictionary. The class can be used as a value in a list. The class can be used as a value in a tuple. The class can be used as a value in a set. The class can be used as a value in a dictionary. The class can be used as a value in a list. The class can be used as a value in a tuple. The class can be used as a value in a set. The class can be used as a value in a dictionary.





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

```
import os
from absl import logging

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

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

Example:

```
import os
from absl import logging

logging.info('PAGER: %s', os.getenv('PAGER', default=''))
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'Invalid HOME, $HOME={homedir!r}')
```

Example:

Example (Example: ValueError (Invalid HOME, \$HOME=...)):

1. Invalid HOME, \$HOME=...
2. Invalid HOME, \$HOME=...
3. Invalid HOME, \$HOME=... (Invalid, Invalid grepping).

Example:

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

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

Example:

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

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

try:
```

(continues on next page)



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

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

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

`?? TODO(???)`. `TODO(???)`. `??, ????` `TODO(??, ????)`.

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

`TODO(???)`, `TODO(???)` (“200911??”) `TODO(???)` (“?? XML ??, ???”), `TODO(???)`.

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

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

??:

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

??:

```
import os, sys
```

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

1. `Python ? __future__`. ??:

```
from __future__ import annotations
```

`???` `__future__`.

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

```
import sys
```

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

```
import tensorflow as tf
```

4. `???`. ??:

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

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

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

Python

(from path import ... path)

```
import collections
import queue
import sys

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

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

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

### 4.4.13

Tip:

try / except, try except

:

```
if 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`)

`getter`, `setter`, `property`, `getter`.

`getter` (`attribute`), `setter`. `getter`, `setter`, `property`.

(`getter`: `getter`, `setter`: `setter`! `getter`, `setter`. `Pythonic`)

`getter`, `getter` `getter()` `setter()`.

`getter`, `getter`/`setter`. `getter`, `setter`.

#### 4.4.15 `getter`

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`.

`getter`, `getter`. `getter`, `getter`.

`getter.py` `getter`.

`getter`

1. `getter`, `getter`:
  1. `getter` (`i`, `j`, `k`, `v`).
  2. `try/except` `e`.
  3. `with` `f`.
  4. `constrain` (`type variable`, `_T = TypeVar("_T"), _P = ParamSpec("_P")`).
2. `getter(-)` `getter`.
3. `getter`, `__double_leading_and_trailing_underscore__` (`Python`).
4. `getter`.
5. `getter` (`id_to_name_dict`).

`getter`

1. “`(Internal)`” `getter`.
2. `getter`, `getter` (`getter`).
3. `getter` (`dunder`) `getter` (`name mangling`).
4. `getter`. `Java`, `getter`.

5. `lower_with_under` (CapWords), `lower_with_under.py`. `lower_with_under.py` `lower_with_under`, `lower_with_under`, `lower_with_under` (“`lower_with_under`, `lower_with_under` import StringIO `lower_with_under` from StringIO import StringIO?”).
6. `lower_with_under` PEP 8, `lower_with_under`, `lower_with_under` test\_<lower\_with\_under>\_<lower\_with\_under>. `lower_with_under` CapWords `lower_with_under`, `lower_with_under`, `lower_with_under` test `lower_with_under`, `lower_with_under`. `lower_with_under` test\_<lower\_with\_under>\_<lower\_with\_under>.

lower\_with\_under

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

Python Guido

Table 1: lower\_with\_under

| lower_with_under | lower_with_under | lower_with_under                    |
|------------------|------------------|-------------------------------------|
| lower_with_under | lower_with_under | lower_with_under                    |
| lower_with_under | lower_with_under | lower_with_under (lower_with_under) |
| lower_with_under | lower_with_under | lower_with_under (lower_with_under) |
| lower_with_under | lower_with_under | lower_with_under                    |
| lower_with_under | lower_with_under | lower_with_under                    |

Table 2: lower\_with\_under

| lower_with_under | lower_with_under   | lower_with_under    |
|------------------|--------------------|---------------------|
| lower_with_under | lower_with_under   | lower_with_under    |
| lower_with_under | lower_with_under   | _lower_with_under   |
| lower_with_under | CapWords           | _CapWords           |
| lower_with_under | CapWords           | lower_with_under    |
| lower_with_under | lower_with_under() | _lower_with_under() |
| lower_with_under | CAPS_WITH_UNDER    | _CAPS_WITH_UNDER    |
| lower_with_under | lower_with_under   | _lower_with_under   |
| lower_with_under | lower_with_under   | _lower_with_under   |
| lower_with_under | lower_with_under() | _lower_with_under() |
| lower_with_under | lower_with_under   | lower_with_under    |
| lower_with_under | lower_with_under   | lower_with_under    |

lower\_with\_under

lower\_with\_under, lower\_with\_under, lower\_with\_under. lower\_with\_under, lower\_with\_under, lower\_with\_under. lower\_with\_under API lower\_with\_under PEP8 lower\_with\_under, lower\_with\_under API lower\_with\_under.



5. Annotations

1. Python API.
2. Annotations, Annotations, Annotations.
3. Annotations (Annotations).
4. Annotations.
5. Annotations. Annotations.

??

Annotations.

Annotations, Annotations (signature) Annotations. Annotations, Annotations.

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

Annotations, Annotations. ?, Annotations, Annotations.

```
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]:
    ...
```

Annotations.

???:

```
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:
    ...
```

PEP-008, PEP-0484 (alias PEP-484). PEP-484.

PEP:

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

PEP:

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

PEP (forward declaration)

PEP-0484 (PEP-484), PEP-484 from \_\_future\_\_ import annotations

PEP:

```
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:
    ...
```

PEP

PEP-008, PEP-0484, PEP-484 = PEP-484.

PEP:

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

PEP:

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

### NoneType

Python `NoneType` “`None`” type. `None`, `NoneType` `None`. `None`, `Optional` `Union`.

`X | None` `PEP 484` `a: str = None` `a: str | None = None`.

Example:

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

### Typing Aliases

`typing.TypeAlias` (Python 3.10) `CapWording`, `__Private`.

`typing.TypeAlias` Python 3.10

```
from typing import TypeAlias

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

### Pytype

`# type: ignore`

`pytype` `(pytype)`:

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

### Decorators

`SomeUndecoratedFunction`

`SomeUndecoratedFunction`, `SomeUndecoratedFunction`: `SomeUndecoratedFunction`, `SomeUndecoratedFunction`.

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

Example:

`(Python 3.6)`, `# type: <code>`

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

Annotations

Annotations are used to describe the types of variables, function arguments, and return values.

(Annotations: 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)
```

Annotations (type variable)

Python Annotations (generics). Annotations: TypeVar, ParamSpec.

Annotations:

```
from collections.abc import Callable
from typing import ParamSpec, TypeVar
_P = ParamSpec("_P")
_T = TypeVar("_T")
...
def next(l: list[_T]) -> _T:
    return l.pop()

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

TypeVar Annotations.

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

AnyStr typing Annotations. Annotations: bytes, str.

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

(Annotations: x, bytes, str.)

Annotations:

- Annotations
- Annotations

Annotations:

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

??:

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

?????

???????????? typing.Text. ????????????? Python 2/3 ??????

? str ?????/?????. ? bytes ?????????.

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

????????????????????, ?????????????????????, ????? AnyStr.

?????

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

```
from collections.abc import Mapping, Sequence
from typing import Any, Generic
```

????????, ?????????????????????, ????? typing ? collections.abc ????????????????????? (keyword) ??????. ?????????????????????????????, ??????????????????. ?????????????????????????????, ??? import x as y ??????:

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

????, ??????????. ?? Python 3.9 ??? PEP-585, ?????????????????????????????.

```
def generate_foo_scores(foo: set[str]) -> list[float]:
    ...
```

?: Apache Beam ????????????????? typing ?????????????????.

```
from typing import Set, List

# ????????? Apache Beam ????? PEP 585 ?????, ????
# ????? Python 3.9 ?????????, ??????????.
def generate_foo_scores(foo: Set[str]) -> List[float]:
    ...
```

????????

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

???????????????????????? if TYPE\_CHECKING: ????

1. ?????, ?????????????????????????????, ????? Python 3.6 ??????????. ?? Python 3.6 ?????????????????.
2. ?????????????????????????????????????, ??????. ?????????????, ?????????????????.

3. `typing.TypeVar`.
4. `typing.Type`.
5. `typing.Type`, `typing.TypeVar`.

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

**Annotations**

`typing.Any`, `typing.Optional`. `typing.TypeVar`. `typing.Type`, `typing.TypeVar` (build system) `typing.Type`.

`typing.Any` `typing.TypeVar`. `typing.TypeVar`, `typing.TypeVar` (Any `typing.TypeVar` Any). `typing.TypeVar`.

```
from typing import Any

some_mod = Any # typing.TypeVar some_mod.py typing.TypeVar.
...

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

**Generics**

`typing.TypeVar`, `typing.TypeVar`. `typing.TypeVar`, `typing.TypeVar` Any .

`typing.TypeVar`:

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

`typing.TypeVar`:

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

`typing.TypeVar` Any, `typing.TypeVar`, `typing.TypeVar` `typing.TypeVar`.

`typing.TypeVar`:

```
def get_names(employee_ids: Sequence[Any]) -> Mapping[Any, str]:
    """typing.TypeVar IDtyping.TypeVar."""
```

`typing.TypeVar`:

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

## 4.5

## SHELL 5.1 - 5.2

### Contents

- Shell 5.1 - 5.2

## 5.1 5.1

5.1

1.26

5.1

Paul Armstrong

5.1

5.1

Bean Zhang v1.26

5.1

- Google Style Guide
- Google 5.1 - 5.2

## 5.2 5.2

### 5.2.1 5.2.1 Shell

---

Tip: Bash 5.1 shell 5.1

---

```
5.1 #!/bin/bash 5.1 set 5.1 shell 5.1 bash <script_name>
5.1
```

```
5.1 shell 5.1 bash 5.1 shell 5.1
```

```
5.1 Solaris SVR4 5.1 Bourne shell 5.1
```

## 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: STDERR

```
err() {
    echo "[$(date +%Y-%m-%dT%H:%M:%S%z)]: $@" >&2
}

if ! do_something; then
    err "Unable to do_something"
    exit "${E_DID_NOTHING}"
fi
```

## 5.5

### 5.5.1

---

Tip:

---

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

### 5.5.2

---

Tip:

---

- 
- 
- 
- 

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

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

#####
# Cleanup files from the backup dir
# Globals:
#   BACKUP_DIR
```

(continues on next page)

(continued from previous page)

```
# ORACLE_SID
# Arguments:
# None
# Returns:
# None
#####
cleanup() {
    ...
}
```

### 5.5.3

Tip:

### 5.5.4 TODO

Tip: TODO

C++

TODOs TODO bug ticket

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

## 5.6

### 5.6.1

Tip:

## 5.6.2

Tip: `cat <<END`

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: `command1 | command2`

Long commands

`command1 \`

```
# 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`

Shell

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

(continues on next page)

```

else
  mkdir -p "${dir}/${ORACLE_SID}"
  if [[ "$?" -ne 0 ]]; then
    error_message
  fi
fi
done

```

### 5.6.5 case

Tip:

- `case` is a shorthand for `if` statements.
- `case` is useful for handling multiple options or arguments.
- `case` is often used in shell scripts for menu-driven interfaces.

`case` is a shorthand for `if` statements. It is useful for handling multiple options or arguments. It is often used in shell scripts for menu-driven interfaces.

```

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

```

Example of `case` statement usage in a script.

```

verbose='false'
aflag=''
bflag=''
files=''
while getopts 'abf:v' flag; do
  case "${flag}" in
    a) aflag='true' ;;
    b) bflag='true' ;;
    f) files="${OPTARG}" ;;
    v) verbose='true' ;;
    *) error "Unexpected option ${flag}" ;;
  esac
done

```

## 5.6.6

Tip: `{var} $var`

- 1.
- 2.
3. `shell`

```
# Section of recommended cases.

# Preferred style for 'special' variables:
echo "Positional: $1" "$5" "$3"
echo "Specials: !=$, -=$, _=$_. ?=$?, #=$# *=$* @=$@ \=$$ ..."

# Braces necessary:
echo "many parameters: ${10}"

# Braces avoiding confusion:
# Output is "a0b0c0"
set -- a b c
echo "${1}0${2}0${3}0"

# Preferred style for other variables:
echo "PATH=${PATH}, PWD=${PWD}, mine=${some_var}"
while read f; do
  echo "file=${f}"
done < <(ls -l /tmp)

# Section of discouraged cases

# Unquoted vars, unbraced vars, brace-quoted single letter
# shell specials.
echo a=$avar "b=$bvar" "PID=${$}" "${1}"

# Confusing use: this is expanded as "${1}0${2}0${3}0",
# not "${10}${20}${30}"
set -- a b c
echo "$10$20$30"
```

## 5.6.7

Tip:

- `shell`
- 
- 
- `[ [`

- `set -- $e` `set -- $*`

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



### 5.7.3

Tip:

Bash

```
# Do this:
if [[ "${my_var}" = "some_string" ]]; then
    do_something
fi

# -z (string length is zero) and -n (string length is not zero) are
# preferred over testing for an empty string
if [[ -z "${my_var}" ]]; then
    do_something
fi

# This is OK (ensure quotes on the empty side), but not preferred:
if [[ "${my_var}" = "" ]]; then
    do_something
fi

# Not this:
if [[ "${my_var}X" = "some_stringX" ]]; then
    do_something
fi
```

`-z`-n``

```
# Use this
if [[ -n "${my_var}" ]]; then
    do_something
fi

# Instead of this as errors can occur if ${my_var} expands to a test
# flag
if [[ "${my_var}" ]]; then
    do_something
fi
```

### 5.7.4

Tip:

`- /* *`

```
# Here's the contents of the directory:
# -f -r somedir somefile

# This deletes almost everything in the directory by force
psa@bilby$ rm -v *
removed directory: `somedir`
```

(continues on next page)

(continued from previous page)

```
removed `somefile'

# As opposed to:
psa@bilby$ rm -v ./.*
removed `./-f'
removed `./-r'
rm: cannot remove `./somedir': Is a directory
removed `./somefile'
```

## 5.7.5 Eval

Tip: `eval`

`Eval`

```
# What does this set?
# Did it succeed? In part or whole?
eval $(set_my_variables)

# What happens if one of the returned values has a space in it?
variable="$(eval some_function)"
```

## 5.7.6 `while`

Tip: `for` `while` `while` `shell` `shell`

`while` `shell` `bug`

```
last_line=NULL
your_command | while read line; do
    last_line="${line}"
done

# This will output 'NULL'
echo "${last_line}"
```

`for`

```
total=0
# Only do this if there are no spaces in return values.
for value in $(command); do
    total+="${value}"
done
```

`shell` `bash` `while` `shell`

```
total=0
last_file=
while read count filename; do
```

(continues on next page)

(continued from previous page)

```
total+="${count}"
last_file="${filename}"
done <<(your_command | uniq -c)

# This will output the second field of the last line of output from
# the command.
echo "Total = ${total}"
echo "Last one = ${last_file}"
```

shellwhile"awkshell

```
# Trivial implementation of awk expression:
# awk '$3 == "nfs" { print $2 " maps to " $1 }' /proc/mounts
cat /proc/mounts | while read src dest type opts rest; do
  if [[ ${type} == "nfs" ]]; then
    echo "NFS ${dest} maps to ${src}"
  fi
done
```

## 5.8

### 5.8.1

---

Tip: :: function

---

:: Google

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

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

() function

### 5.8.2

---

Tip:

---

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

### 5.8.3

Tip:

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

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

getopts getopts  
declare readonly export

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

### 5.8.4

Tip:

Google maketemplate make\_template make-template

### 5.8.5

Tip: readonly declare -r

shell

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

### 5.8.6 ██████████

---

Tip: `local` ██████████

---

```
local ██████████
█████████ local ██████████
```

```
my_func2() {
  local name="$1"

  # Separate lines for declaration and assignment:
  local my_var
  my_var="$(my_func)" || return

  # DO NOT do this: $? contains the exit code of 'local', not my_func
  local my_var="$(my_func)"
  [[ $? -eq 0 ]] || return

  ...
}
```

### 5.8.7 ██████████

---

Tip: ██████████

---

```
█████████includes set ██████████
```

### 5.8.8 ██████████main

---

Tip: ██████████ main ██████████

---

```
█████████main ██████████
main ██████████
```

```
main "$@"
```

```
█████████main ██████████
```

## 5.9

### 5.9.1

Tip:

```

$? if

```

```

if ! mv "${file_list}" "${dest_dir}/" ; then
    echo "Unable to move ${file_list} to ${dest_dir}" >&2
    exit "${E_BAD_MOVE}"
fi

# Or
mv "${file_list}" "${dest_dir}/"
if [[ "$?" -ne 0 ]]; then
    echo "Unable to move ${file_list} to ${dest_dir}" >&2
    exit "${E_BAD_MOVE}"
fi

```

Bash PIPESTATUS

```

tar -cf - ./* | ( cd "${dir}" && tar -xf - )
if [[ "${PIPESTATUS[0]}" -ne 0 || "${PIPESTATUS[1]}" -ne 0 ]]; then
    echo "Unable to tar files to ${dir}" >&2
fi

```

PIPESTATUS PIPESTATUS PIPESTATUS

```

tar -cf - ./* | ( cd "${DIR}" && tar -xf - )
return_codes=(${PIPESTATUS[*]})
if [[ "${return_codes[0]}" -ne 0 ]]; then
    do_something
fi
if [[ "${return_codes[1]}" -ne 0 ]]; then
    do_something_else
fi

```

### 5.9.2

Tip: shell

```

bash (1) sed

```

```

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

```

(continues on next page)

(continued from previous page)

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

## 5.10 ██

████████████████

██████████████C++████████████████████

## JAVASCRIPT [?] - [?]

### 6.1 [?]

Google JavaScript JavaScript

### 6.2 Javascript [?]

#### 6.2.1 var [?]

var

[?]

var var var docu  
var

#### 6.2.2 [?]

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

[?]

[?]

CONSTANT\_VALUE\_CASE

number string boolean

## Constants

```

@const
const
const
@const

```

### Example

```

@const CONSTANT_VALUE_CASE

```

```

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

```

```

60 @const

```

```

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

```

## 6.2.3

### Example

```

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

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

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

// 2. IE Firefox.
//.
[normalVersion, ffVersion][isIE]();

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

```

(continues on next page)

(continued from previous page)

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

### 6.2.3

1. js42
2. “no sush property in undefined” x[normalVersion, ffVersion][isIE] ()
3. die resultOfOperation() NaN THINGS\_TO\_EAT die ()

### 6.2.4

```
js”)”}”]”
```

### 6.2.5

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

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

### 6.2.4

### 6.2.5

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

ECMAScript ECMA-262 13 14 EcmaScript ECM

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

### 6.2.6

### 6.2.7

### 6.2.8

string.charAt(3) string[3]

### 6.2.9

```
var x = new Boolean(false);
if (x) {
  alert('hi'); //“hi”
}
```

```
var x = Boolean(0);
if (x) {
  alert('hi'); //
}
typeof Boolean(0) == 'boolean';
typeof new Boolean(0) == 'object';
```

### 6.2.10

class B class D goog.inherits()

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

(continues on next page)

(continued from previous page)

```
D.prototype.method =function() {
    ...
};
```

### 6.2.11

```
/** */ function SomeConstructor() { this.someProperty = 1; } Foo.prototype.
someMethod = function() { ... };
```

“new”

```
Foo.prototype.bar = function() {
    /* ... */
};
```

```
/** @constructor */
function Foo() {
    this.bar = value;
}
```

JavaScript

### 6.2.12

this.foo = null

```
o.prototype.dispose = function() {
    this.property_ = null;
};
```

```
Foo.prototype.dispose = function() {
    delete his.property_;
};
```

JavaScript if (key in obj)

### 6.2.13

DOM

JavaScript DOM

DOM

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

DOM

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

function bar(a, b) {
  return function() { /* a b */ };
}
```

### 6.2.14 eval()

RPC

eval() eval() eval()

eval()

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

eval()

eval() RPC XMLHttpRequest RPC JavaScript

```
var userOnline = false;
var user = 'nusrat';
var xmlhttp = new XMLHttpRequest();
xmlhttp.open('GET', 'http://chat.google.com/isUserOnline?user=' + user, false);
xmlhttp.send('');
// 
// userOnline = true;
if (xmlhttp.status == 200) {
  eval(xmlhttp.responseText);
}
```

(continues on next page)

(continued from previous page)

```
}
// userOnline true
```

### 6.2.15 with() {}

with()

with() with() with()

```
with (foo) {
  var x = 3;
  return x;
}
```

x foo setter 3 with()

### 6.2.16 this

this

this eval() DOM HTML call() apply()

this

- 
- 

### 6.2.17 for-in

for-in

for-in 0 length-1

```
function printArray(arr) {
  for (var key in arr) {
    print(arr[key]);
  }
}

printArray([0,1,2,3]); //

var a = new Array(10);
printArray(a); //

a = document.getElementsByTagName('*');
printArray(a); //

a = [0,1,2,3];
a.buhu = 'wine';
printArray(a); //

a = new Array;
```

(continues on next page)

(continued from previous page)

```
a[3] = 3;
printArray(a); //
```

```
function printArray(arr) {
  var l = arr.length;
  for (var i = 0; i < l; i++) {
    print(arr[i]);
  }
}
```

### 6.2.18

.....

JS Date RegExp String

### 6.2.19

```
var myString = 'A rather long string of English text, an error message \
  actually that just keeps going and going -- an error \
  message to make the Energizer bunny blush (right through \
  those Schwarzenegger shades)! Where was I? Oh yes, \
  you\'ve got an error and all the extraneous whitespace is \
  just gravy. Have a nice day.';
```

ECMAScript

```
var myString = 'A rather long string of English text, an error message ' +
  'actually that just keeps going and going -- an error ' +
  'message to make the Energizer bunny blush (right through ' +
  'those Schwarzenegger shades)! Where was I? Oh yes, ' +
  'you\'ve got an error and all the extraneous whitespace is ' +
  'just gravy. Have a nice day.';
```

## 6.2.20

```
// 3
var a1 = new Array(x1, x2, x3);

// 2
var a2 = new Array(x1, x2);

// If x1 is a number and it is a natural number the length will be x1.
// If x1 is a number but not a natural number this will throw an exception.
// Otherwise the array will have one element with x1 as its value.
var a3 = new Array(x1);

// 0
var a4 = new Array();
```

```
var a = [x1, x2, x3];
var a2 = [x1, x2];
var a3 = [x1];
var a4 = [];
```

```
var o = new Object();

var o2 = new Object();
o2.a = 0;
o2.b = 1;
o2.c = 2;
o2['strange key'] = 3;
```

```
var o = {};

var o2 = {
  a: 0,
  b: 1,
  c: 2,
  'strange key': 3
};
```

### 6.2.21

Object.prototype Array.prototype Function.prototype

### 6.2.22 Internet Explorer

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

JavaScript

## 6.3 Javascript

### 6.3.1

functionNamesLikeThis variableNamesLikeThis ClassNamesLikeThis  
EnumNamesLikeThis methodNamesLikeThis CONSTANT\_VALUES\_LIKE\_THIS foo.  
namespaceNamesLikeThis.bar filenameslikethis.js

- 
- 

opt\_

var\_args var\_args arguments

@param

### getter/setter

EcmaScript 5 getter/setter getter

```
/**
 *--.
 */
var foo = { get next() { return this.nextId++; } };
};
```

### 

getter/setter getFoo() setFoo(value) isFoo()

### 

JavaScript

JavaScript

### 

“Project Sloth” sloth.\*

```
var sloth = {};

sloth.sleep = function() {
  ...
};
```

JavaScript the Closure Library Dojo toolkit

```
goog.provide('sloth');

sloth.sleep = function() {
  ...
};
```

### 

sloths hats Sloth sloth.hats

foo.hats.\*

foo.hats.\*

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

API

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

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

goog.scope

goog.scope

```
/**
 * @constructor
 */
some.long.namespace.MyClass = function() {
};

/**
 * @param {some.long.namespace.MyClass} a
 */
some.long.namespace.MyClass.staticHelper = function(a) {
  ...
};

myapp.main = function() {
  var MyClass = some.long.namespace.MyClass;
  var staticHelper = some.long.namespace.MyClass.staticHelper;
  staticHelper(new MyClass());
};
```

goog.scope

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

(continues on next page)



### 6.3.4

window window window

### 6.3.5

C++

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

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

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

//
this.rows_ = [
  "Slartibartfast" <fjordmaster@magrathea.com>',
  "Zaphod Beeblebrox" <theprez@universe.gov>',
  "Ford Prefect" <ford@theguide.com>',
  "Arthur Dent" <has.no.tea@gmail.com>',
  "Marvin the Paranoid Android" <marv@googlemail.com>',
  'the.mice@magrathea.com'
];

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







### ternary operator

ternary operator, ternary operator

```

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;

```

### chaining

```

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

```

## 6.3.6

### delete

delete operator

delete operator delete operator void return throw case in new

## 6.3.7

string literals

string literals "HTML"

```

var msg = 'This is some HTML';

```

## 6.3.8

@private @protected JSDoc

JSDoc @private @protected

--jscomp\_warning=visibility

@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 */
AA_PublicClass = function() {
    /** @private */
    this.privateProp_ = 2;

    /** @protected */
    this.protectedProp = 4;
};

/** @private */
AA_PublicClass.staticPrivateProp_ = 1;

/** @protected */
AA_PublicClass.staticProtectedProp = 31;

/** @private */
AA_PublicClass.prototype.privateMethod_ = function() {};

/** @protected */
AA_PublicClass.prototype.protectedMethod = function() {};

// File 2.

/**
 * @return {number} The number of ducks we've arranged in a row.
 */
AA_PublicClass.prototype.method = function() {
    // Legal accesses of these two properties.
    return this.privateProp_ + AA_PublicClass.staticPrivateProp_;
};

// File 3.
```

(continues on next page)

(continued from previous page)

```

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

/**
 * @return {number} The number of ducks we've arranged in a row.
 */
AA_SubClass.prototype.method = function() {
  // Legal access of a protected instance property.
  return this.protectedProp;
};

```

JavaScript AA\_PrivateClass\_ public private

### 6.3.9 JavaScript

JSDoc EcmaScript 4

JavaScript

ES4 JavaScript JsDoc

ES4

| Code | JavaScript                                            | ES4                                | ES5                              |
|------|-------------------------------------------------------|------------------------------------|----------------------------------|
|      | <code>{null} {undefined}</code>                       |                                    |                                  |
|      | <code>{boolean} {number}</code>                       |                                    |                                  |
|      | <code>{string}</code>                                 |                                    |                                  |
|      | <code>{Object} {Function} {EventTarget}</code>        | <code>@constructor</code>          |                                  |
|      | <code>EventTarget</code>                              | <code>JSDoc @interface JS-</code>  |                                  |
|      | <code>get</code>                                      |                                    |                                  |
|      | <code>{goog.events. EventType}</code>                 | <code>@enum</code>                 |                                  |
|      | <code>goog.events.EventType</code>                    | <code>JSDoc ES4</code>             |                                  |
|      | <code>{Array.&lt;string&gt;}</code>                   | <code>Java</code>                  |                                  |
|      | <code>{Object. &lt;string, number&gt;}</code>         |                                    |                                  |
|      | <code>{(number boolean)}</code>                       | <code>A?B?</code>                  | <code>{(number, boolean)}</code> |
|      |                                                       | <code>{number boolean}</code>      | <code>{(number boolean)}</code>  |
|      | <code>{?number}</code>                                | <code>syntactic sugar</code>       | <code>{number?}</code>           |
|      | <code>{!Object}</code>                                | <code>null</code>                  | <code>{Object!}</code>           |
|      | <code>{myNum: number, myObject}</code>                | <code>myNum number myObject</code> | <code>length</code>              |
|      | <code>{function(string, boolean)}</code>              |                                    |                                  |
|      | <code>{function(): number}</code>                     |                                    |                                  |
|      | <code>{function(this:goog.ui.Menu, string)}</code>    |                                    |                                  |
|      | <code>goog.ui.Menu</code>                             |                                    |                                  |
|      | <code>{function(new:goog.ui.Menu, string)}</code>     |                                    |                                  |
|      | <code>goog.ui.Menu</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      | Value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Constructor |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| number         | <pre>1 1.0 -5 1e5 Math.PI</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Number      |
| Number         | <pre>new Number(true)</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Number      |
| string         | <pre>'Hello' "World" String(42)</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | String      |
| String         | <pre>new String('Hello') new String(42)</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | String      |
| boolean        | <pre>true false Boolean(0)</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Boolean     |
| Boolean        | <pre>new Boolean(true)</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 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.<string> | <pre>['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten', 'eleven', 'twelve', 'thirteen', 'fourteen', 'fifteen', 'sixteen', 'seventeen', 'eighteen', 'nineteen', 'twenty']</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |
| Object         | <pre>{   foo: 'bar',   baz: 123,   qux: true,   quux: null,   quuz: undefined,   quaa: 'quaa',   quab: 'quab',   quac: 'quac',   quad: 'quad',   quae: 'quae',   quaf: 'quaf',   quag: 'quag',   quah: 'quah',   quai: 'quai',   quaj: 'quaj',   quak: 'quak',   qual: 'qual',   quam: 'quam',   quan: 'quan',   quao: 'quao',   quap: 'quap',   quaq: 'quaq',   quar: 'quar',   quas: 'quas',   quat: 'quat',   quau: 'quau',   quav: 'quav',   quaw: 'quaw',   quax: 'quax',   quay: 'quay',   quaz: 'quaz',   quaa: 'quaa',   quab: 'quab',   quac: 'quac',   quad: 'quad',   quae: 'quae',   quaf: 'quaf',   quag: 'quag',   quah: 'quah',   quai: 'quai',   quaj: 'quaj',   quak: 'quak',   qual: 'qual',   quam: 'quam',   quan: 'quan',   quao: 'quao',   quap: 'quap',   quaq: 'quaq',   quar: 'quar',   quas: 'quas',   quat: 'quat',   quau: 'quau',   quav: 'quav',   quaw: 'quaw',   quax: 'quax',   quay: 'quay',   quaz: 'quaz' }</pre> |             |

???

????????????????????????????????????????????????????????????

```
/** @type {number} */ (x)
```

????????????????

??Javascript????????????????????????????????????????????????????

????????????????????????????

```
/**
 * ?????????
 * @param {Object} value???
 * @constructor
 */
function MyClass(value) {
  /**
   * Some value.
   * @type {Object}
   * @private
   */
  this.myValue_ = value;
}
```

???? myValue\_ ??????null???? myValue\_ ??????null, ???????:

```
/**
 * ???null????????
 * @param {!Object} value???
 * @constructor
 */
function MyClass(value) {
  /**
   * Some value.
   * @type {!Object}
   * @private
   */
  this.myValue_ = value;
}
```

???????????????????? MyClass ??????null????????????

????????????????????????undefined????????????????????

```
/**
 * ?????????????
 * @param {Object=} opt_value????????
 * @constructor
 */
function MyClass(opt_value) {
  /**
   * Some value.
   * @type {Object|undefined}
   * @private
   */
```

(continues on next page)

(continued from previous page)

```

    this.myValue_ = opt_value;
}

```

myValue\_ null undefined  
 : opt\_value {Object=} {Object|undefined} unde-  
 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> <strong> <ul> <ol> <li> <a>`

JSDoc

```
/**
 * Computes weight based on three factors:
 * items sent
 * items received
 * last timestamp
 */
```

Computes weight based on three factors: items sent items received items received last timestamp

```
/**
 * Computes weight based on three factors:
 * <ul>
 * <li>items sent
 * <li>items received
 * <li>last timestamp
 * </ul>
 */
```

JavaDoc doc

```
/**
 * @fileoverview Description of file, its uses and information
 * about its dependencies.
 */
```

## Class

Example 1

```
/**  
 * Class making something fun and easy.  
 * @param {string} arg1 An argument that makes this more interesting.  
 * @param {Array.<number>} arg2 List of numbers to be processed.  
 * @constructor  
 * @extends {goog.Disposable}  
 */  
project.MyClass = function(arg1, arg2) {  
  // ...  
};  
goog.inherits(project.MyClass, goog.Disposable);
```

Example 2

Example 3

```
/**  
 * Operates on an instance of MyClass and returns something.  
 * @param {project.MyClass} obj Instance of MyClass which leads to a long  
 *   comment that needs to be wrapped to two lines.  
 * @return {boolean} Whether something occurred.  
 */  
function PR_someMethod(obj) {  
  // ...  
}
```

Example 4

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

## JSDoc

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

continues on next page



Table 1 – continued from previous page

| ??                 | ??????                                                                                                                                                                                                                                                                                                                                                                 | ??                                                                                                                              |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| <p>@deprecated</p> | <p><b>@deprecated Description</b><br/>???</p> <pre> /**  * Determines whether a  * ↪node is a field.  * @return {boolean}  * ↪True if the contents  * ↪of  *   the element are  * ↪editable, but the  * ↪element  *   itself is not.  * @deprecated Use  * ↪isField().  */ BN_EditUtil. ↪isTopEditableField = ↪function(node) {   // ... };                     </pre> | <p>????????????????????????????????????????????????????????????</p>                                                             |
| <p>@dict</p>       | <p><b>@dict Description</b><br/>???</p> <pre> /**  * @constructor  * @dict  */ function Foo(x) {   this['x'] = x; } var obj = new Foo(123); var num = obj.x; // ↪warning (** @dict \*/ { x: 1 } ↪).x = 123; // ↪warning                     </pre>                                                                                                                     | <p>?????? (?????Foo)????? @dict ???????????????????<br/>                     Foo ??????????????????????????????????????????</p> |
| <p>@enum</p>       | <p><b>@enum {Type}</b><br/>???</p> <pre> /**  * Enum for tri-state  * ↪values.  * @enum {number}  */ project.TriState = {   TRUE: 1,   FALSE: -1,   MAYBE: 0 };                     </pre>                                                                                                                                                                             |                                                                                                                                 |

continues on next page



Table 1 – continued from previous page

| ??            | ??????                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | ??                                                                                                                             |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| @fileoverview | <p><b>@fileoverview</b> 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><b>@implements</b> 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><b>@inheritDoc</b></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

| ??             | ??????                                                                                                                                                                                                                                                                                                                                                   | ??                                                                                                |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| @noalias       | <p data-bbox="451 279 553 306">@noalias</p> <p data-bbox="451 310 505 338">???</p> <pre data-bbox="451 352 802 415">/** @noalias */ function Range() {}</pre>                                                                                                                                                                                            | <p data-bbox="824 279 1321 306">????????????????????????????????????????????????????????????</p>  |
| @nosideeffects | <p data-bbox="451 436 613 464">@nosideeffects</p> <p data-bbox="451 468 505 495">???</p> <pre data-bbox="451 510 802 972">/** @nosideeffects */ function ↳noSideEffectsFn1() {   // ... }; /** @nosideeffects */ var noSideEffectsFn2 = ↳function() {   // ... }; /** @nosideeffects */ a.prototype. ↳noSideEffectsFn3 = ↳function() {   // ... };</pre> | <p data-bbox="824 436 1624 464">????????????????????????????????????????????????????????????</p>  |
| @override      | <p data-bbox="451 991 565 1018">@override</p> <p data-bbox="451 1022 505 1050">???</p> <pre data-bbox="451 1064 802 1413">/**  * @return {string} ↳Human-readable ↳representation of ↳project.SubClass.  * @override  */ project.SubClass. ↳prototype.toString() ↳{   // ... };</pre>                                                                    | <p data-bbox="824 991 1624 1018">????????????????????????????????????????????????????????????</p> |

continues on next page

Table 1 – continued from previous page

| ??                | ?????                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | ??                                                                                                                                                                                                                                                                                                                                                  |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>@param</p>     | <p>@param {Type} varname De-<br/>                     scription<br/>                     ???</p> <pre data-bbox="451 373 803 940">                     /**                     * Queries a Baz for                     * items.                     * @param {number}                     * groupNum Subgroup id                     * to query.                     * @param                     * {string number null}                     * term An itemName,                     * or itemId, or                     * null to search                     * everything.                     */                     goog.Baz.prototype.                     query =                     function(groupNum,                     term) {                     // ...                     };                     </pre> | <p>????????????????????????????????<br/>                     ??? ???? ?????????????????????????????????</p>                                                                                                                                                                                                                                         |
| <p>@private</p>   | <p>@private @private {type}<br/>                     ???</p> <pre data-bbox="451 1024 803 1266">                     /**                     * Handlers that are                     * listening to this                     * logger.                     * @private {!Array.                     * &lt;Function&gt;}                     */                     this.handlers\_ = [];                     </pre>                                                                                                                                                                                                                                                                                                                                                                                                     | <p>???????????????????????????????? ??? ????<br/>                     @private ?????????????????????????????????</p>                                                                                                                                                                                                                                |
| <p>@protected</p> | <p>@protected @protected {type}<br/>                     ???</p> <pre data-bbox="451 1354 803 1854">                     /**                     * Sets the component's                     * root element to the                     * given element.                     * Considered                     * protected and final.                     * @param {Element}                     * element Root element                     * for the component.                     * @protected                     */                     goog.ui.Component.                     prototype.                     setElementInternal =                     function(element) {                     // ...                     };                     </pre>                                                              | <p>???????????????? `?????` &lt;<a href="http://google-styleguide.googlecode.com/svn/trunk/javascriptguide.xml#Visibility__private_and_protected_fields_">http://google-styleguide.googlecode.com/svn/trunk/javascriptguide.xml#Visibility__private_and_protected_fields_</a>&gt;`_`<br/>                     ?????????????????????????????????</p> |

continues on next page



Table 1 – continued from previous page

| ??                | ?????                                                                                                                                                                                                                                                                  | ??                                      |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| <p>@supported</p> | <p><b>@supported</b> Description<br/>???</p> <pre> /**  * @fileoverview Event_  * ↪Manager  * Provides an_  * ↪abstracted interface_  * ↪to the  * browsers' event_  * ↪systems.  * @supported So far_  * ↪tested in IE6 and_  * ↪FF1.5  */                     </pre> | <p>????????????????????????????????</p> |
| <p>@suppress</p>  | <p><b>@suppress</b> {warning1 warning2}<br/>???</p> <pre> /**  * @suppress  * ↪{deprecated}  */ function f() {   ↪   ↪deprecatedVersionOfF();   ↪ }                     </pre>                                                                                         | <p>??????????????????????????????</p>   |
| <p>@template</p>  | <p><b>@template</b><br/>???</p> <pre> /**  * @param  * ↪{function(this:T, ...  * ↪)} fn  * @param {T} thisObj  * @param {...*} var_  * ↪args  * @template T  */ goog.bind =_ ↪function(fn, thisObj, ↪ var_args) {   ... };                     </pre>                  | <p>?????????????????? ???? ?</p>        |

continues on next page



- @default
- @event
- @example
- @field
- @function
- @ignore
- @inner
- @link
- @memberOf
- @name
- @namespace
- @property
- @public
- @requires
- @returns
- @since
- @static
- @version

### 6.3.11 goog.provide

Example

goog.provide('namespace.MyClass');

Example

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

Example

```

goog.provide('namespace.MyClass');
goog.provide('namespace.MyClass.Enum');
goog.provide('namespace.MyClass.InnerClass');
goog.provide('namespace.MyClass.TypeDef');
goog.provide('namespace.MyClass.CONSTANT');
goog.provide('namespace.MyClass.staticMethod');
```

Example

```

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

### 6.3.12

JS Closure Compiler

### 6.3.13

JavaScript

True False

false

- null
- undefined
- ""
- 0

true

- "0"
- []
- {}

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

x=0 false

```
while (x) {
```

null

```
if (y != null && y != '') {
```

```
if (y) {
```

- Boolean('0') == true '0' != true
- 0 != null 0 == [] 0 == false
- Boolean(null) == false null != true null != false
- Boolean(undefined) == false undefined != true undefined != false
- Boolean([]) == true [] != true [] == false
- Boolean({}) == true {} != true {} != false

ternary operator

ternary operator

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

ternary operator

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

ternary operator HTML

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

&& ||

ternary operator, logical operators

“||” default value

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

ternary operator

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

“&&” short-circuiting

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

ternary operator

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

??????

```
var kid = node && node.kids && node.kids[index];
if (kid) {
  foo(kid);
}
```

????????????????

```
node && node.kids && node.kids[index] && foo(node.kids[index]);
```

??????

????????????????????????????????????????????????????????????lengthO(n)lengthO(n^2)

```
var paragraphs = document.getElementsByTagName('p');
for (var i = 0; i < paragraphs.length; i++) {
  doSomething(paragraphs[i]);
}
```

??????

```
var paragraphs = document.getElementsByTagName('p');
for (var i = 0, paragraph; paragraph = paragraphs[i]; i++) {
  doSomething(paragraph);
}
```

????????????????(????????????false) ?????

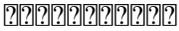
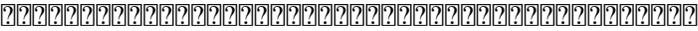
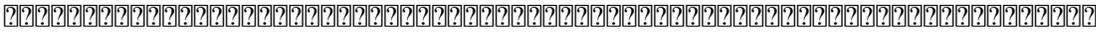
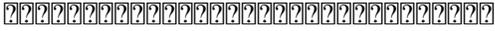
????????????????firstChild?nextSibling????????

```
var parentNode = document.getElementById('foo');
for (var child = parentNode.firstChild; child; child = child.nextSibling) {
  doSomething(child);
}
```

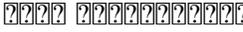
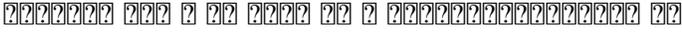
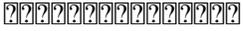


## TYPESCRIPT

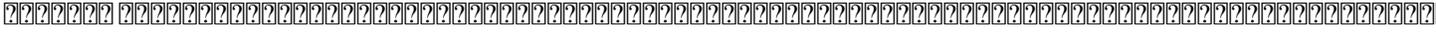
### 7.1

 TypeScript  TypeScript  
  


#### 7.1.1

 RFC 2119   


#### 7.1.2



#### 7.1.3



2024  02  29 



- Frank Li



Google TypeScript Style Guide



### ???

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 units = '\u03bcs'; // Greek letter mu, 's'

// 
const output = '\uffff' + content;
```

### 7.2.3

#### JSDoc

Typescript JSDoc /\*\* ... \*/ // ... /\*\* ... \*/

- /\*\* JSDoc \*/
- /\*\* //

JSDoc

#### JSDoc

JSDoc      JavaScript      JavaScript

#### 

```
/** JSDoc */
/** JSDoc */
```

#### TypeScript

```
@param @return implements enum private @imple-
ments @enum @private
```

@override

```
TypeScript @override @override
```

#### 

```
/** 
/** @param fooBarService Foo Bar */
```

```
@param @return
```

```

/**
 * POST
 * @param amountLitres
 */
brew(amountLitres: number, logger: Logger) {
  // ...
}

```

### Annotations

Annotations are used to provide metadata for the classes and methods in your code.

```

class Foo {
  constructor(private readonly bar: Bar) { }
}

```

Foo has a constructor that takes a Bar parameter.

Annotations can be used to describe parameters in JSDoc.

```

/**
 * @param percolator
 * @param beans
 */
class ParamProps {
  constructor(
    private readonly percolator: Percolator,
    private readonly beans: CoffeeBean[] {}
  ) {}
}

```

```

/**
 * @param initialBean
 */
class OrdinaryClass {
  /**
   * nextBean
   */
  nextBean: CoffeeBean;

  constructor(initialBean: CoffeeBean) {
    this.nextBean = initialBean;
  }
}

```

### Decorators

Decorators are a way to modify classes and methods at runtime.

```

// Decorator function
new Percolator().brew(/* amountLitres= */ 5);

// Decorator function
new Percolator().brew({amountLitres: 5});

```

```

/**
 * @link CoffeeBrewer
 */
export class Percolator implements CoffeeBrewer {
  /**

```

(continues on next page)

(continued from previous page)

```

*
* @param amountLitres
*/
brew(amountLitres: number) {
  //
  // TODO(b/12345):
}
}

```

## 

@Component JSDoc JSDoc  
 JSDoc

```

// @Component FooComponent
@Component({
  selector: 'foo',
  template: 'bar',
})
/** "bar" */
export class FooComponent {}

```

JSDoc

```

/** "bar" */
@Component({
  selector: 'foo',
  template: 'bar',
})
export class FooComponent {}

```

## 7.3

### 7.3.1

- 
- 
- TypeScript public public readonly public public

```

class Foo {
  public bar = new Bar(); // public

  constructor(public readonly baz: Baz) {} // readonly baz
  ↪public public
}

```

```

class Foo {
  bar = new Bar(); // 
  constructor(public baz: Baz) {} // 
}

```

### 7.3.2

```

// 
const x = new Foo();

// 
const x = new Foo();

```

ES2015

```

// 
class UnnecessaryConstructor {
  constructor() {}
}

```

```

// 
class UnnecessaryConstructorOverride extends Base {
  constructor(value: number) {
    super(value);
  }
}

```

```

// 
class DefaultConstructor {
}

// 
class ParameterProperties {
  constructor(private myService) {}
}

// 
class ParameterDecorators {
  constructor(@SideEffectDecorator myService) {}
}

// 
class NoInstantiation {
  private constructor() {}
}

```

### 7.3.3

#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;
  }
}
```

```
//
class Foo {
  constructor(private readonly barService: BarService) {}
}
```

JSDoc @param

### Accessors

Accessors are used to get and set the value of a property.

```
// Accessors
class Foo {
  private readonly userList: string[];
  constructor() {
    this.userList = [];
  }
}
```

```
// Accessors
class Foo {
  private readonly userList: string[] = [];
}
```

### Property Visibility

Property visibility is used to control access to class members. In TypeScript, you can use `public`, `protected`, and `private` to control access to class members.

`public` is the default visibility. `protected` is used for members that should only be accessible to the class and its subclasses. `private` is used for members that should only be accessible to the class.

AngularJS uses `private` for controller members.

### Property Visibility

Property visibility is used to control access to class members.

```
class Foo {
  constructor(private readonly someService: SomeService) {}

  get someMember(): string {
    return this.someService.someVariable;
  }

  set someMember(newValue: string) {
    this.someService.someVariable = newValue;
  }
}
```

Property visibility is used to control access to class members.

```
class Foo {
  private wrappedBar = '';
  get bar() {
```

(continues on next page)

(continued from previous page)

```

        return this.wrappedBar || 'bar';
    }

    set bar(wrapped: string) {
        this.wrappedBar = wrapped.trim();
    }
}

```

```

class Bar {
    private barInternal = '';
    // barInternal is private, bar is public
    get bar() {
        return this.barInternal;
    }

    set bar(value: string) {
        this.barInternal = value;
    }
}

```

### 7.3.4 Enum

TypeScript defines several built-in enums: String, Boolean, Number, and new Boolean(false) and true.

```

// Enum
const s = new String('hello');
const b = new Boolean(false);
const n = new Number(5);

```

```

// Enum
const s = 'hello';
const b = false;
const n = 5;

```

### 7.3.5 Array

TypeScript defines Array() constructor to create new arrays.

```

// Array
const a = new Array(2); // 2 undefined [undefined, undefined]
const b = new Array(2, 3); // 2, 3 [2, 3]

```

Array() constructor from Array

```

const a = [2];
const b = [2, 3];

// Array(2)
const c = [];
c.length = 2;

```

(continues on next page)

(continued from previous page)

```
// [0, 0, 0, 0, 0]
Array.from<number>({length: 5}).fill(0);
```

### 7.3.6

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) {...}
```

### 7.3.7

const let const var

```
const foo = otherValue; // foo const
let bar = someValue;    // bar let
```

const let var JavaScript bug TypeScript var

```
// 
var foo = someValue;
```

### 7.3.8

new Error() Error() new

```
// 
throw new Error('Foo is not a valid bar.');
```

```
// 
throw Error('Foo is not a valid bar.');
```

### 7.3.9

for (... in ...) for (... in ...)

```
// 
for (const x in someObj) {
  // x someObj 
}
```

if for (... of Object.keys(...))

```
// 
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 
}
```

### 7.3.10

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

### 7.3.11

[...foo]{...bar}

```
const foo = {
  num: 1,
};

const foo2 = {
  ...foo,
  num: 5,
};

const foo3 = {
  num: 5,
  ...foo,
}

// foo2 num 1 num 5
foo2.num === 5;

// foo3 num 5 num 1
foo3.num === 1;
```

null undefined

```
// 
const foo = {num: 7};
const bar = {num: 5, ...(shouldUseFoo && foo)}; // undefined
```

```
// 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'];
```

### 7.3.12 for / if

for

```
// Example
for (let i = 0; i < x; i++) {
  doSomethingWith(i);
  andSomeMore();
}
if (x) {
  doSomethingWithALongMethodName(x);
}
```

```
// Example
if (x)
  x.doFoo();
for (let i = 0; i < x; i++)
  doSomethingWithALongMethodName(i);
```

if

```
// Example
if (x) x.doFoo();
```

### 7.3.13 switch

switch default

```
// Example
switch (x) {
  case Y:
    doSomethingElse();
    break;
  default:
    // ...
}
```

case ...

```
// Example
switch (x) {
  case X:
    doSomething();
    // ...
  case Y:
    // ...
}
```

switch

```
// Example
switch (x) {
  case X:
  case Y:
    doSomething();
}
```

(continues on next page)

(continued from previous page)

```
break;
default: //
}
```

### 7.3.14

JavaScript `===` `!==` JavaScript

```
//
if (foo == 'bar' || baz != bam) {
  //
}
```

```
//
if (foo === 'bar' || baz !== bam) {
  //
}
```

`null` `==` `!` `null` `undefined`

```
//
if (foo == null) {
  // foo null undefined
}
```

### 7.3.15

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;
}
```

(continues on next page)

(continued from previous page)

```
const fooSearch: SearchFunction = (source, subString) => { ... };
```

### 7.3.16

#### 

ES6 function

```
// 
bar(() => { this.doSomething(); })
```

```
// 
bar(function() { ... })
```

this function this this

#### 

```
// 
function someFunction() {
  // 
  const receipts = books.map(b: Book) => {
    const receipt = payMoney(b.price);
    recordTransaction(receipt);
    return receipt;
  });

  // 
  const longThings = myValues.filter(v => v.length > 1000).map(v => String(v));

  function payMoney(amount: number) {
    // this
  }
}
```

```
// 
myPromise.then(v => console.log(v));
```

```
// 
myPromise.then(v => {
  console.log(v);
});

// 
const transformed = [1, 2, 3].map(v => {
  const intermediate = someComplicatedExpr(v);
```

(continues on next page)



(continued from previous page)

```

// this
setTimeout(this.patienceTracker, 5000);
}
private patienceTracker = () => {
    this.waitedPatiently = true;
}
}

```

```

// this
class DelayHandler {
    constructor() {
        //
        setTimeout(() => {
            this.patienceTracker();
        }, 5000);
    }
    private patienceTracker() {
        this.waitedPatiently = true;
    }
}

```

???

this

```

//
class Component {
    onAttached() {
        //
        this.addEventListener('click', () => {
            this.listener();
        });
        // this.listener
        window.addEventListener('onbeforeunload', this.listener);
    }
    onDetached() {
        // window this.listener
        // this this
        window.removeEventListener('onbeforeunload', this.listener);
    }
    // this
    private listener = () => {
        confirm('Do you want to exit the page?');
    }
}

```

bind

```

// bind
class Component {
    onAttached() {
        //
        window.addEventListener('onbeforeunload', this.listener.bind(this));
    }
}

```

(continues on next page)

(continued from previous page)

```

onDetached() {
  // bind
  window.removeEventListener('onbeforeunload', this.listener.bind(this));
}
private listener() {
  confirm('Do you want to exit the page?');
}
}

```

### 7.3.17

ASI Bug clang-format

### 7.3.18 @ts-ignore

@ts-ignore any

### 7.3.19

x as SomeType y! any

```

//
(x as Foo).foo();

y!.bar();

```

```

//
// Foo
if (x instanceof Foo) {
  x.foo();
}

if (y) {
  y.bar();
}

```

```

//
// x Foo
(x as Foo).foo();

```

(continues on next page)



(continued from previous page)

```

memberA: string,
memberB: number,
}

```

Example 7.3.1: Defining a type alias

```

// Example 7.3.1
type SomeTypeAlias = {
  memberA: string,
  memberB: number,
};

let someProperty: {memberC: string, memberD: number};

```

Example 7.3.2: Accessing a property

Example 7.3.3: Using a type alias

```

// Example 7.3.3
// Example 7.3.3
console.log(x['someField']);
console.log(x.someField);

```

Example 7.3.4: Using a type alias

```

// Example 7.3.4
declare interface ServerInfoJson {
  appVersion: string;
  user: UserJson;
}

const data = JSON.parse(serverResponse) as ServerInfoJson;
console.log(data.appVersion); // Example 7.3.4

```

Example 7.3.5: Using a type alias

Example 7.3.6: Using a type alias

```

// Example 7.3.6
import {method1, method2} from 'utils';
class A {
  readonly utils = {method1, method2};
}

```

```

// Example 7.3.7
import * as utils from 'utils';
class A {
  readonly utils = utils;
}

```

### 7.3.20

Web

### 7.3.21

enum TypeScript const enum JavaScript

### 7.3.22 debugger

debugger

```
// function debugMe() { debugger; }
```

### 7.3.23

@MyDecorator

- Angular @Component @NgModule
Polymer @property

TC39 Bug

```
/** JSDoc */
@Component({...}) //
class MyComp {
  @Input() myField: string; // .....
  @Input()
  myOtherField: string; // .....
}
```

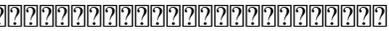
## 7.4

### 7.4.1

#### 

TypeScript . .. root/path/to/file

./foo path/to/foo

../../../../ 

```
import {Symbol1} from 'google3/path/from/root';
import {Symbol2} from '../parent/file';
import {Symbol3} from './sibling';
```

## 

### 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 

## 7.4.2

```
// Use named exports:
export class Foo { ... }
```

```
// 
export default class Foo { ... }
```

```
// 
import Foo from './bar'; // 
import Bar from './bar'; // 
```

 foo.ts 

```
// 
const foo = 'blah';
export default foo;
```

bar.ts

```
// 
import {fizz} from './foo';
```

error TS2614: Module './foo' has no exported member 'fizz'

```
// 
import fizz from './foo';
```

fizz === foo

```
// 
export default class Foo {
  static SOME_CONSTANT = ...
  static someHelpfulFunction() { ... }
  ...
}
```

Foo

```
// 
export const SOME_CONSTANT = ...
export function someHelpfulFunction()
export class Foo {
  // Foo
}
```

### 

TypeScript API

### 

export let

```
// 
export let foo = 3;
// ES6 foo foo
// TypeScript foo
window.setTimeout(() => {
  foo = 4;
}, 1000 /* ms */);
```

```
// 
let foo = 3;
window.setTimeout(() => {
  foo = 4;
```

(continues on next page)

(continued from previous page)

```
}, 1000 /* ms */);
//                                                                    
export function getFoo() { return foo; };
```

```
function pickApi() {
  if (useOtherApi()) return OtherApi;
  return RegularApi;
}
export const SomeApi = pickApi();
```

### 

```
//                    
export class Container {
  static FOO = 1;
  static bar() { return 1; }
}
```

```
//                    
export const FOO = 1;
export function bar() { return 1; }
```

### 7.4.3

 ES6  TypeScript                  

|              |   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | import * as foo from '...';                                                                                                                                             | TypeScript                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|              | import {Something} from '...';                                                                                                                                          | TypeScript                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|              | import Something from '...';                                                                                                                                            |                  |
|              | import '...';                                                                                                                                                           |                  |

```
//                    
import * as ng from '@angular/core';
import {Foo} from './foo';

//                    
import Button from 'Button';

//                    
import 'jasmine';
import '@polymer/paper-button';
```

### Imports

Imports

Imports \* as tableview from './tableview';  
API

Jasmine describe it

```
// Imports
import {TableViewItem, TableViewHeader, TableViewRow, TableViewModel,
TableViewRenderer} from './tableview';
let item: TableViewItem = ...;
```

```
// Imports
import * as tableview from './tableview';
let item: tableview.Item = ...;
```

```
import * as testing from './testing';

// Imports
// Imports
// Imports
testing.describe('foo', () => {
testing.it('bar', () => {
  testing.expect(...);
  testing.expect(...);
});
});
```

```
// Imports
import {describe, it, expect} from './testing';

describe('foo', () => {
it('bar', () => {
  expect(...);
  expect(...);
});
});
...
```

### Imports

import {Something as SomeOtherThing}

Imports

- Imports
- Imports
- Imports RxJS from observableFrom

import type export type

import type ... from export type ... from

Tip: export type Foo = ...;

```
// 
import type {Foo} from './foo';
export type {Bar} from './bar';
```

```
// 
import {Foo} from './foo';
export {Bar} from './bar';
```

TypeScript

```
import type import import type 
import '...' 
export type API import type 
export type API 
UserService AjaxUserService
```

### 7.4.4

products checkout backend views models controllers

## 7.5

### 7.5.1

TypeScript google3 any

```
const x = 15; // x
```

string number boolean RegExp new

```
// boolean 
const x: boolean = true;
```

```
// Set 
const x: Set<string> = new Set();
```

```
// TypeScript 
const x = new Set<string>();
```



undefined

TypeScript

```
interface CoffeeOrder {
  sugarCubes: number;
  milk?: Whole|LowFat|HalfHalf;
}

function pourCoffee(volume?: Milliliter) { ... }
```

|undefined {sugarCubes: 1} CoffeeOrder milk

|undefined

```
class MyClass {
  field = '';
}
```

### 7.5.3

TypeScript

Mock

```
// 
const foo: Foo = {
  a: 123,
  b: 'abc',
}
```

```
// 
const badFoo = {
  a: 123,
  b: 'abc',
}
```

badFoo badFoo

badFoo Foo

```
interface Animal {
  sound: string;
  name: string;
}

function makeSound(animal: Animal) {}

/**
```

(continues on next page)

(continued from previous page)

```

* 'cat' '{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);

```

### 7.5.4

TypeScript

```

//
interface User {
  firstName: string;
  lastName: string;
}

```

```

//
type User = {
  firstName: string,
  lastName: string,
}

```

TypeScript

### 7.5.5 Array<T>

TypeScript `Array<T>` is a type that represents an array of elements of type `T`. It is a subtype of `T[]`.

```
// Example 1
const a: string[];
const b: readonly string[];
const c: ns.MyObj[];
const d: Array<string|number>;
const e: ReadonlyArray<string|number>;
```

```
// Example 2
const f: Array<string>; // Array<string>
const g: ReadonlyArray<string>;
const h: {n: number, s: string}[]; // Array<{n: number, s: string}>
const i: (string|number)[];
const j: readonly (string|number)[];
```

### 7.5.6 {[key: string]: number}

JavaScript `{[key: string]: number}` is a type that represents an object with string keys and number values.

```
const fileSizes: {[fileName: string]: number} = {};
fileSizes['readme.txt'] = 541;
```

TypeScript `{[key: string]: number}` is a type that represents an object with string keys and number values.

```
// Example 1
const users: {[key: string]: number} = ...;
```

```
// Example 2
const users: {[userName: string]: number} = ...;
```

TypeScript `Record<Keys, ValueType>` is a type that represents an object with keys of type `Keys` and values of type `ValueType`. It is a supertype of `{[key: string]: number}`.

TypeScript `Record<Keys, ValueType>` is a type that represents an object with keys of type `Keys` and values of type `ValueType`.

### 7.5.7 Partial<T>

TypeScript `Partial<T>` is a type that represents an object with the same keys as `T`, but where the values are optional. It is a subtype of `T`.

TypeScript `Partial<T>` is a type that represents an object with the same keys as `T`, but where the values are optional.

- TypeScript `Partial<T>` is a type that represents an object with the same keys as `T`, but where the values are optional.
- TypeScript `Partial<T>` is a type that represents an object with the same keys as `T`, but where the values are optional.
- TypeScript `Partial<T>` is a type that represents an object with the same keys as `T`, but where the values are optional.

- IDE “” Pick<T, Keys>

- 
- 
- 

TypeScript Pick<T, Keys> T

```
interface User {
  shoeSize: number;
  favoriteIcecream: string;
  favoriteChocolate: string;
}

// FoodPreferences favoriteIcecream favoriteChocolate shoeSize
type FoodPreferences = Pick<User, 'favoriteIcecream'|'favoriteChocolate'>;
```

FoodPreferences

```
interface FoodPreferences {
  favoriteIcecream: string;
  favoriteChocolate: string;
}
```

User FoodPreferences User FoodPreferences

```
interface FoodPreferences { /* */ }

interface User extends FoodPreferences {
  shoeSize: number;
  // User FoodPreferences
}
```

IDE

### 7.5.8 any

TypeScript any

any

- 
- unknown any
- Lint any

## Enums

### Enums

```
// Enumerated type JSON
declare interface MyUserJson {
  name: string;
  email: string;
}

// Enumerated type
type MyType = number|string;

// Enumerated type
function getTwoThings(): {something: number, other: string} {
  // ...
  return {something, other};
}

// Enumerated type any
// Enumerated type
// Enumerated type "enum"
function nicestElement<T>(items: T[]): T {
  // 1 items
  // Enumerated type T extends HTMLElement
}

```

### Enum unknown any

any  
unknown unknown any

```
// Enum
// Enum null undefined val
// Enum
const val: unknown = value;
```

```
// Enum
const danger: any = value /* Enum */;
danger.whoops(); // Enum
```

### Lint any

any Mock Lint any

```
// Enum BookService
// Enum Mock
// tslint:disable-next-line:no-any
const mockBookService = ({get() { return mockBook; }} as any) as BookService;
// Enum
// tslint:disable-next-line:no-any
const component = new MyComponent(mockBookService, /* unused ShoppingCart */ null as any);
```

### 7.5.9

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);
```

### 7.5.10

JavaScript

- String Boolean Number string boolean number
- Object {} object {} null undefined " " symbol bigint

### 7.5.11

API API

## 7.6

### 7.6.1

- 1.

- any
- TypeScript
- .
- 
- private

- 2.

JavaScript

- 
- x as T
- Array<[number, number]>

- 3.

TypeScript

- 
- Closure TS
- 
- google3

- 4.

Bug



## HTML/CSS [?] - [?]

### 8.1 [?]

HTML/CSS [?] HTML CSS [?]

### 8.2 [?]

#### 8.2.1 [?]

[?]

URL [?] http: https: [?] URL [?]

```
<!-- [?] -->
<script src="http://www.google.com/js/gweb/analytics/autotrack.js"></script>

<!-- [?] -->
<script src="//www.google.com/js/gweb/analytics/autotrack.js"></script>
```

```
/* [?] */
.example {
  background: url(http://www.google.com/images/example);
}

/* [?] */
.example {
  background: url(//www.google.com/images/example);
}
```

### 8.3 [?]

#### 8.3.1 [?]

[?]

TAB [?] TAB [?]

```
<ul>
  <li>Fantastic
  <li>Great
</ul>
```

```
.example {
  color: blue;
}
```

### 8.3.2

HTML text/CDATA CSS

```
<!-- -->
<A HREF="/">Home</A>

<!-- -->

```

```
/* */
color: #E5E5E5;

/* */
color: #e5e5e5;
```

### 8.3.3

```
<!-- -->
<p>What?_

<!-- -->
<p>Yes please.
```

## 8.4

### 8.4.1

UTF-8 BOM

UTF-8

HTML <meta charset="utf-8"> UTF-8

Handling character encodings in HTML and CSS

## 8.4.2

HTML/CSS

## 8.4.3

```

<!-- TODO: -->
<center>Test</center>
<ul>
  <li>Apples</li>
  <li>Oranges</li>
</ul>
  
```

```

{# TODO(john.doe): #}
<center>Test</center>

<!-- TODO: -->
<ul>
  <li>Apples</li>
  <li>Oranges</li>
</ul>
  
```

## 8.5 HTML

### 8.5.1

HTML5

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
  
```

### 8.5.2 HTML

W3C HTML validator

```

<!-- -->
<title>Test</title>
<article>This is only a test.

<!-- -->
<!DOCTYPE html>
<meta charset="utf-8">
  
```

(continues on next page)

(continued from previous page)

```
<title>Test</title>
<article>This is only a test.</article>
```

### 8.5.3

HTML

“” p a

HTML

```
<!-- -->
<div onclick="goToRecommendations();">All recommendations</div>

<!-- -->
<a href="recommendations/">All recommendations</a>
```

### 8.5.4

canvas alt

alt CSS alt=""

```
<!-- -->


<!-- -->

```

### 8.5.5

HTML HTML

HTML

```
<!-- -->
<!DOCTYPE html>
<title>HTML sucks</title>
<link rel="stylesheet" href="base.css" media="screen">
<link rel="stylesheet" href="grid.css" media="screen">
<link rel="stylesheet" href="print.css" media="print">
<h1 style="font-size: 1em;">HTML sucks</h1>
<p>I've read about this on a few sites but now I'm sure:
  <u>HTML is stupid!!1</u>
```

(continues on next page)

(continued from previous page)

```
<center>I can't believe there's no way to control the styling of
  my website without doing everything all over again!</center>

<!-- -->
<!DOCTYPE html>
<title>My first CSS-only redesign</title>
<link rel="stylesheet" href="default.css">
<h1>My first CSS-only redesign</h1>
<p>I've read about this on a few sites but today I'm actually
  doing it: separating concerns and avoiding anything in the HTML of
  my website that is presentational.
<p>It's awesome!
```

### 8.5.6

UTF-8 &mdash; &rdquo; &#x263a; HTML

```
<!-- -->
The currency symbol for the Euro is &ldquo;&eur;&rdquo;.

<!-- -->
The currency symbol for the Euro is "€".
```

### 8.5.7

HTML5 Web

```
<!-- -->
<!DOCTYPE html>
<html>
  <head>
    <title>Spending money, spending bytes</title>
  </head>
  <body>
    <p>Sic.</p>
  </body>
</html>

<!-- -->
<!DOCTYPE html>
<title>Saving money, saving bytes</title>
<p>Qed.
```

### 8.5.8 type

type

CSS JavaScript type

HTML5 text/css text/javascript type

## 8.6 HTML

### 8.6.1

CSS display

li Linter

```

<blockquote>
  <p><em>Space</em>, the final frontier.</p>
</blockquote>

<ul>
  <li>Moe
  <li>Larry
  <li>Curly
</ul>

<table>
  <thead>
    <tr>
      <th scope="col">Income
      <th scope="col">Taxes
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>$ 5.00
      <td>$ 4.50
    </tr>
  </tbody>
</table>

```

### 8.6.2 HTML

```

<!-- -->
<a class='maia-button maia-button-secondary'>Sign in</a>

<!-- -->
<a class="maia-button maia-button-secondary">Sign in</a>

```



### 8.7.4

```

id class
id class
id class

```

```

/* */
ul#example {}
div.error {}

/* */
#example {}
.error {}

```

### 8.7.5

```

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;

```

### 8.7.6

```

"O"

```

```

margin: 0;
padding: 0;

```

### 8.7.7

“0”  
 1

```
font-size: .8em;
```

### 8.7.8

3  
 3

```
/*  */
color: #eebbcc;

/*  */
color: #ebc;
```

### 8.7.9

idclass

```
.adw-help {} /* AdWords */
#maia-note {} /* Maia */
```

### 8.7.10 idclass

ID

```
/* :  */
.demoimage {}

/*  */
.error_status {}

/*  */
#video-id {}
.ads-sample {}
```

## 8.7.11 Hacks

CSS “hacks”

CSS hacks

## 8.8 CSS

### 8.8.1

css

CSS -moz -webkit

```
background: fuchsia;
border: 1px solid;
-moz-border-radius: 4px;
-webkit-border-radius: 4px;
border-radius: 4px;
color: black;
text-align: center;
text-indent: 2em;
```

### 8.8.2

```
@media screen, projection {
  html {
    background: #fff;
    color: #444;
  }
}
```

### 8.8.3

```
/* */
.test {
  display: block;
  height: 100px;
}
/* */
```

(continues on next page)

(continued from previous page)

```
.test {
  display: block;
  height: 100px;
}
```

### 8.8.4 CSS

```
/* */
h3 {
  font-weight:bold;
}

/* */
h3 {
  font-weight: bold;
}
```

### 8.8.5

```
/* */
#video{
  margin-top: 1em;
}

/* */
#video
{
  margin-top: 1em;
}

/* */
#video {
  margin-top: 1em;
}
```

### 8.8.6

```

/* */
a:focus, a:active {
  position: relative; top: 1px;
}

/* */
h1,
h2,
h3 {
  font-weight: normal;
  line-height: 1.2;
}

```

### 8.8.7 CSS

```

html {
  background: #fff;
}

body {
  margin: auto;
  width: 50%;
}

```

### 8.8.8 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;
}

```

## 8.9 CSS

### 8.9.1

css

```
/* Header */  
#adw-header {}  
  
/* Footer */  
#adw-footer {}  
  
/* Gallery */  
.adw-gallery {}
```

## 8.10

"#"

""







Tip: ASCII

### 9.4 3.

- 1.
- 2.
- 3. Import statements
- 4.

#### 9.4.1 3.1.

#### 9.4.2 3.2.

Column limit: 100

#### 9.4.3 3.3.

##### 3.3.1.

##### 3.3.2.

Column limit: 100

##### 3.3.3.

- 
- 

```

ASCII
import packageA.ClassA; import packageA.
ClassA.; ASCII';

```

### 3.3.4.

## 9.4.4 3.4.

### 3.4.1.

### 3.4.2.

### 3.4.2.1.

static private

## 9.5 4.

4.8.3.1

### 9.5.1 4.1.

#### 4.1.1.

if else for do while

lambda

#### 4.1.2. K & R

Kernighan & Ritchie

- 
- 
- 
- else

;

```

return () -> {
    while (condition()) {
        method();
    }
};

return new MyClass() {
    @Override public void method() {
        if (condition()) {
            try {
                something();
            } catch (ProblemException e) {
                recover();
            }
        } else if (otherCondition()) {
            somethingElse();
        } else {
            lastThing();
        }
        {
            int x = foo();
            frob(x);
        }
    }
};

```

4.8.1,

### 4.1.3.

K & R 4.1.2 K & R if/else try/catch/finally

```

// 
void doNothing() {}

// 
void doNothingElse() {
}

```

```

// 
try {
    doSomething();
} catch (Exception e) {}

```

### 9.5.2 4.2. `try-with-resources`

4.1.2 `try-with-resources` & `try`

### 9.5.3 4.3. `try`

### 9.5.4 4.4. `try` 100

Java `try` 100 `try` "try" `Unicode` 4.5 `try`

Tip: `try` `Unicode` `try`

- `Javadoc` `URL` `JSNI`
- `3.2` `3.3`
- `shell`
- `google-java-format`

### 9.5.5 4.5. `try`

Tip: `try`

Tip: `try`

### 4.5.1. `try`

1. `try` `Google` `C++` `JavaScript`
  - `try`
    - \* `try` . `try`
    - \* `try` :: `try`
    - \* `try` `<T extends Foo & Bar>`
    - \* `catch` `catch (FooException | BarException e)`
2. `try`



- `<T extends Foo & Bar>`
  - `catch (FooException | BarException e)`
  - `for` / `foreach` :
  - `lambda (String str) -> str.length()`
  - 
  - `:: Object::toString`
  - `. object.toString()`
- 5. `, : ; ()`
  - 6. `//`
  - 7. `//`
  - 8. `List<String> list`
  - 9. 
    - `new int[] {5, 6} and new int[] { 5, 6 }`
  - 10. `[] ...`

### 4.6.3.

Google

```
private int x; //
private Color color; //

private int x; //
private Color color; //
```

Tip: conflicts

### 9.5.7 4.7.

Java

### 9.5.8 4.8. Enum

#### 4.8.1. Enum

enum Answer {

```
private enum Answer {
    YES {
        @Override public String toString() {
            return "yes";
        }
    },
    NO,
    MAYBE
}
```

enum Suit { CLUBS, HEARTS, SPADES, DIAMONDS }

```
private enum Suit { CLUBS, HEARTS, SPADES, DIAMONDS }
```

enum Suit { CLUBS, HEARTS, SPADES, DIAMONDS }

#### 4.8.2. Enum

##### 4.8.2.1. Enum

int a, b; for

##### 4.8.2.2. Enum

#### 4.8.3. Enum

##### 4.8.3.1. Enum

int[] arr = { 0, 1, 2, 3 }

```
new int[] {
    0, 1, 2, 3
}

new int[] {
    0, 1,
    2, 3
}

new int[] {
    0,
    1,
}
```

(continues on next page)

(continued from previous page)

```

    2,
    3
}

new int[]
    {0, 1, 2, 3}

```

#### 4.8.3.2. `String[] args`

`String[] args`

#### 4.8.4. `switch`

`switch` `case` `FOO:` `default:`

##### 4.8.4.1. `break`

`switch`

##### 4.8.4.2. `break`, `continue`, `return`

`switch` `break` `continue` `return`  
 // fall through `switch`

```

switch (input) {
    case 1:
    case 2:
        prepareOneOrTwo();
        // fall through
    case 3:
        handleOneTwoOrThree();
        break;
    default:
        handleLargeNumber(input);
}

```

`switch` `case 1`

#### 4.8.4.3. default

```

switch default
enum switch default
IDE

```

#### 4.8.5.

##### 4.8.5.1.

```

@Target (ElementType.TYPE_USE)

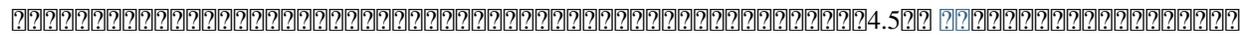
```

```

final @Nullable String name;
public @Nullable Person getPersonByName (String name)

```

##### 4.8.5.2.

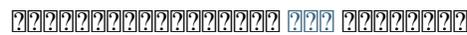
 4.5 

```

@Deprecated
@CheckReturnValue
public final class Frozzler { ... }

```

##### 4.8.5.3.

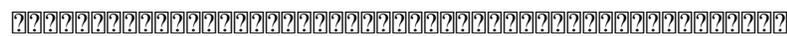


```

@Deprecated
@Override
public String getNameIfPresent () { ... }

```

##### 4.8.5.4.



```

@Partial @Mock DataLoader loader;

```

#### 4.8.5.5. `/**`

`/**`

#### 4.8.6. `/**`

`/**`Javadoc `/**` Javadoc `/**`

`/**`

#### 4.8.6.1. `/**`

`/** ... */ // ... /** ... */ *  
/** * /**`

```
/*  
 * This is           // And so           /* Or you can  
 * okay.             // is this.         * even do this. */  
*/
```

`/**`

Tip: `/** ... */`  
`/** // ... /**`

#### 4.8.7. `/**`

`/**`Java`/**`

```
public protected private abstract default static final transient volatile_  
↪synchronized native strictfp
```

#### 4.8.8. `/**`

`long L 1 3000000000L 3000000000L`

### 9.6 5. `/**`

#### 9.6.1 5.1. `/**`

`ASCII \w+`  
`Google name_ mName s_name kName`

## 9.6.2 5.2. `com.example.deepspace`

### 5.2.1. `com.example.deep_space`

`com.example.deepspace` `com.example.deep_space`

### 5.2.2. `UpperCamelCase`

`UpperCamelCase`

`Character` `ImmutableList` `List`  
`Readable`

`Test` `HashIntegrationTest` `Test` `HashImplTest`

`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`

```
// Constants
static final int NUMBER = 5;
static final ImmutableList<String> NAMES = ImmutableList.of("Ed", "Ann");
static final Map<String, Integer> AGES = ImmutableMap.of("Ed", 35, "Ann", 32);
static final Joiner COMMA_JOINER = Joiner.on(','); // because Joiner is immutable
static final SomeMutableType[] EMPTY_ARRAY = {};

// Not constants
static String nonFinal = "non-final";
final String nonStatic = "non-static";
static final Set<String> mutableCollection = new HashSet<String>();
static final ImmutableSet<SomeMutableType> mutableElements = ImmutableSet.of(mutable);
static final ImmutableMap<String, SomeMutableType> mutableValues =
    ImmutableMap.of("Ed", mutableInstance, "Ann", mutableInstance2);
static final Logger logger = Logger.getLogger(MyClass.getName());
static final String[] nonEmptyArray = {"these", "can", "change"};
```

5.2.5. `computedValues`

```

    computedValues = index

```

5.2.6. `public`

```

    public

```

5.2.7. `final`

```

    final

```

5.2.8. `E, T, X, T2`

- `E, T, X, T2`
- `RequestT, FooBarT`

9.6.3 5.3. `Miller's algorithm`

“IPv6” “iOS” “Google”

1. “Miller’s algorithm” “Muellers algorithm”
2. “AdWords” “ad words” “iOS”
3. ...
4. ...

| String                  | String            | String            |
|-------------------------|-------------------|-------------------|
| “XML HTTP request”      | XmlHttpRequest    | XMLHttpRequest    |
| “new customer ID”       | newCustomerId     | newCustomerId     |
| “inner stopwatch”       | innerStopwatch    | innerStopWatch    |
| “supports IPv6 on iOS?” | supportsIpv6OnIos | supportsIPv6OnIOS |
| “YouTube importer”      | YouTubeImporter   | YoutubeImporter * |

\*`checkNonEmpty`

Tip: `checkNonEmpty` “nonempty” “non-empty” `checkNonEmpty` `checkNonEmpty`

## 9.7 6. `checkNonEmpty`

### 9.7.1 6.1 `@Override`

`@Override` `@Deprecated` `@Override`

### 9.7.2 6.2. `AssertionError`

`AssertionError`

`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);
```

`expected`

```
try {
    emptyStack.pop();
    fail();
} catch (NoSuchElementException expected) {
}
```

### 9.7.3 6.3. `Foo`

`Foo`

```
Foo aFoo = ...;
Foo.aStaticMethod(); //
aFoo.aStaticMethod(); //
somethingThatYieldsAFoo().aStaticMethod(); //
```

### 9.7.4 6.4. Object.finalize

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. Javadoc

HTML <ul> <table> <p>

#### 7.1.3. Javadoc

@param @return @throws @deprecated

### 9.8.2 7.2. Javadoc

Javadoc A {code Foo} is a... This method returns... Save the record.

Tip: Javadoc /\*\* @return the customer ID \*/ Returns the customer ID. \*/

### 9.8.3 7.3. Javadoc

```
public public protected Javadoc  
7.3.4 Javadoc Javadoc
```

#### 7.3.1.

```
getFoo() Javadoc "Returns the foo"
```

---

Tip: `getCanonicalName()` "canonical name" `/** Returns the canonical name. */`

---

#### 7.3.2.

```
Javadoc
```

#### 7.3.4. Javadoc 7.3.3

```
Javadoc
```

```
Javadoc /**
```

```
Javadoc 7.1.1 7.1.2 7.1.3 7.2
```