

首頁 Python教學 C/C++教學 歸檔 標籤 Projects 關於我



ShengYu Talk

2021-02-26 • C/C++教學

GoogleTest 寫 C++ 單元測試的用法與教學



No-Hassle Migrations

Our White Glove Service Makes Migrations Simple and Painless. Get Started Today.

Liquid Web

[Lear](#)

本篇 ShengYu 介紹 GoogleTest 用法與教學，

以下內容將分為這幾部份，

- 下載編譯安裝
- googletest 常用語法
- googletest 基本的 C++ 單元測試
- 查詢 googletest 的版本

下載編譯安裝

以下是 GoogleTest 的下載與編譯步驟，

```
1 $ git clone https://github.com/google/googletest
2 $ cd googletest
```

googletest 原始碼目錄下有 googletest 與 googlemock，

注意要切換到 googletest/ 頂層目錄去編譯，不是切換到 googletest/googletest/ 下編譯，否則你可能會遇到 cmake 的錯誤，

在 Ubuntu 16.04 下可能會遇到編譯錯誤的訊息，需要加上 `std=c++11` 的編譯選項，一種方法是在 `CMakeLists.txt` 裡加入 `set(CMAKE_CXX_STANDARD 11)` 或 `-DCMAKE_CXX_FLAGS=-std=c++11`

另外如果電腦是多核心可以下 `make -j4` 有助於提升編譯速度，

如果要安裝到系統的話，請輸入下列指令，

```
1 sudo make install
```

你可以在 `/usr/local/include/` 目錄下發現有新的 `gtest/gtest.h` 標頭檔

```
/usr/local/include/gtest/gtest.h
```

你也可以在 `/usr/local/lib/` 目錄下發現有新的 `libgtest` 靜態函式庫

```
/usr/local/lib/libgtest.a
```

這些待會都會用到，

googletest 常用語法

這邊列出幾個 googletest 常用斷言，googletest 中定義的斷言如下，

基本斷言

判斷 condition 為真

```
ASSERT_TRUE(condition);
```

```
EXPECT_TRUE(condition);
```

判斷 condition 為假

```
ASSERT_FALSE(condition);
```

```
EXPECT_FALSE(condition);
```

數字比較

語法為 `ASSERT_EQ(期待值, 實際值)`，期待值和實際值不等時，測試失敗

判斷兩數是否相等，`expected == actual`

```
ASSERT_EQ(expected, actual);
```

```
EXPECT_EQ(expected, actual);
```

判斷兩數是否不相等 · $val1 \neq val2$

```
ASSERT_NE(val1, val2);
```

```
EXPECT_NE(val1, val2);
```

判斷 $val1$ 是否小於 $val2$ · $val1 < val2$

```
ASSERT_LT(val1, val2);
```

```
EXPECT_LT(val1, val2);
```

判斷 $val1$ 是否小於等於 $val2$ · $val1 \leq val2$

```
ASSERT_LE(val1, val2);
```

```
EXPECT_LE(val1, val2);
```

判斷 $val1$ 是否大於 $val2$ · $val1 > val2$

```
ASSERT_GT(val1, val2);
```

```
EXPECT_GT(val1, val2);
```

判斷 $val1$ 是否大於等於 $val2$ · $val1 \geq val2$

```
ASSERT_GE(val1, val2);
```

```
EXPECT_GE(val1, val2);
```

字元字串比較

判斷兩個 C-style 字元字串有相同的內容

```
ASSERT_STREQ(expected_str, actual_str);
```

```
EXPECT_STREQ(expected_str, actual_str);
```

判斷兩個 C-style 字元字串有不同的內容

```
ASSERT_STRNE(str1, str2);
```

```
EXPECT_STRNE(str1, str2);
```

判斷兩個 C-style 字元字串有相同的內容，忽略大小寫

```
ASSERT_STRCASEEQ(expected_str, actual_str);  
EXPECT_STRCASEEQ(expected_str, actual_str);
```

判斷兩個 C-style 字元字串有不同的內容，忽略大小寫

```
ASSERT_STRCASENE(str1, str2);  
EXPECT_STRCASENE(str1, str2);
```

googletest 基本的 C++ 單元測試

這邊就示範一下怎麼用 GoogleTest 來寫單元測試，假如我要測試我寫的 myadd 函式的話，

gtest-add.cpp

```
1  #include <gtest/gtest.h>  
2  
3  int myadd(int a, int b) {  
4      return a + b;  
5  }  
6  
7  TEST(testCase, test1) {  
8      EXPECT_EQ(myadd(2, 3), 5);  
9  }  
10  
11 int main(int argc, char **argv) {  
12     testing::InitGoogleTest(&argc, argv);  
13     return RUN_ALL_TESTS();  
14 }
```

接下來用 cmake 編譯或 g++ 編譯都可以，我這邊簡單的用 g++ 作示範，

```
1 g++ gtest-add.cpp -o a.out -std=c++11 -lgtest -lpthread
```

或者使用 pkg-config 來輔助，

```
1 g++ gtest-add.cpp -o a.out -std=c++11 `pkg-config --libs gtest`
```

執行 a.out 後就可以看到測試的輸出囉～

```
1 $ ./a.out
2 [=====] Running 1 test from 1 test suite.
3 [-----] Global test environment set-up.
4 [-----] 1 test from testCase
5 [ RUN      ] testCase.test1
6 [          OK ] testCase.test1 (0 ms)
7 [-----] 1 test from testCase (0 ms total)
8
9 [-----] Global test environment tear-down
10 [=====] 1 test from 1 test suite ran. (0 ms total)
11 [ PASSED  ] 1 test.
```

多加幾個測試項目，

gtest-arithmetic.cpp

```
1 // g++ gtest-arithmetic.cpp -o a.out -std=c++11 -lgtest -lpthread
2 #include <gtest/gtest.h>
3
4 int myadd(int a, int b) {
5     return a + b;
6 }
7
8 int mysub(int a, int b) {
9     return a - b;
10 }
11
12 int mymul(int a, int b) {
13     return a * b;
14 }
15
16 int mydiv(int a, int b) {
17     return a / b;
18 }
19
20 TEST(testCase, test1) {
21     EXPECT_EQ(myadd(2, 3), 5);
22 }
23
24 TEST(testCase, test2) {
25     EXPECT_EQ(mysub(2, 3), -1);
26 }
27
```



```
28 TEST(testCase, test3) {
29     EXPECT_EQ(mymul(3, 4), 12);
30 }
31
32 TEST(testCase, test4) {
33     EXPECT_EQ(mydiv(21, 7), 3);
34 }
35
36 int main(int argc, char **argv) {
37     testing::InitGoogleTest(&argc, argv);
38     return RUN_ALL_TESTS();
39 }
```

輸出結果如下，

```
1  $ ./a.out
2  [=====] Running 4 tests from 1 test suite.
3  [-----] Global test environment set-up.
4  [-----] 4 tests from testCase
5  [ RUN      ] testCase.test1
6  [          OK ] testCase.test1 (0 ms)
7  [ RUN      ] testCase.test2
8  [          OK ] testCase.test2 (0 ms)
9  [ RUN      ] testCase.test3
10 [          OK ] testCase.test3 (0 ms)
11 [ RUN      ] testCase.test4
12 [          OK ] testCase.test4 (0 ms)
```

```
13  [-----] 4 tests from testCase (0 ms total)
14
15  [-----] Global test environment tear-down
16  [=====] 4 tests from 1 test suite ran. (0 ms total)
17  [  PASSED  ] 4 tests.
```

查詢 googletest 的版本

googletest 版本號碼在 CMakeLists.txt 裡會去設定的 GOOGLETEST_VERSION 變數，

```
1  set(GOOGLETEST_VERSION 1.10.0)
```

其他參考

CXX_STANDARD — CMake 3.20.0-rc2 Documentation

https://cmake.org/cmake/help/latest/prop_tgt/CXX_STANDARD.html

C++ 单元测试框架-gtest | Mike's Blog

<https://mikeblog.top/2019/01/02/googletest/>

快速上手Google C++ 測試框架googletest_記錄學習的過程-CSDN博客

<https://blog.csdn.net/weiwei9363/article/details/103469525>

c++单元测试之gtest测试框架快速上手_guotianqing的博客-CSDN博客

<https://blog.csdn.net/guotianqing/article/details/104055221>

C Unit Test Framework 介紹 (Googletest) | by 亮谷 | Medium

<https://medium.com/@ktvexe/c-unit-test-framework-%E4%BB%8B%E7%B4%B9-googletest-9713dadceb7a>

輕鬆編寫 C++ 單元測試 | 開源互助社區

<https://coctec.com/docs/linux/show-post-68880.html>

其它相關文章推薦

[C/C++ 新手入門教學懶人包](#)

[C/C++ 3種字串轉數字的方法與範例](#)

[C++ virtual 的兩種用法](#)

[C/C++ 字串反轉 reverse](#)

[C/C++ call by value傳值, call by pointer傳址, call by reference傳參考 的差別](#)

[C++ 類別樣板 class template](#)

[std::sort 用法與範例](#)

[std::find 用法與範例](#)

[std::queue 用法與範例](#)

[std::map 用法與範例](#)

[std::deque 用法與範例](#)

[std::vector 用法與範例](#)

#C/C++ #Google

[Share](#)



The Industry's Best Support

Need Help At 2AM? We've Got Your Back With Around-The-Monitoring. Switch Today.

Liquid Web

[Learn](#)

NEWER

[Ubuntu cmake 編譯安裝](#)

OLDER

[Android adb 同步時間/設定時間](#)

精選文章

[Python 新手入門教學懶人包](#)

[C/C++ 新手入門教學懶人包](#)

[Linux 常用指令教學懶人包](#)

最新文章

[Python hex string to int 16進位字串轉數字](#)


[C/C++ 二元搜尋法 Binary Search](#)

[Python random.randint 產生隨機數字](#)

[C/C++ 循序搜尋法 Sequential Search](#)

[Python import 用法與範例](#)

粉絲專頁



ShengYu

1 人說這讚

[說這專頁讚](#)[分享](#)

ShengYu

shengyu7697

[Edit profile](#)

10 followers · 30 following · 135 stars

Taiwan
shengyu7697@gmail.com
<https://shengyu7697.github.io/>

TinyTcp

This is a C++ socket wrapper designed to fast development of network related functionality.

C++ ☆ 3


YahooStock

Get stock information from yahoo tw website

Python ☆ 1

Achievements


2 405 contributions in the last year



You have a new badge!


You contributed code to repositories used in the Mars 2020 Helicopter Mission and now have a badge for it. Thank you!

[OK, got it](#) [Opt out in settings.](#)



Contribution activity

April 2021



ShengYu

約 3 週前

分類

[Arduino](#) (3)

[C/C++教學](#) (81)

[Github](#) (12)

[Git教學](#) (19)

[OpenCV教學](#) (4)

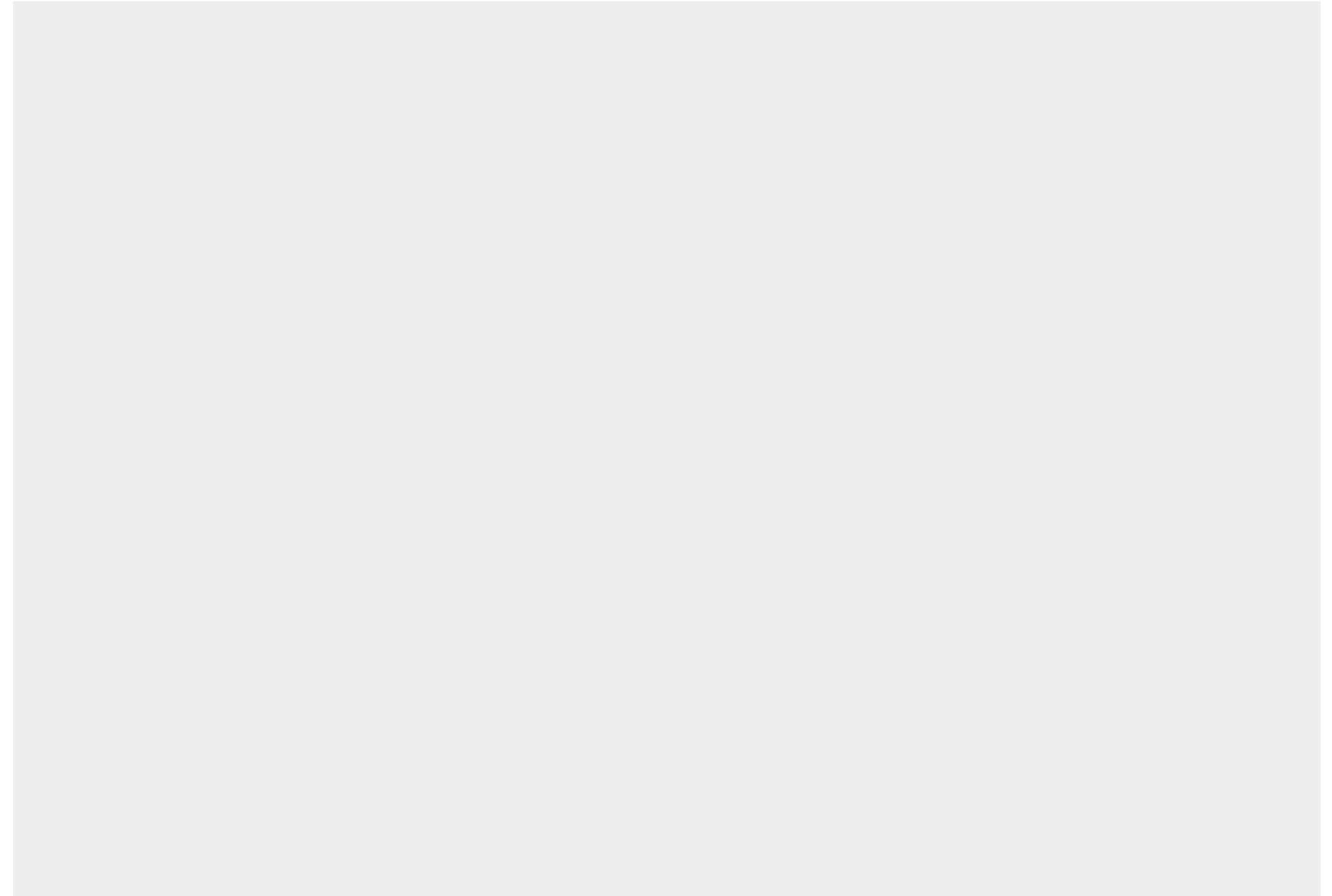
[Projects](#) (6)

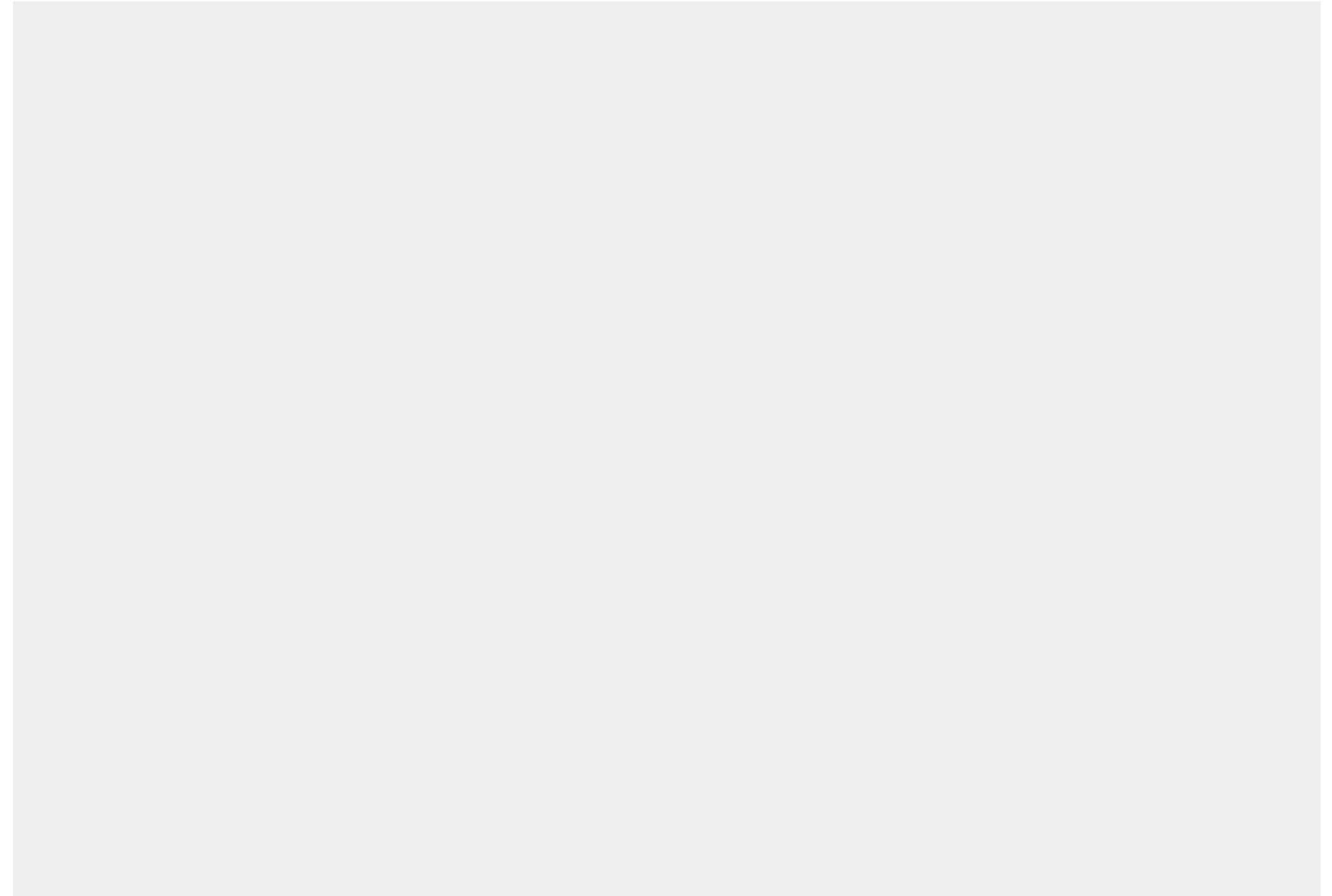
[Python教學](#) (167)

[Qt教學](#) (6)



OZIO-潤白凝露





© 2021 ShengYu
Powered by [Hexo](#)