廖雪峰

<https://www.liaoxuefeng.com/wiki/0014316089557264a6b348958f449949df42a6d3a2e542c000>

■书籍推荐

<https://www.cnblogs.com/ipaomi/p/7715674.html>

■环境搭建

Anaconda下载地址

<https://www.anaconda.com/download/#download>

安装时注意勾选添加环境变量

安装完命令行打python

退出是exit()

Anaconda提供了显示高亮颜色的命令行，输入ipython即可

■编辑器和IDE

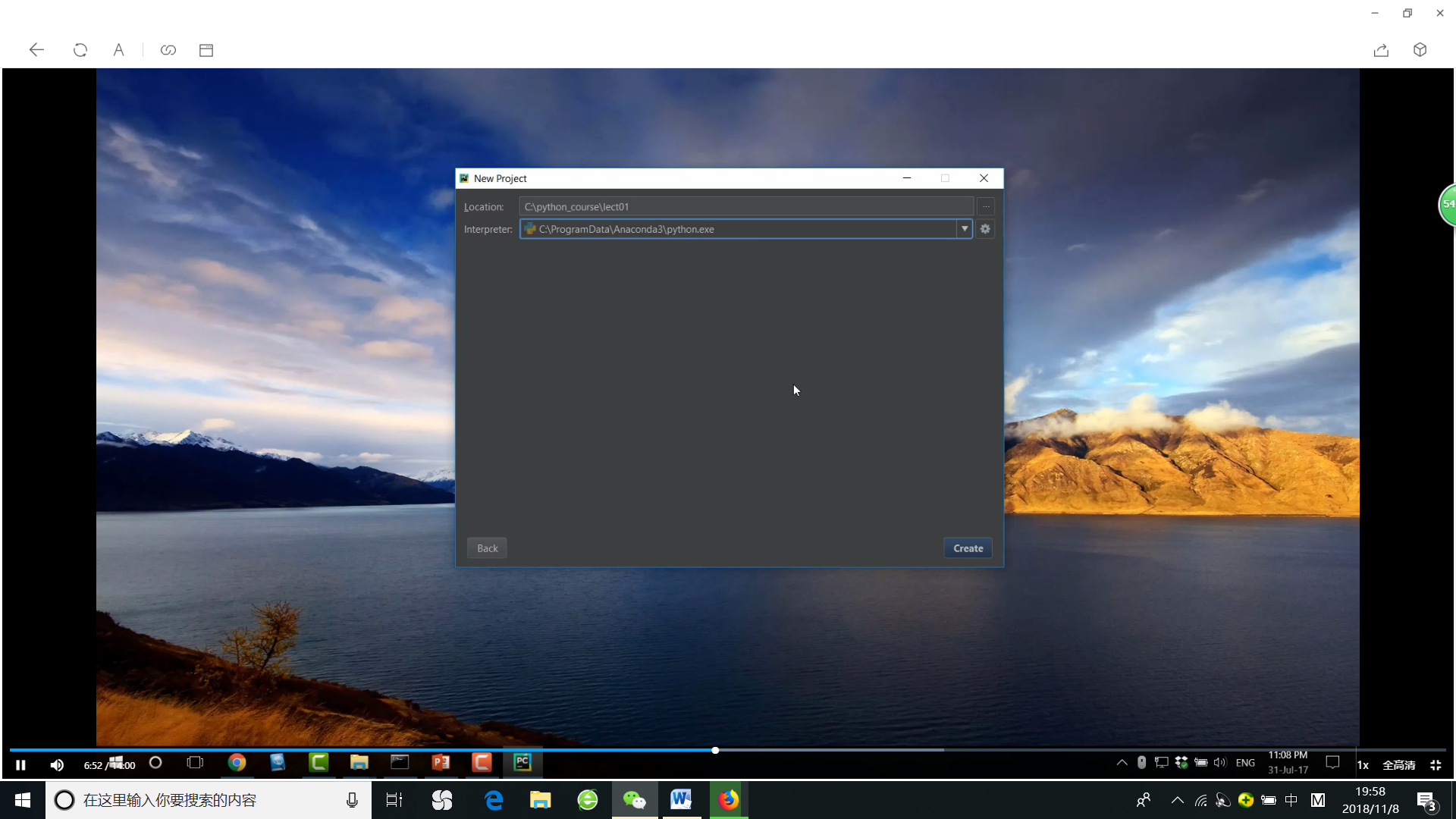
使用anaconda自带的编辑器idle

在命令行输入idle即可

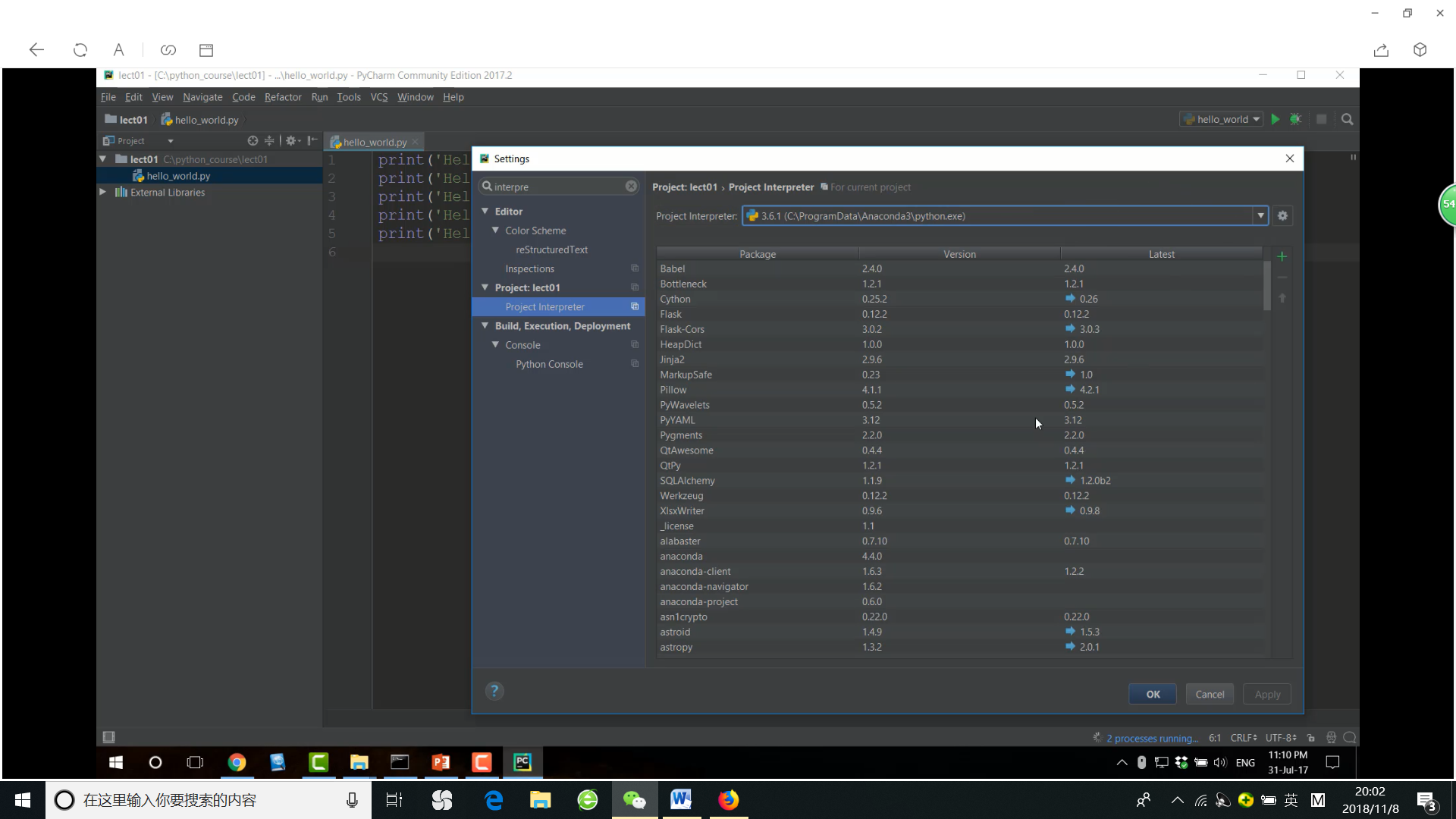
大型程序可以使用pyCharm

<http://www.jetbrains.com/pycharm/>

安装pyCharm后需要配置anaconda解释器



之后在setting里也可以配置



■占位符pass

Pass表示先占位，之后再完善其内容。这样系统不会报错

例如

a = 1  
if a == 1:  
 pass  
else:  
 pass

■输入输出

输入

input()

输出

print()

■注释

快捷键 ctrl + /

单行在开头加 #

多行可以用”””

"""  
这里是多行注释，一般用在文档的开头  
表明一些版本信息等，如：  
测试程序v1.1  
 v1.0更新内容为...  
"""  
  
# 这里是单行注释

■变量和常量

●变量命名推荐

变量名必须是大小写英文、数字和\_的组合，且不能用数字开头！！！

Python对大小写敏感，所以可以使用驼峰

有意义的名字，数据类型，变量标记。中间用下划线隔开

moneyRmb\_str\_val = '100'  
moneyRmb\_num\_val = 100

python是动态语言，也就是弱类型语言，类似javascript

java是静态语言，也就是强类型语言

●常量

通常用全大写字符代表常量。Python没有任何机制保证常量的不变性，因此用全大写命名常量只是提醒程序员这是常量，不要改变它

■缩进

Python缩进只能是4个空格，或者tab 退缩进是shift + tab

必须严格缩进，因为python的缩进用来代表块

■计算

除法

# 用/是精确除法  
print(10 / 3)  
# 输出：3.3333333333333335  
  
# 用//是取整，也叫地板除  
print(10 // 3)  
# 输出： 3

# %取余  
print(9 % 2)  
# 输出：1

幂

# 乘方用\*\*表示  
  
# 2的3次方  
print(2 \*\* 3)  
# 输出： 8  
  
# 5的平方  
print(5 \*\* 2)  
# 输出： 25

●字符串也能用乘法

print('你好' \* 10)  
# 输出：你好你好你好你好你好你好你好你好你好你好

●优先级

\*\* > \*/%// > +-

绝对值

# abs()取绝对值  
print(abs(-10))  
# 输出：10  
print(abs(11))  
# 输出：11

最大值

# max()取最大值  
print(max(1, 3, -10, 125))  
# 输出： 125

■数据类型转换

# int()取整数  
print(int(3.14))  
# 输出： 3  
  
# str()取字符串  
print(str(25))  
# 输出： 25  
  
# float()取浮点数  
print(float(3))  
# 输出：3.0  
  
# type()输出数据类型  
print(type(3))  
# 输出： <class 'int'>

# bool()转换成布尔  
print(bool(1))  
# 输出： True  
print(bool(''))  
# 输出： False

Isinstance()

# isinstance()用来判断数据是否是某个类型  
print(isinstance(1, str))  
# 输出：False  
  
print(isinstance('我', str))  
# 输出：True  
  
print(isinstance(1.5, int))  
# 输出：False  
  
# 只要括号内有一个类型符合，就返回true  
print(isinstance(1.5, (int, float)))  
# 输出: True

■

# 执行程序时，数据被复制到内存中，每个数据都有自己在内存的位置。  
# 变量引用的是数据在内存中位置的地址。  
# 使用id可以查看变量引用的地址  
  
  
a = 1  
# 使用id()查看a所引用的数据1在内存中位置的地址  
print(id(a))  
# 输出：140722382565120  
  
# 使用id()查看数据1在内存中位置的地址,和上面a引用的地址一致  
print(id(1))  
# 输出：140722382565120  
  
b = a  
print(id(b))  
# 输出：140722382565120

# 调用函数时，传递给函数的实参，调用的实际是实参对数据的引用  
# 函数返回的也是对数据地址的引用  
b = 10  
  
  
def test(num):  
 print('b被传到函数内部成为实参%s，它的内存地址是：%s' % (num, id(num)))  
 c = 'hello'  
 print('函数要返回的数据的内存地址是%s' % id(c))  
 return c  
  
  
print('b的内存地址是：%s' % id(b))  
  
test(b)  
print('外部接收到的函数返回数据的内存地址是%s' % id(test(b)))  
  
# 输出结果：  
# b的内存地址是：140722373390368  
# b被传到函数内部成为实参10，它的内存地址是：140722373390368  
# 函数要返回的数据的内存地址是1546483078008  
# 外部接收到的函数返回数据的内存地址是1546483078008

■

可变和不可变类型  
#  
# 不可变类型：内存中的数据不允许被修改  
# 数字类型 int bool float complex long(2, x)  
# 字符串 str  
# 元组 tuple  
#  
# 可变类型：内存中的数据可以被修改  
# 列表 list  
# 字典 dict

■全局变量与局部变量

# 全局变量是在函数外部定义的变量（没有定义在某一个函数内），所有函数内部都可以使用这个变量。  
# 函数内部只能使用全局变量，但是不能修改全局变量的值  
  
# 局部变量是在函数内部定义的变量，只能在函数内部使用。  
  
# 局部变量的生命周期：  
# 当执行到函数内部定义变量的代码后，该局部变量生成。  
# 函数执行完毕后，其局部变量一同被系统回收。

●全局变量的命名，建议在前面加上一个g\_或者gl\_前缀，以免和局部变量重名

例如

g\_val

gl\_vae

Python规定函数内部不能修改全局变量的规则，是外来避免全局变量滥用造成的混乱

a = 1  
  
  
def b():  
 # 函数内部无法修改全局变量a，  
 # 如果以下写法，代表在函数内部新建一个局部变量a，和外部的全局变量a没有关系  
 a = 2  
 print(a)  
  
  
b()  
# 在外部输出a，结果仍然是1  
print(a)  
  
# 输出结果：  
# 2  
# 1

如果非得要在函数内部修改全局变量，可以用global来声明

a = 1  
  
  
def b():  
 # 函数内部无法修改全局变量a，  
 # 如果业务需要，非得要修改全局变量，可以用global来声明  
 global a  
 a = 2  
 print(a)  
  
  
b()  
# 在外部输出a，也被函数内部global声明的a修改了  
print(a)  
  
# 输出结果：  
# 2  
# 2

■代码结构示意图

|  |
| --- |
| Shebang（水半，标识用哪个解释器来解释当前文件） |
| Import模块 |
| 全局变量 |
| 函数定义 |
| 执行代码 |