

python语言入门与实践

作业课1

主讲人：吴陈炜 博士

复旦大学智能机器人研究院 工程博士

北京大学信息科学与技术学院 理学硕士

浙江大学信息与电子工程学系 工学学士

网易（杭州）网络有限公司 项目经理

美国项目管理协会 PMP（项目管理专业人士）

之江实验室 科研主管

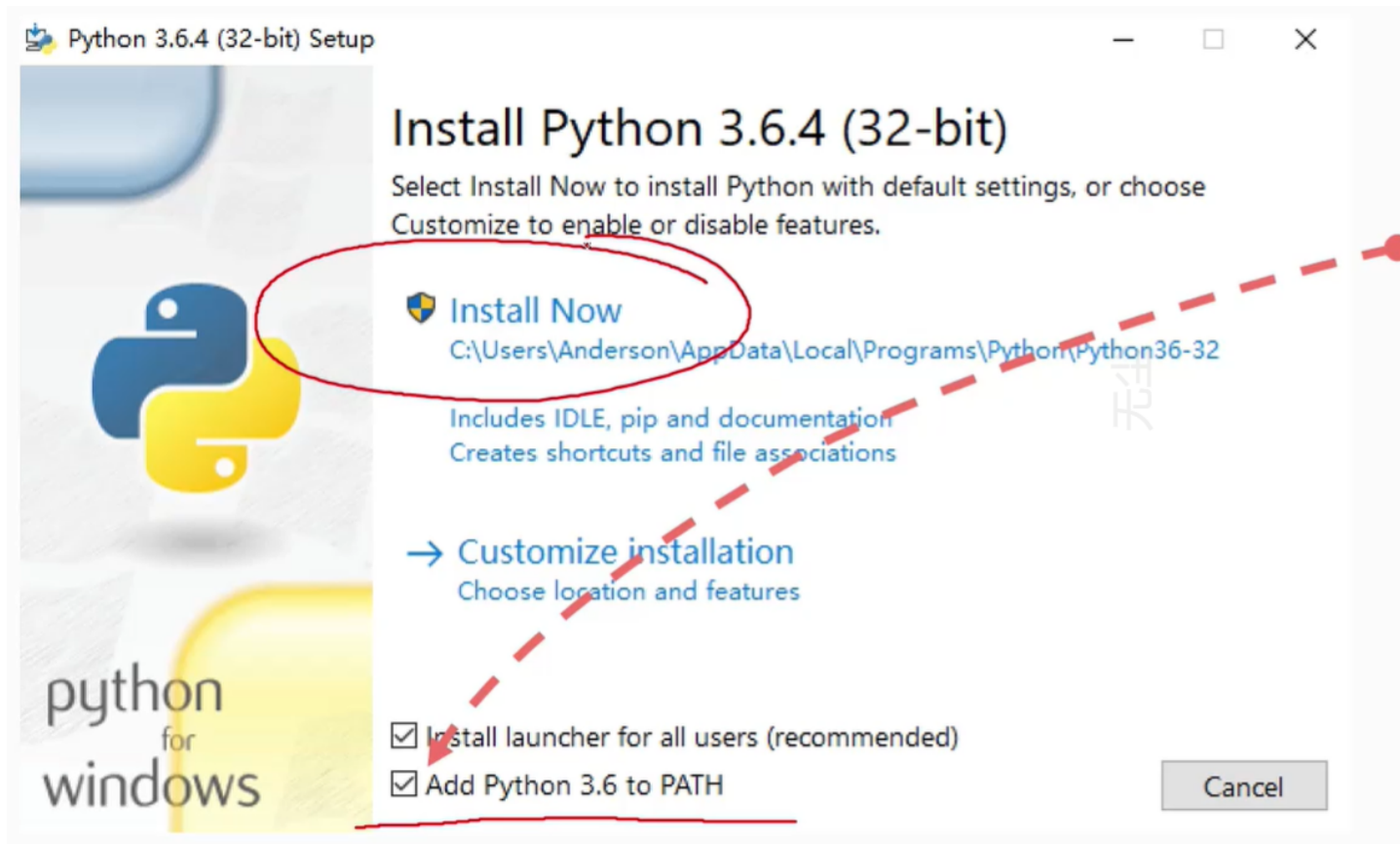
2020 年 4 月 30 日

上节课回顾

Python的应用

- 机器学习
- 网络爬虫
- 数据分析
- 数值计算
- 自动办公
- 硬件编程
- 量化交易
- 游戏开发
- 网站搭建
- 黑客攻防

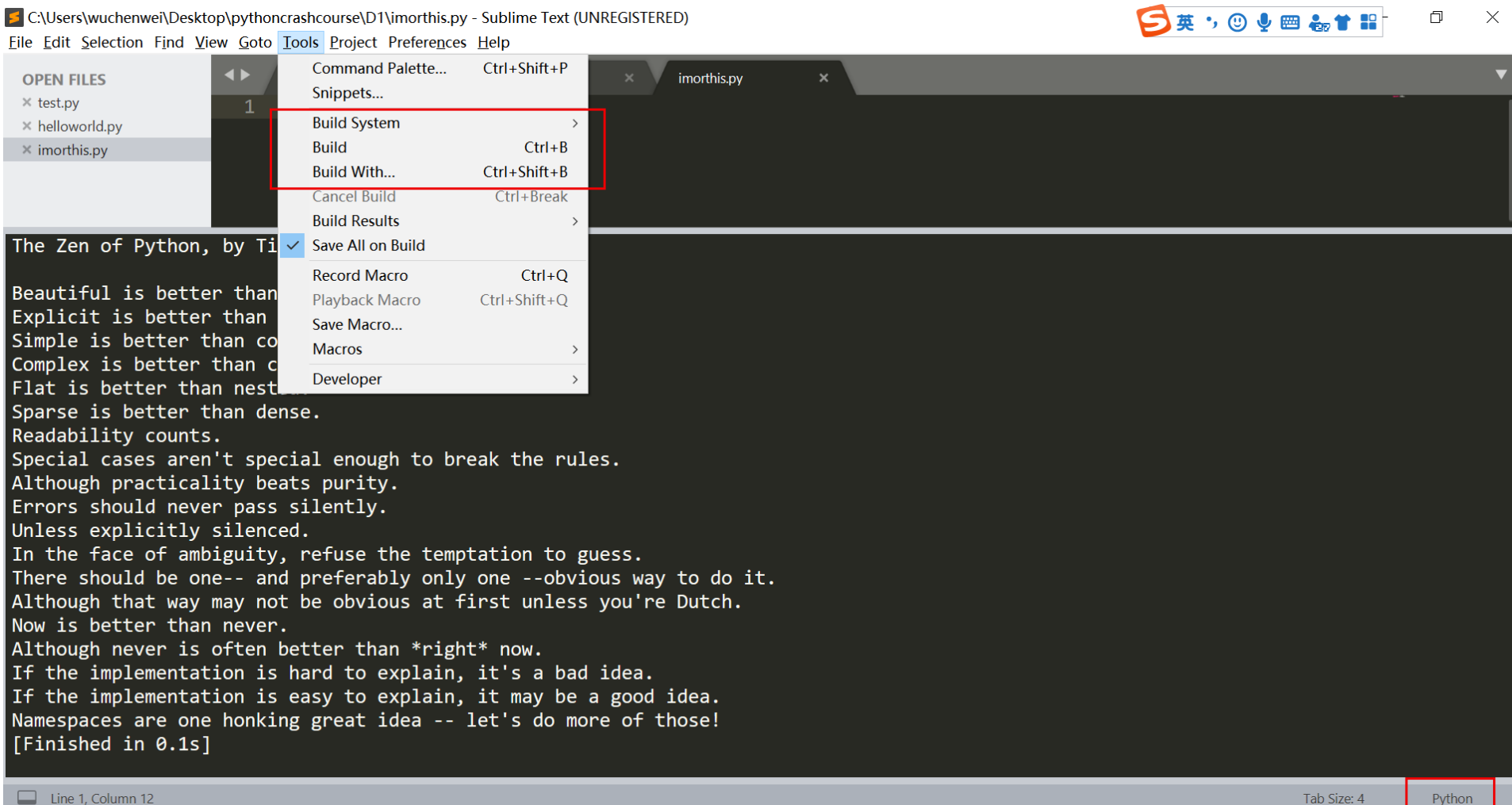
安装python



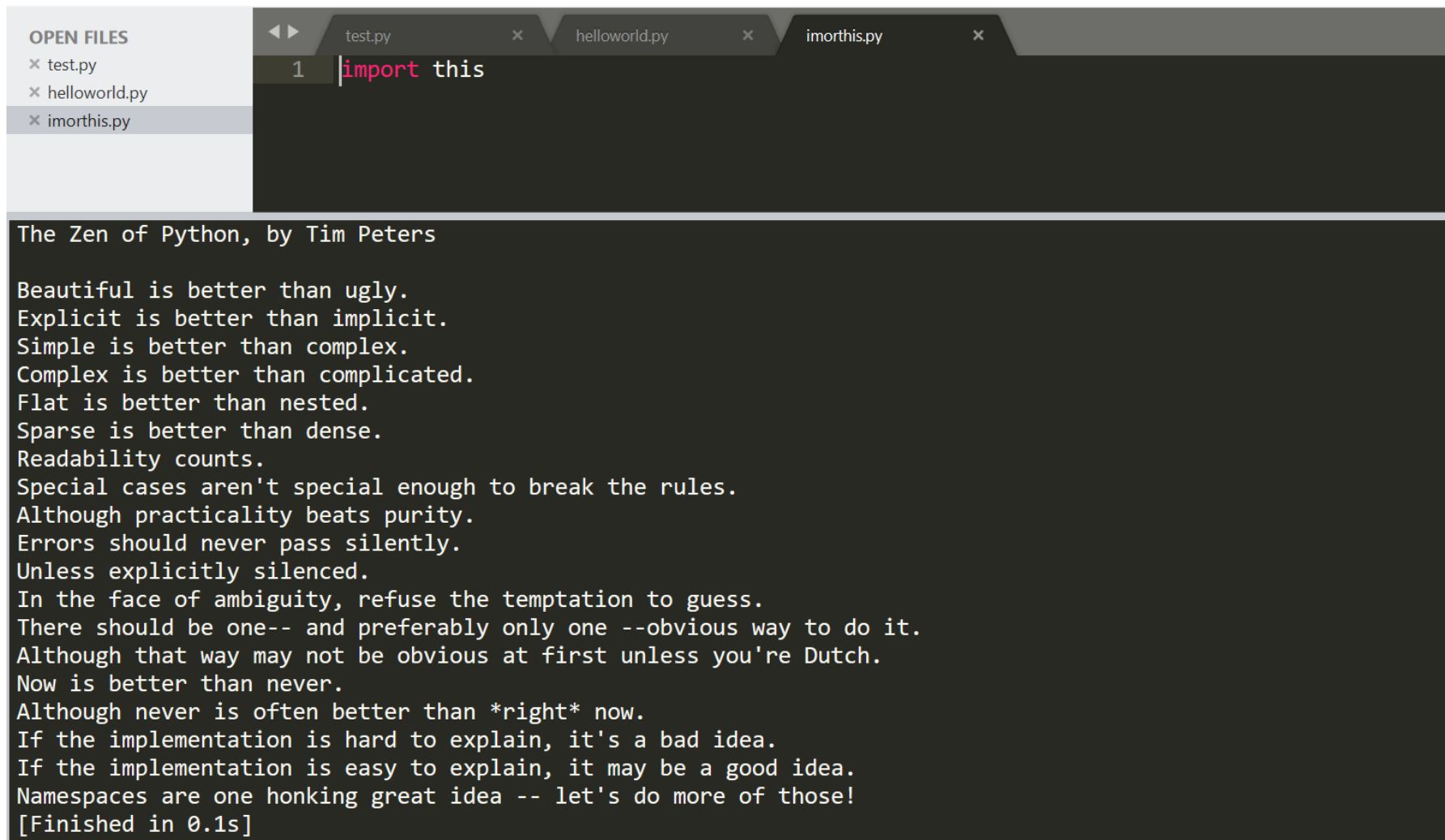
若干注意要点:

1. 推荐根据提供的链接安装python3.6.4版本;
2. 默认安装pip, 非常有用;
3. 在安装过程中注意务必勾选Add Python 3.6 to PATH选项;
4. 根据不同的操作系统选择安装方式, 参考 (python for kid) 。

安装Sublime



1行代码打印python之禅



The screenshot shows a Python IDE with three open files: test.py, helloworld.py, and imorthis.py. The active file, test.py, contains a single line of code: `import this`. Below the code editor, the output of the program is displayed, showing the Zen of Python by Tim Peters.

```
OPEN FILES
x test.py
x helloworld.py
x imorthis.py

test.py
1 import this

The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
[Finished in 0.1s]
```

今日目标：使用python程序批量下载MV



周杰伦——出道以来所有单曲MV大盘点 含HD单曲【合

294.3万 2019-07-15

吸血鬼的智齿

```
site: Bilibili
title: 【经典】周杰伦MV 【185P】 (P1. 可爱女人【1st JAY】)
stream:
  - format: flv
    container: flv
    quality: 高清 1080P
    size: 17.4 MiB (18284207 bytes)
    # download-with: you-get --format=flv [URL]
```

Downloading 【经典】周杰伦MV 【185P】 (P1. 可爱女人【1st JAY】).flv ...

0.0%	(0.0/ 17.4MB)		[1/1]	
1.4%	(0.2/ 17.4MB)		[1/1]	1 MB/s
2.9%	(0.5/ 17.4MB)		[1/1]	5 MB/s
4.3%	(0.8/ 17.4MB)		[1/1]	661 kB/s
5.7%	(1.0/ 17.4MB)		[1/1]	198 kB/s
7.2%	(1.2/ 17.4MB)		[1/1]	213 kB/s
8.6%	(1.5/ 17.4MB)		[1/1]	242 kB/s
10.0%	(1.8/ 17.4MB)		[1/1]	238 kB/s
11.5%	(2.0/ 17.4MB)		[1/1]	249 kB/s

涉及知识点

- 操作系统基础知识
- 使用import关键字调用python模块
- pip指令的使用
- 变量和字符串的使用

操作系统命令行的使用

操作系统

- 操作系统(Operating System, 简称OS)是管理计算机硬件与软件资源的计算机程序。操作系统需要处理如管理与配置内存、决定系统资源供需的优先次序、控制输入设备与输出设备、操作网络与管理文件系统等基本事务。操作系统也提供一个让用户与系统交互的操作界面。



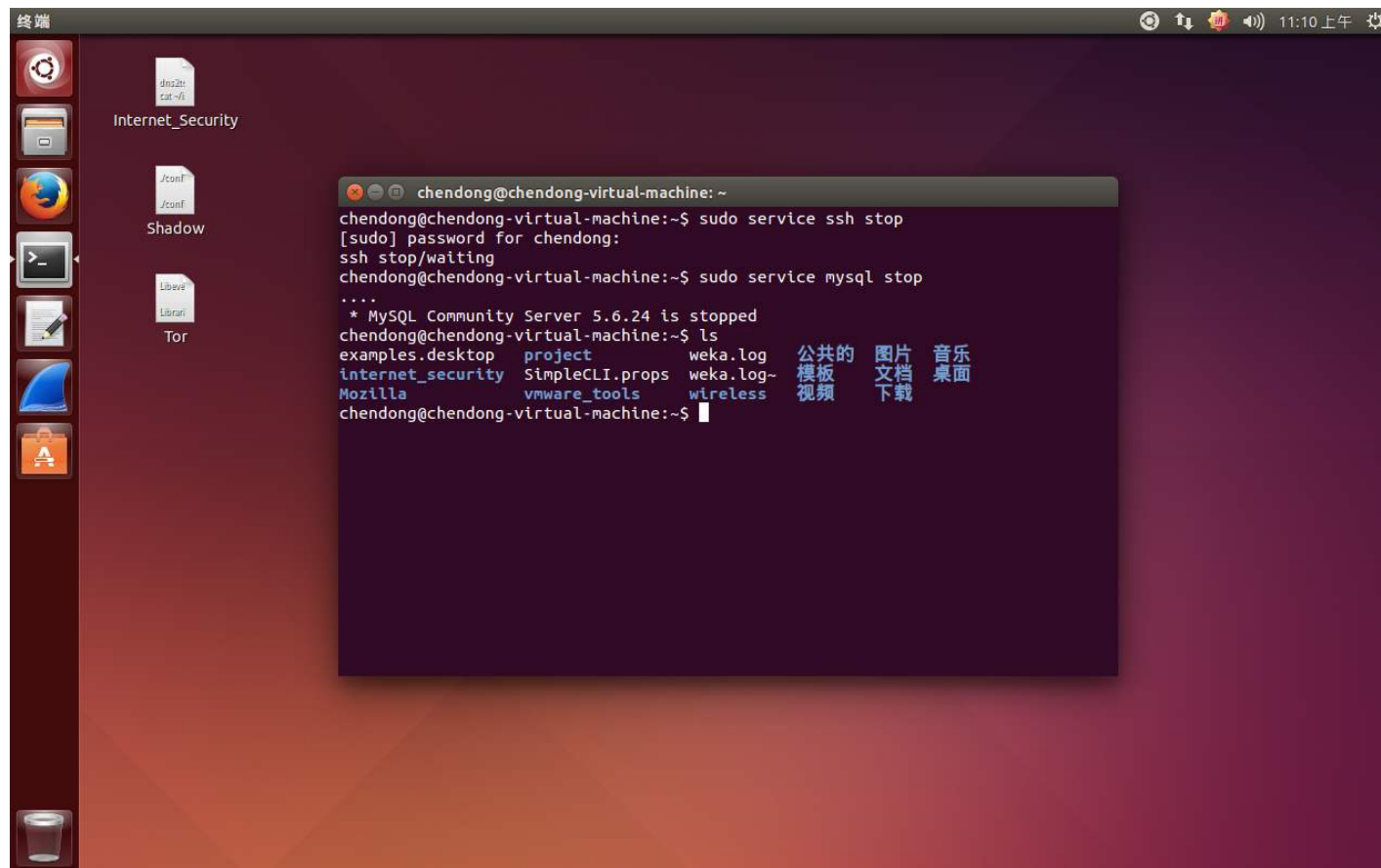
图形界面v.s.命令行界面

Windows:

cmd

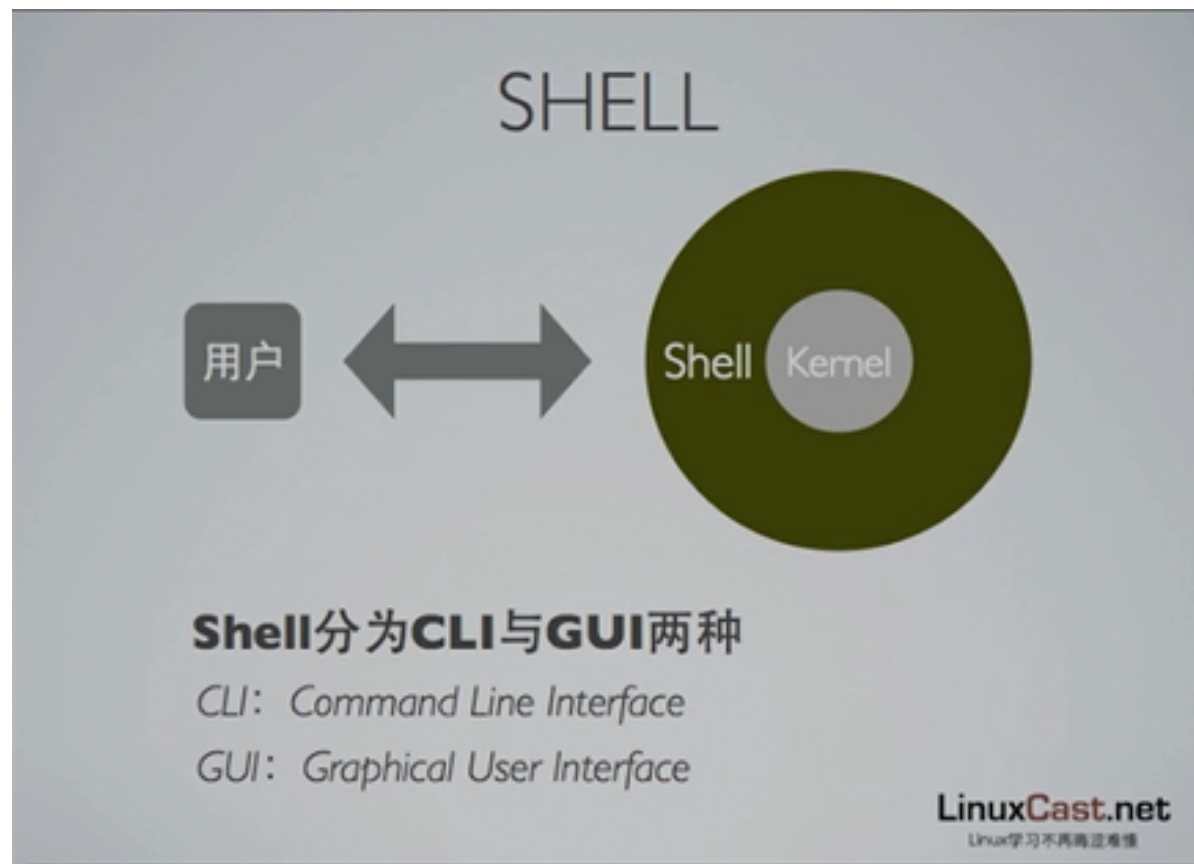
MacOS:

Terminal



Shell v.s Kernel

在计算机科学中，Shell俗称壳（用来区别于核），是指“为用户提供操作界面”的软件（命令解析器）。它类似于DOS下的command.com和后来的cmd.exe。它接收用户命令，然后调用相应的应用程序。



在命令行中运行python解释器

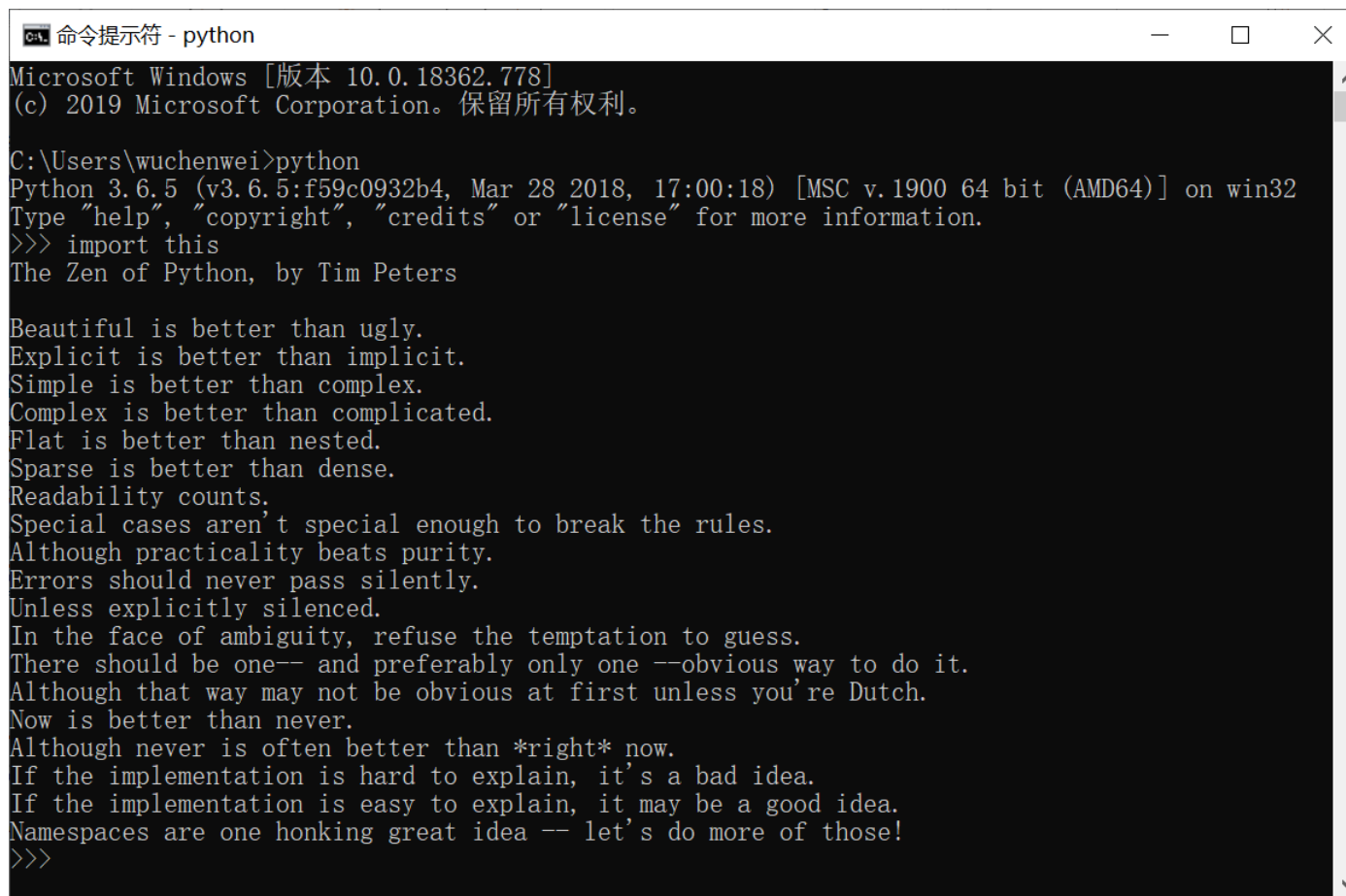
我们可以在命令行界面中使用指令来启动python解释器运行一些python程序。

Windows:

`python`

MacOS:

`python3`



```
命令提示符 - python
Microsoft Windows [版本 10.0.18362.778]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\wuchenwei>python
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

在命令行中运行python程序（模块）

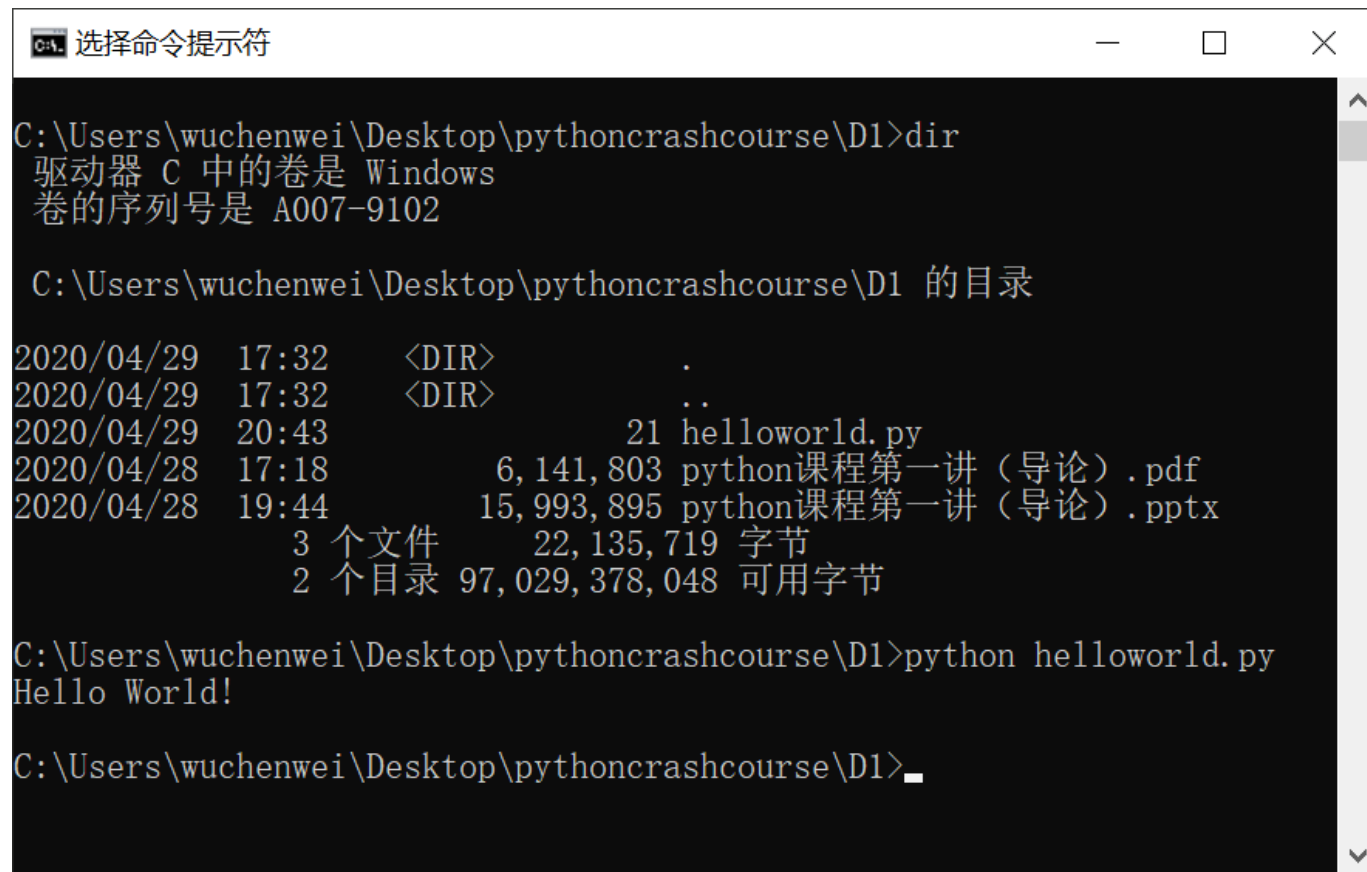
我们也可以使用指令
在命令行中运行.py文件
(python模块)

Windows:

```
python hello.py
```

MacOS:

```
python3 hello.py
```



```
选择命令提示符

C:\Users\wuchenwei\Desktop\pythoncrashcourse\D1>dir
驱动器 C 中的卷是 Windows
卷的序列号是 A007-9102

C:\Users\wuchenwei\Desktop\pythoncrashcourse\D1 的目录

2020/04/29  17:32    <DIR>          .
2020/04/29  17:32    <DIR>          ..
2020/04/29  20:43                21 helloworld.py
2020/04/28  17:18           6,141,803 python课程第一讲（导论）.pdf
2020/04/28  19:44       15,993,895 python课程第一讲（导论）.pptx
                    3 个文件      22,135,719 字节
                    2 个目录  97,029,378,048 可用字节

C:\Users\wuchenwei\Desktop\pythoncrashcourse\D1>python helloworld.py
Hello World!

C:\Users\wuchenwei\Desktop\pythoncrashcourse\D1>_
```

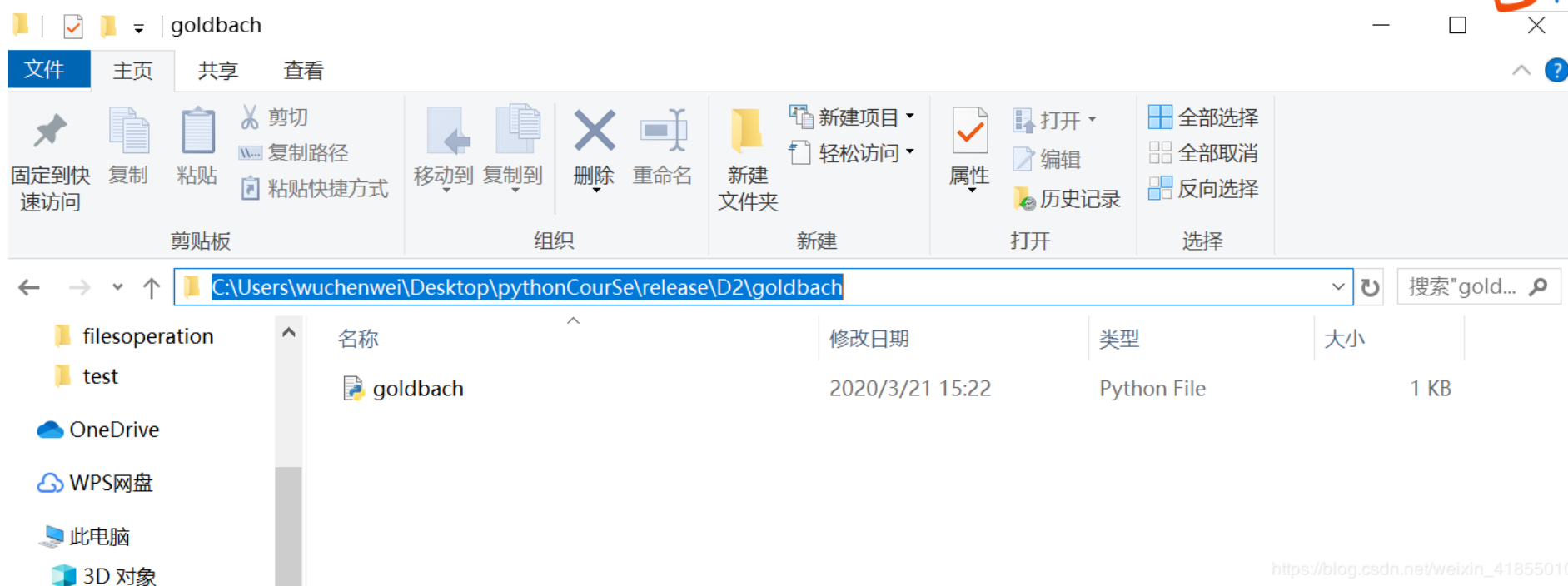
注意命令行和解释器界面的区别

- 在命令行中使用shell指令（操作系统的语言）
- 在解释器中使用python命令（python语言）
- 注意python解释器界面会出现>>>符号

文件的路径 (path)

绝对路径

- 以windows为例，我们可以在资源管理器的输入框中方便的找到绝对路径，如下图：



相对路径

- 我们可以使用点号 (.) 来代表参考所在文件的绝对路径（依旧是当前路径或者叫相对路径），这样就省去了冗长的字符串，使得代码看起来更加清晰。比如以下代码就表示目标的路径是文件所在目录下的jaychou子目录：

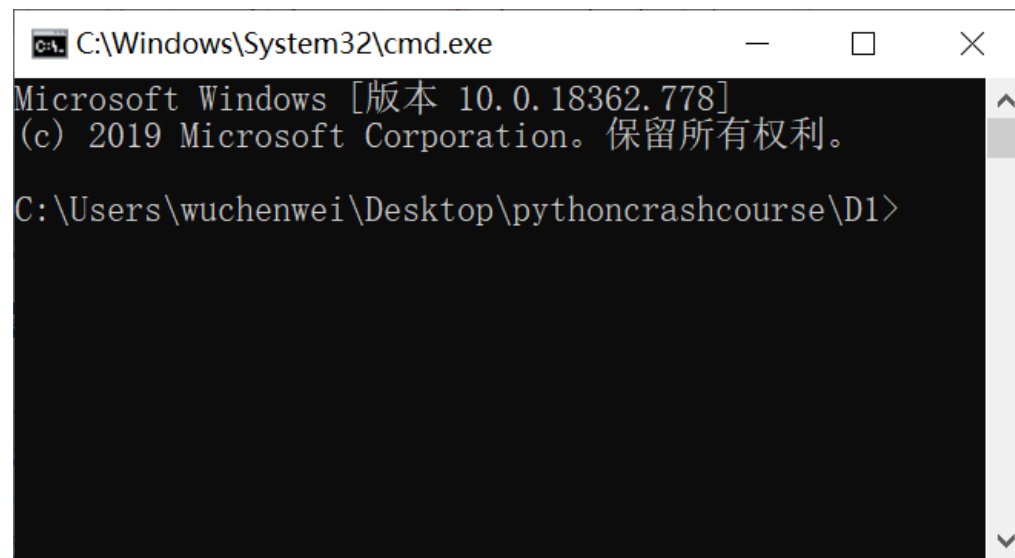
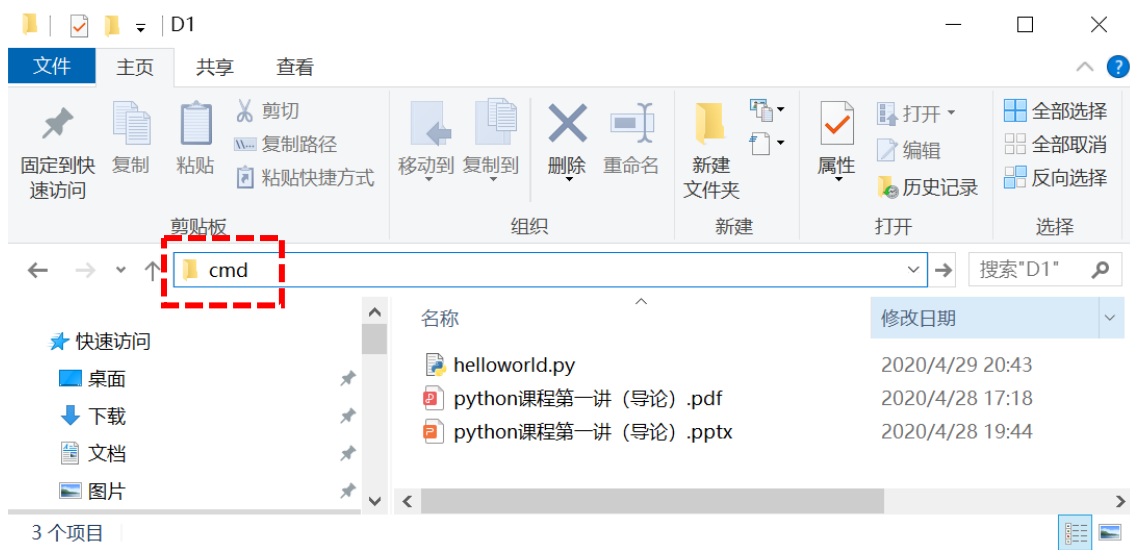
```
1 | tar_path = './jaychou'
```

- 我们还可以使用..来表示文件所在当前目录的上一层目录：

```
C:\Users\wuchenwei\Desktop\pythoncrashcourse\D1> cd ..  
C:\Users\wuchenwei\Desktop\pythoncrashcourse>_
```

小技巧：快速进入当前路径下的cmd

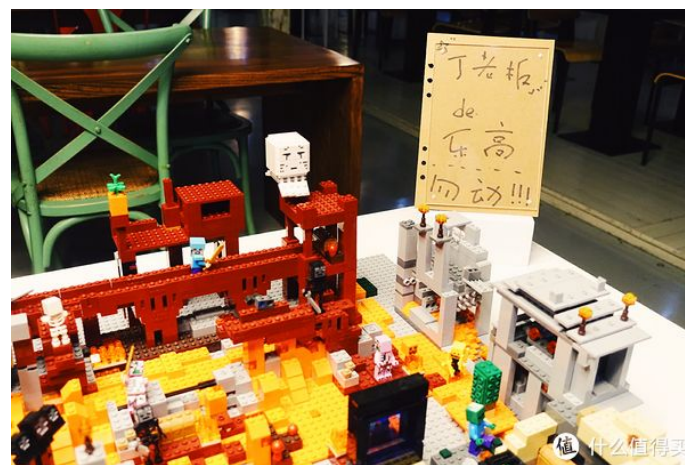
- 在资源管理器界面的快速访问栏中输入cmd命令，就可以直接在cmd中进入到当前的目录下。



import关键词

import关键词的作用:调用模块

- 模块: Python程序文件 (.py文件)
- 我们一般会把大工程拆分成若干小模块进行管理
- 我们可以自己编写, 也可以调用别人写好的模块
- 几乎每个程序都会使用import导入模块
- 像搭乐高积木一样来编写程序
- 面向对象编程 (OOP), python处处皆对象
- 提高编程效率, 不要重复造轮子



import的三种主要方法

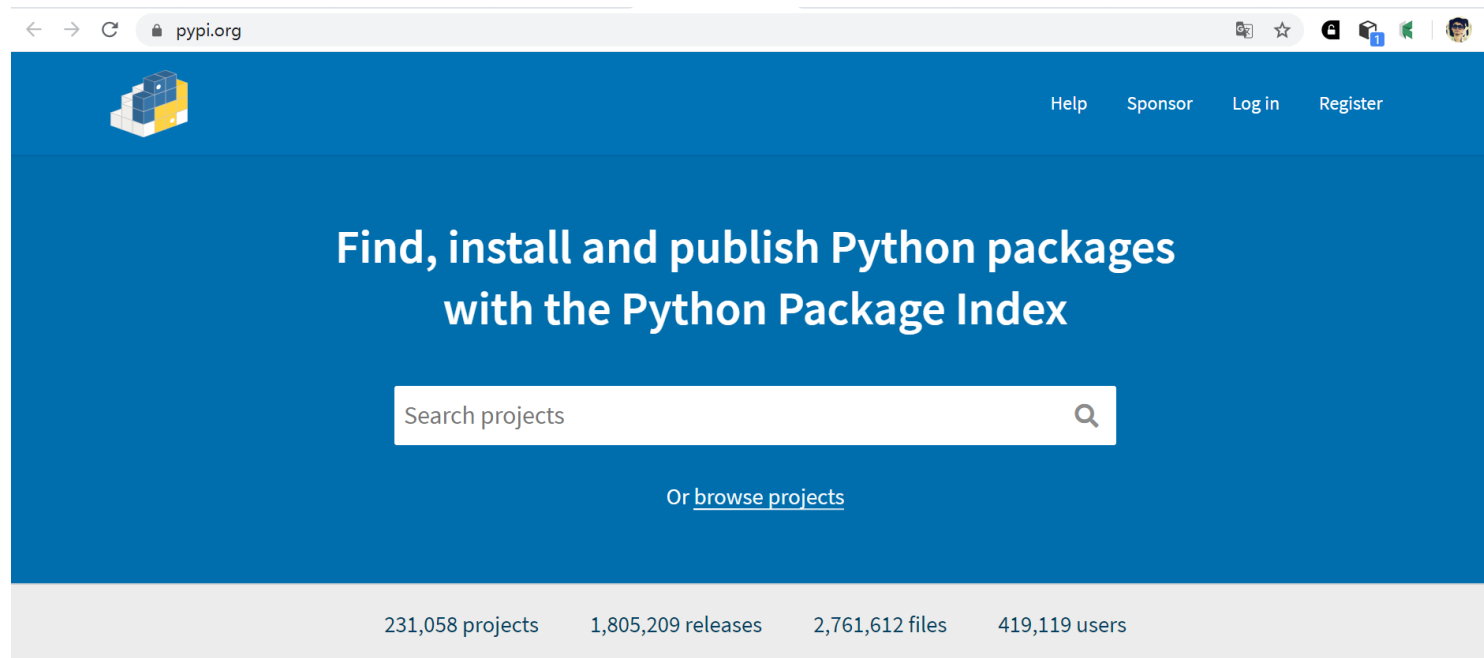
- 调用自己写的模块
- 调用系统
- 调用第三方模块

```
helloworld.py x test.py x
1 import helloworld
2 import pinyin
3 import time
4 seconds = time.time()
5 local_time = time.localtime(seconds)
6 str_time = time.asctime(local_time)
7 print(str_time)
8 pynome = pinyin.get('名字', delimiter=' ')
9 print(pynome)
```

使用pip下载第三方模块

pip简介

- pip is the package installer for Python. You can use pip to install packages from the Python Package Index and other indexes.



pip的安装

- python在安装的时候默认将pip同时安装
- 如果没有安装成功，可以在官网文档上找到解决方案

Installing with get-pip.py

To install pip, securely [1] download `get-pip.py` by following this link: [get-pip.py](https://bootstrap.pypa.io/get-pip.py). Alternatively, use `curl`:

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
```

Then run the following command in the folder where you have downloaded `get-pip.py`:

```
python get-pip.py
```

<https://pip.pypa.io/en/stable/installing/>

使用pip

- `pip list` 查看已经安装的第三方模块及版本。
- `pip install 模块名` 从官方资源网站 <https://pypi.org> 下载安装指定名称的第三方模块。
- `pip install -i 镜像站点名 模块名` 从指定的镜像站点下载安装指定名称的第三方模块。国内常用镜像站点见下页。
- `pip uninstall 模块名` 卸载已经安装的指定模块。
- `python -m pip install -U pip` （在Windows下）或 `pip install -U pip` （在Linux/Mac下）对pip工具本身进行升级，保持pip最新版本以便顺利下载最新资源。

目前国内用户最常使用的 pip 镜像下载网站包括：

- 阿里云 <http://mirrors.aliyun.com/pypi/simple/>
- 中国科技大学 <https://pypi.mirrors.ustc.edu.cn/simple/>
- 豆瓣(douban) <http://pypi.douban.com/simple/>
- 清华大学 <https://pypi.tuna.tsinghua.edu.cn/simple/>
- 华中理工大学 <http://pypi.hustunique.com/>
- 山东理工大学 <http://pypi.sdutlinux.org/>

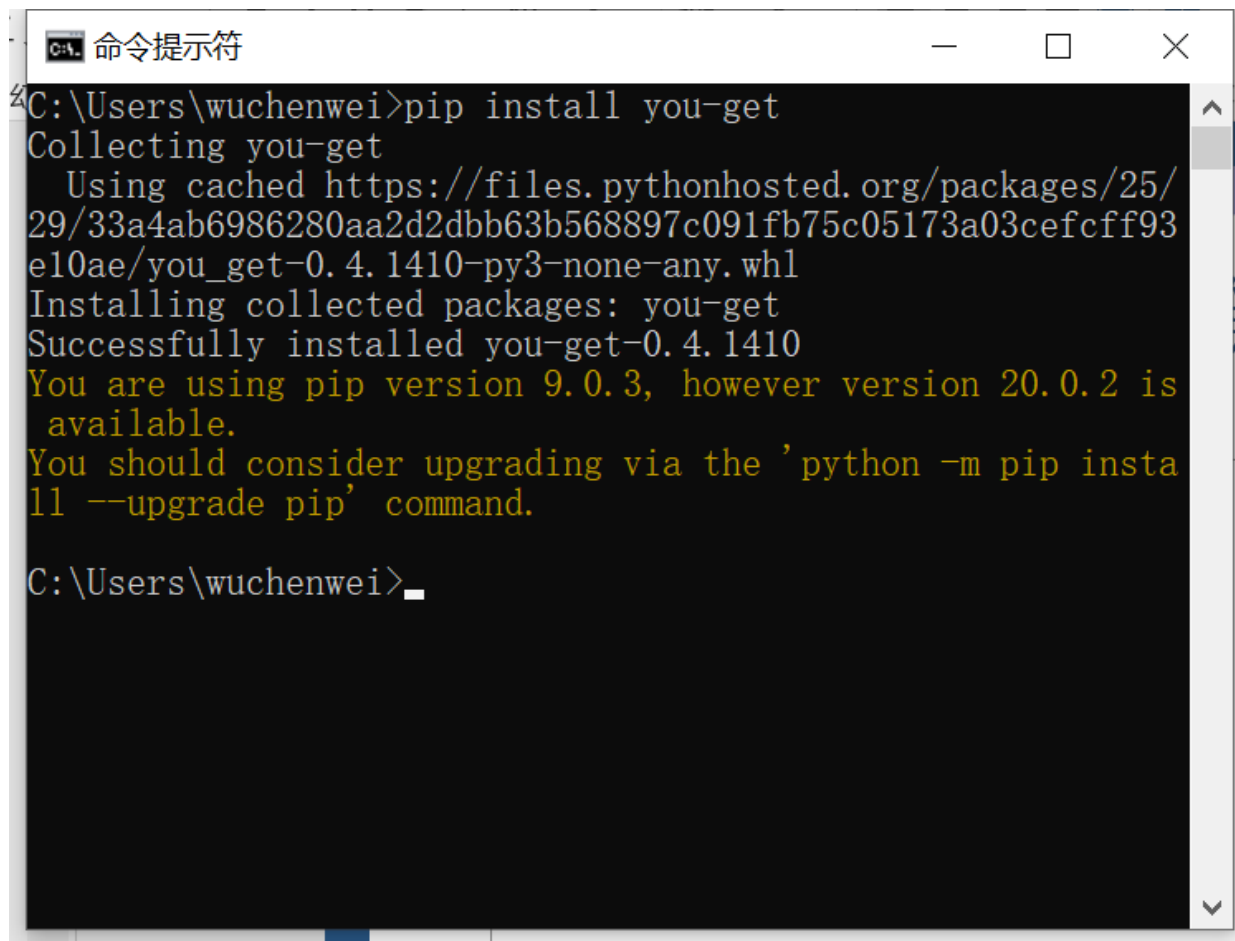
注意：mac因为内置安装了python2，所以在使用pip的时候要用pip3指令

pip install 下载you-get

- you-get是一个用来下载视频的python模块，我们可以使用这个库来让python帮我们下载B站中的视频，首先，我们需要在cmd中输入以下命令安装模块（pip for windows, pip3 for mac）：

pip install you-get

pip3 install you-get



```
命令提示符
C:\Users\wuchenwei>pip install you-get
Collecting you-get
  Using cached https://files.pythonhosted.org/packages/25/29/33a4ab6986280aa2d2dbb63b568897c091fb75c05173a03cefcff93e10ae/you_get-0.4.1410-py3-none-any.whl
Installing collected packages: you-get
Successfully installed you-get-0.4.1410
You are using pip version 9.0.3, however version 20.0.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
C:\Users\wuchenwei>
```

you-get命令

■ 下载单个视频:

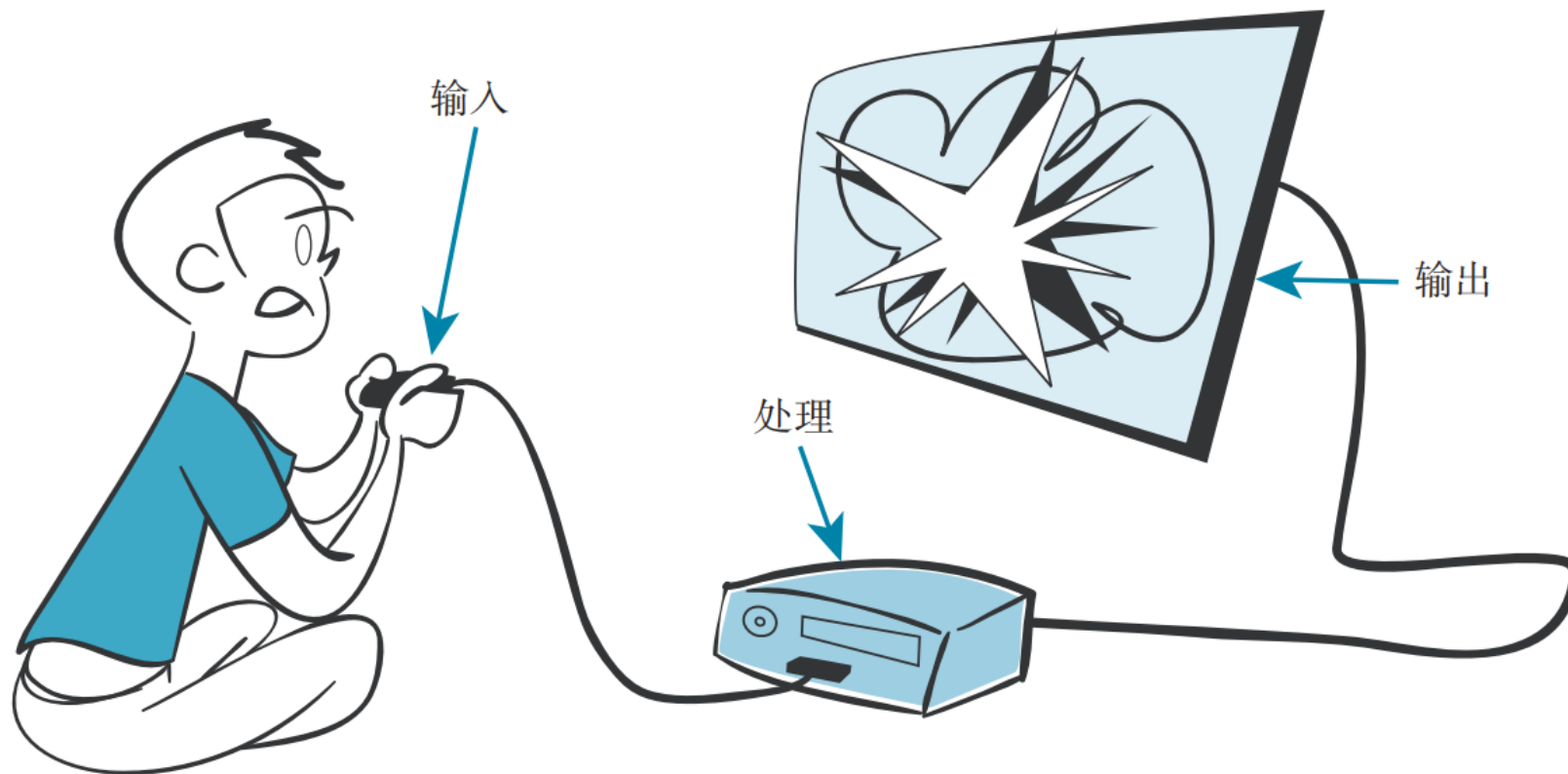
```
you-get -o target_path target_url
```

■ 批量下载视频列表:

```
you-get --playlist -o target_path target_url
```

字符串变量的使用

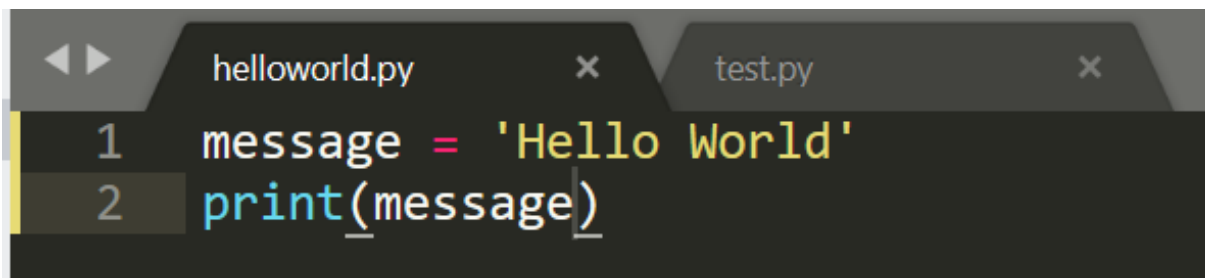
程序的本质是数据变化的过程



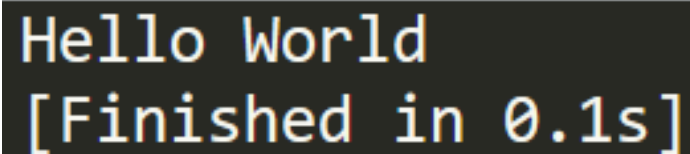
经典的“输入 (input) 输出 (output)”模型，本质上就是数据流的变化

变量：数据的标签

- 世界是数字构成的，计算机在内存（Memory）中存放数据。
- 我们使用变量帮助人类标记（记忆）这些数据。
- 字符串是计算机中重要的一种数据类型。

A screenshot of a code editor with two tabs: 'helloworld.py' and 'test.py'. The 'helloworld.py' tab is active and shows two lines of Python code: '1 message = 'Hello World'' and '2 print(message)'. The code is written in a dark-themed editor with syntax highlighting.

```
helloworld.py x test.py x
1 message = 'Hello World'
2 print(message)
```

A screenshot of a terminal window showing the output of the Python script. It displays 'Hello World' on the first line and '[Finished in 0.1s]' on the second line, indicating the execution time.

```
Hello World
[Finished in 0.1s]
```

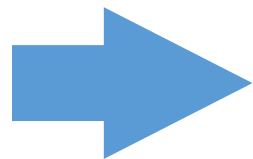
变量的命名规则

- 起个好名字是编程成功的一半
- 尽可能使用有意义的文字给变量命名，比如 `power_level` , `tax_rate` 等。
- Python中变量名可以包含字母、下划线、数字等，但是不能以数字开头；
- 变量名不能与Python语法中规定的保留字相同，比如`import`等高亮字体；
- Python是大小写敏感语言，比如 `abc` 和 `aBc`是两个不同的变量；
- 更多规则参考PEP（Python Enhancement Proposals）标准规范。

创建字符串：引号的使用

- 在python中我，们使用单引号、双引号、三引号来创建字符串
- 对于普通字符串，单引号就够了
- 如果我们的字符串中也包含引号，则使用双引号比较方便
- 我们也使用三引号来进行多行字符串的引用和注释

```
1 str_sgq = 'This Dr.Wu.'  
2 str_dbq = "I'm Dr.Wu."  
3 str_trq = '''Helle! everyone.  
4 I'm Dr.Wu  
5 '''  
6 print(str_sgq)  
7 print(str_dbq)  
8 print(str_trq)
```



```
This Dr.Wu.  
I'm Dr.Wu.  
Helle! everyone.  
I'm Dr.Wu  
  
[Finished in 0.1s]
```

Python中的转义字符 (escape)

Python和其他很多程序语言一样，定了许多转义字符，这些字符用反斜杠（\）开头，在字符串中表示一些特殊的字符，比如引号。Python中的转移字符如下表所示：

转义字符	描述
\\(在行尾时)	续行符
\\	反斜杠符号
\'	单引号
\"	双引号
\a	响铃
\b	退格(Backspace)
\e	转义
\000	空

\n	换行
\v	纵向制表符
\t	横向制表符
\r	回车
\f	换页
\oyy	八进制数，yy代表的字符，例如：\o12代表换行
\xyy	十六进制数，yy代表的字符，例如：\x0a代表换行
\other	其它的字符以普通格式输出

SyntaxError: 语法错误

- Syntax（语法）在编程语言是指一个句子中单词的排列和顺序
- 切记：字符串要放在成对的引号中如果出现了没有成对的情况，python就会报错，其中的EOL是End-Of-Line:

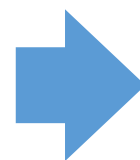
- `SyntaxError: EOL while scanning string literal`

- 解决方法就是把确实的引号补在合适的位置上
- 通俗的讲，也就是字符串的引号必须“成双成对出现”

在字符串中引用变量

- 如果字符串中需要包含变量，可以使用以下两种方法来实现。
 1. %方法：使用%s在字符串中占位，在字符串结尾用%引用；
 2. Fomat方法：使用花括号占位，在结尾用.format引用；
 3. f字符串方法。引号前加上字母f并使用花括号来引用变量。

```
name = "Allen"  
emotion = "happy"  
print("%s is %s" % (name,emotion))  
print("{} is {}".format(name,emotion))  
print(f"{name} is {emotion}.")
```

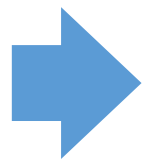


```
Allen is happy  
Allen is happy  
Allen is happy.  
[Finished in 0.1s]
```

字符串的乘法：重复

- 字符串的乘法（*），一般是字符串*整数，表示对该字符串进行重复操作，并返回重复之后的字符串。

```
string1='*'  
string2=string1*2  
print(string1)  
print(string2)
```



```
*  
**  
[Finished in 0.1s]
```

字符串的加法：拼接

- 我们可以使用+来拼接两个字符串，并且返回拼接之后的。我们来使用这种方法，配合字符串的乘法打印一条分割线。

```
16 string1='*'
17 string2=string1*20+'我是分割线'+string1*20
18 print(string1)
19 print(string2)
20 print(string1)
```



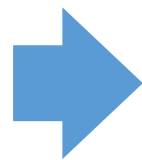
```
*
*****我是分割线*****
*
[Finished in 0.1s]
```

字符串的大小写转换

■ Python为字符串提供了三种方法来转换字符串中字母的大小写：

1. upper()方法：将字符串全部转换为大写字母
2. lower()方法：将字符串全部转换为小写字母
3. title()方法：将字符串转换成称呼模式

```
22 name = 'koBe bryant'  
23 print(name.upper())  
24 print(name.lower())  
25 print(name.title())
```



```
KOBE BRYANT  
kobe bryant  
Kobe Bryant  
[Finished in 0.1s]
```

更多字符串的用法

- **dir()函数**

用dir()函数来查看一下指定模块的功能列表。

- **help() 函数**

用于查看函数或模块用途的详细说明。

```
>>> dir(str)
['_add_', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'count', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isdecimal', 'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
>>> help(str.replace)
Help on method_descriptor:

replace(...)
    S.replace(old, new[, count]) -> str

    Return a copy of S with all occurrences of substring
    old replaced by new.  If the optional argument count is
    given, only the first count occurrences are replaced.

>>>
```


终于可以完成今天的任务了！

在sublime中编写.py文件执行下载命令

知识点

- Import os导入系统os库
- 变量的定义和使用
- 使用字符串拼接命令
- 使用os.system()向操作系统

发送命令（相当于在cmd

窗口中进行命令操作）

- you-get命令的写法

注：也可以直接在cmd中输入该命令

```
1 import os
2 url = 'https://www.bilibili.com/video/av1415480'
3 command = 'you-get --playlist -o F://bilibili/jaychou '+url
4 os.system(command)
5
```

```
site: Bilibili
title: 【经典】周杰伦MV 【185P】 (P1. 可爱女人【1st JAY】)
stream:
  - format: flv
    container: flv
    quality: 高清 1080P
    size: 17.4 MiB (18284207 bytes)
    # download-with: you-get --format=flv [URL]
```

Downloading 【经典】周杰伦MV 【185P】 (P1. 可爱女人【1st JAY】).flv ...

0.0% (0.0/ 17.4MB)		[1/1]	
1.4% (0.2/ 17.4MB)		[1/1]	1 MB/s
2.9% (0.5/ 17.4MB)		[1/1]	5 MB/s
4.3% (0.8/ 17.4MB)		[1/1]	661 kB/s
5.7% (1.0/ 17.4MB)		[1/1]	198 kB/s
7.2% (1.2/ 17.4MB)		[1/1]	213 kB/s
8.6% (1.5/ 17.4MB)		[1/1]	242 kB/s
10.0% (1.8/ 17.4MB)		[1/1]	238 kB/s
11.5% (2.0/ 17.4MB)		[1/1]	249 kB/s

作业

- 编写python程序下载你喜欢的B站视频
- 完成PythonCrashCourse CH1/CH2的课后习题