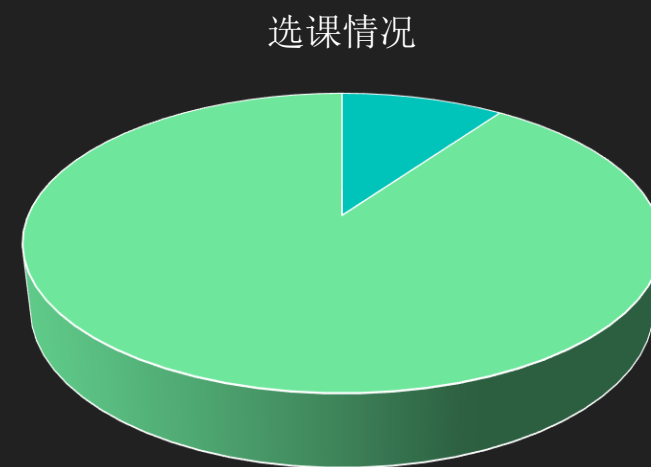


Python与深度学习基础

张越一 zhyuey@ustc.edu.cn

课程信息

- 课程名：Python与深度学习基础
- 教室：3C102（从3A113换到此教室）
- 时间：2019年春季学期 第1-14周 星期一晚
- 授课教师：张越一 熊志伟 郑歆慰 程栋梁
- 学分：2
- 课时：40
- 学生数：60+英才班



■选中人数-60 ■未选中人数 - 550

<10%

课程信息

○ 大纲：

- Python语言的基本特性
- Python语言的高级特性
- 图像处理
- 网络编程
- 人工智能+神经网络
- 深度学习（卷积神经网络）
- 深度学习框架
- 深度学习实战

○ 考核：

- Python编程大作业
- 深度学习大作业

○ 考核方式：

- 代码
- 论文
- Demo或答辩

○ 成绩：

- 五分制

课程信息

- 教材：课程PPT

- 参考教材：

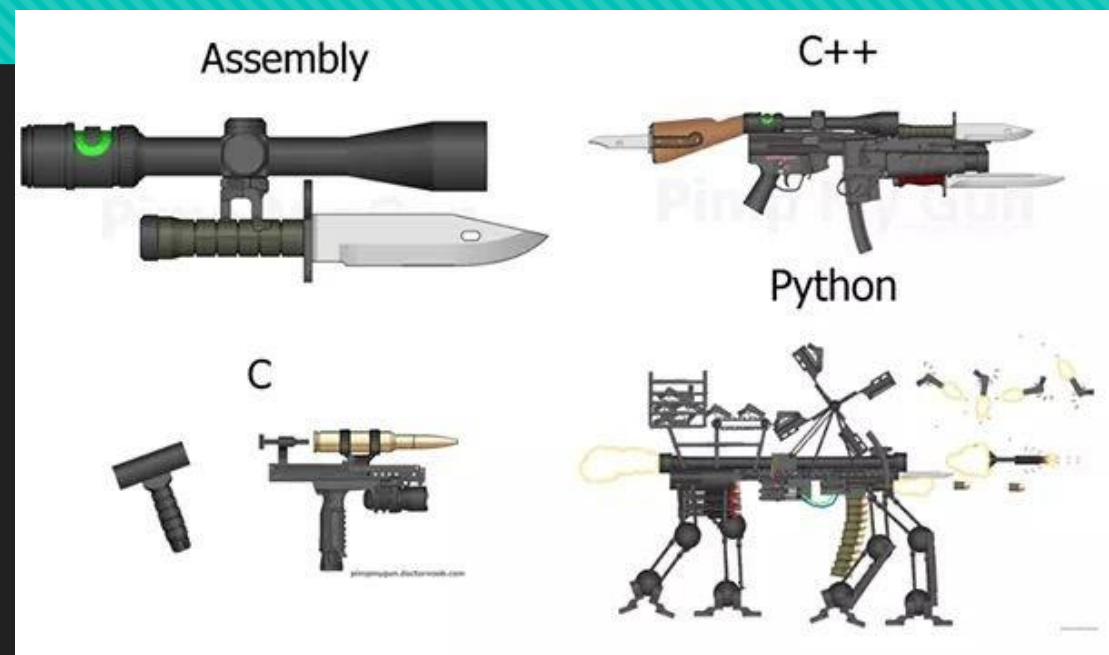
- Python核心编程 Wesley Chun等
- 人工智能-一种现代方法 Stuart J. Russell等
- 深度学习 Ian GoodFellow等

- 网络资源：

- 廖雪峰的Python教程
- <https://github.com/vinta/awesome-python>
- <https://www.practicepython.org/>

Python

- 大蟒蛇, Guido Von Rossum, 荷兰人, CWI
- 命令式编程, 函数式编程, 面向对象编程, 胶水语言
- 1991年, 第一个Python的编译器诞生, 使用C语言实现
- Jython (基于Java), IronPython (基于C#)
- 特点:
 - 可扩展 语法精简
 - 跨平台 动态语言
 - 面向对象 丰富的数据结构
 - 健壮性 强大的社区支持



人生苦短，
何不学python?

Python

- 谁在用python?
- Facebook, Google, Youtube, Spotify, Netflix, Quora, Dropbox, ...
- 每个AI公司都在用python
- https://www.hartmannsoftware.com/Blog//Companies_Using_Python

Oct 2018	Oct 2017	Change	Programming Language	Ratings	Change
1	1		Java	17.801%	+5.37%
2	2		C	15.376%	+7.00%
3	3		C++	7.593%	+2.59%
4	5	⬆	Python	7.156%	+3.35%
5	8	⬆	Visual Basic .NET	5.884%	+3.15%
6	4	⬇	C#	3.485%	-0.37%
7	7		PHP	2.794%	+0.00%
8	6	⬇	JavaScript	2.280%	-0.73%
9	-	⬆	SQL	2.038%	+2.04%
10	16	⬆	Swift	1.500%	-0.17%
11	13	⬆	MATLAB	1.317%	-0.56%
12	20	⬆	Go	1.253%	-0.10%

Python

I 你用 PYTHON 做什么

- 2018
- 2017



2017 Average Developer Salary in the U.S.

indeed.com
estimations (USD)

Language

#1	117,147	Ruby/Ruby on Rails
#2	116,027	Python
#3	115,597	C++
#4	115,273	iOS
#5	110,062	JavaScript
#6	102,043	Java
#7	95,045	C
#8	86,354	PHP
#9	85,812	SQL

Python与教育

- 美国计算机协会会刊（CACM）2014年7月发表的调查报告显示，Python“已经成为目前美国顶尖大学里最受欢迎的计算机编程入门语言”。
- 在计算机专业排名前10的学校里，有8所学校用Python作为编程入门语言
- 在计算机专业排名前39的学校里，有27所学校用Python作为编程入门语言
- Python有望在2019年成为法国高中编程教学语言
- 浙江，北京和山东已经确定把Python编程基础纳入信息技术课程和高考的内容体系
- Python语言已经进入全国计算机等级考试， 2018

Python

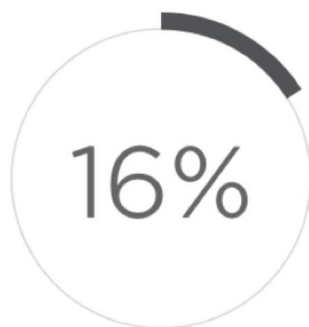
○ 版本

- 最新： 3.7.2 和 2.7.15
- 2 与 3 的争论
- 本课使用3.6+

I PYTHON 3 VS PYTHON 2



Python 3



Python 2

Release version	Release date
Python 3.7.2	2018-12-24
Python 3.6.8	2018-12-24
Python 3.7.1	2018-10-20
Python 3.6.7	2018-10-20
Python 3.5.6	2018-08-02
Python 3.4.9	2018-08-02
Python 3.7.0	2018-06-27

Python安装

- 系统自带或安装
 - (ubuntu) apt install python
 - (centos) yum install rh-python36
- Python官网下载
 - www.python.org
- 源码编译
- 下载其他发行版



pythonTM

Anaconda

○ Anaconda

- 一个开源的Python发行版，包含了上百个科学包
- <https://www.anaconda.com/>
- <http://mirrors.ustc.edu.cn/anaconda/archive/> 科大源
- 推荐下载版本5.1.0以上
- 支持Windows, Linux, MacOS
- 记得修改anaconda源 <http://mirrors.ustc.edu.cn/help/anaconda.html>



人生苦短，
何不用anaconda?

Python IDE



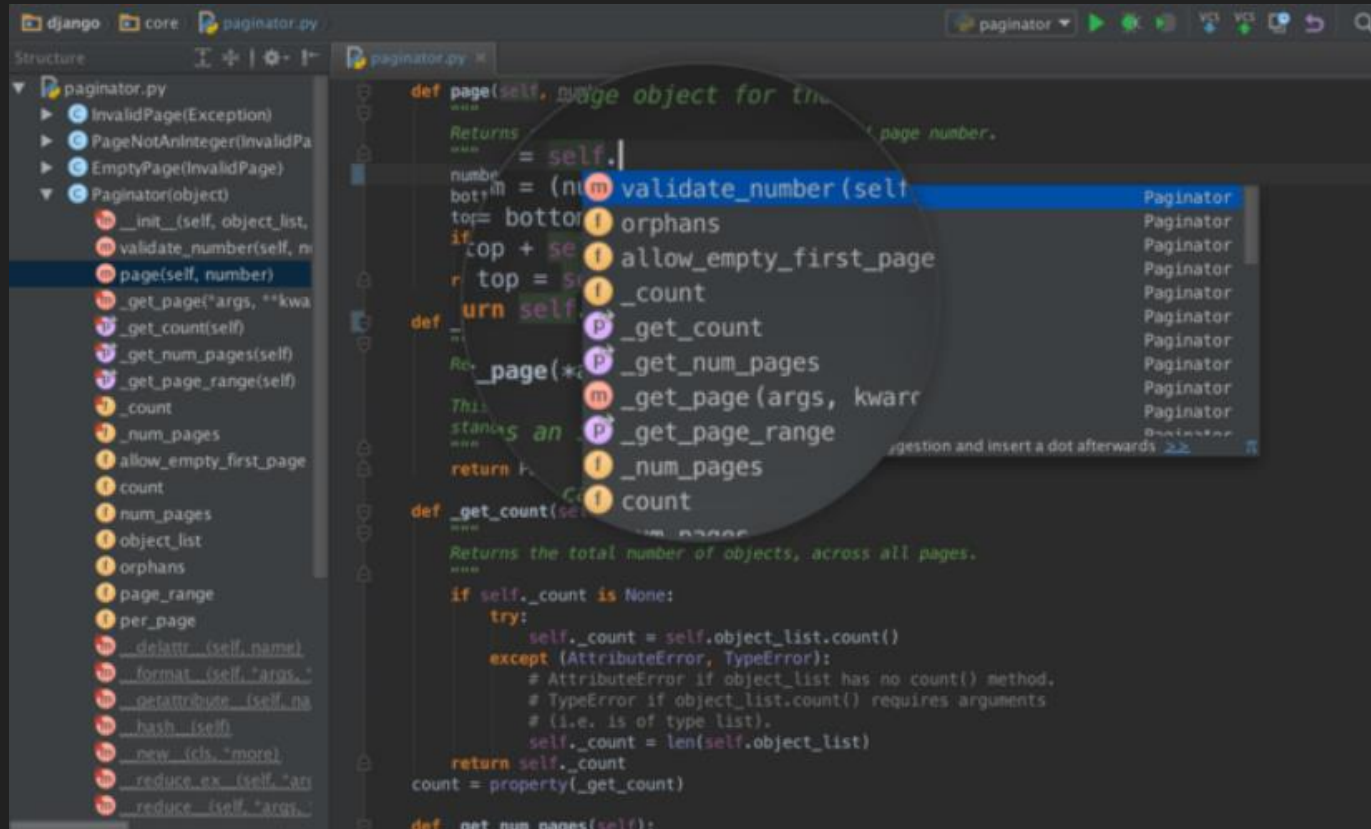
& Many more

www.IndianAIExpert.com

推荐
PyCharm



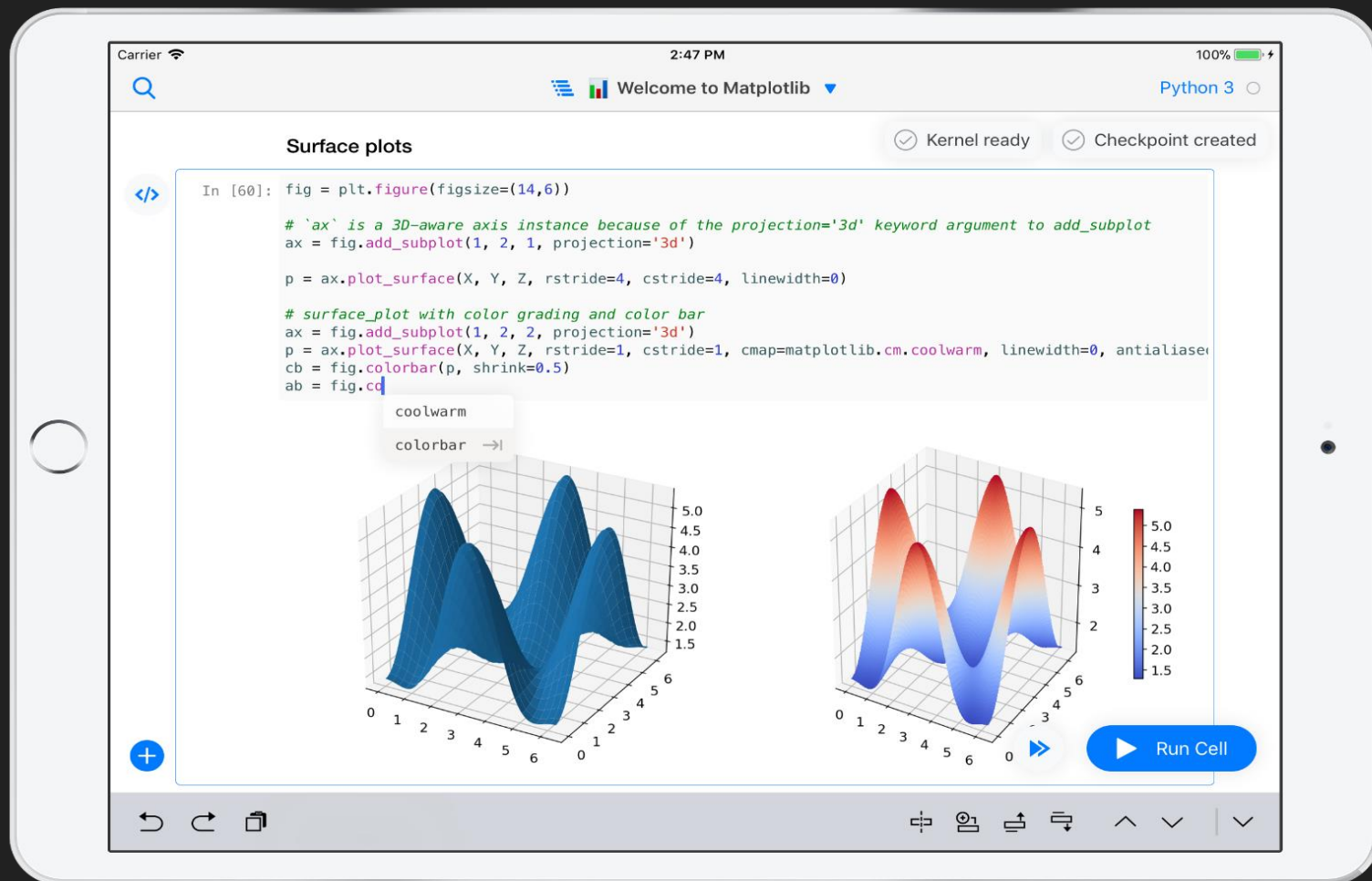
PyCharm



Jupyter



可视化笔记本
基于ipython



Python交互

- 在命令行下输入Python,或打开Python命令行，也可以使用anaconda的命令符

```
(base) C:\Users\zhyue>python
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- 在此处可以直接输入命令来执行

```
>>> 1+1
2
>>> import math
>>> math.sqrt(5)
2.23606797749979
>>> math.sin(3.14)
0.0015926529164868282
>>>
```

Python

- Python 执行一个文件
 - `python filename.py`
- Python 执行文件带命令行
 - `python filename.py argu1 argu2`
- Anaconda下试用ipython
- Anaconda下试用Jupyter
 - Jupyter notebook
 - 本课使用

```
(base) C:\Users\zhyue>ipython
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 12:04:43)
Type 'copyright', 'credits' or 'license' for more info
IPython 6.2.1 -- An enhanced Interactive Python. Type '?' for help

In [1]: import math

In [2]: math.sqrt(999)
Out[2]: 31.606961258558215
```

```
(base) C:\Users\zhyue>jupyter notebook
[I 11:10:47.258 NotebookApp] JupyterLab beta preview enabled
[C 11:10:47.258 NotebookApp] WARNING: The JupyterLab application directory is not set
[C 11:10:47.258 NotebookApp] JupyterLab application directory is not set
jupyter\lab
```


Print与数据类型

- Print 输出信息，打印信息
- 对于Python3,使用Print()
- 语法
 - `print("message")`
 - `print(Expression)`
- 例
 - `print("Hello, World!")`
 - `print(3, 6, "hello word")`

- Numeric
 - 整型
 - `print(2006, 0xffff, 0b101101)`
 - 浮点型
 - `print(1.234, -1.23e11)`
- 布尔值 True False `3 > 2` True or False

变量与常量

- Python 中的变量赋值不需要类型声明
- 等号 (=) 运算符左边是一个变量名
- 等号 (=) 运算符右边是存储在变量中的值
- `areas = 960`
- `g = 9.8`
- 多个等号的情况亦可
- `a = b = c = 1`
- `a, b, c = 1, 2, 3`
- Python中没有特别的定义常量的方式
- 通常使用大写字母来做变量名来表示常量
- `USER = 'Robert'`
- `PI = 3.14159`

字符串

- 单引号 'Hello'
- 双引号 "World"
- 三引号 可以跨行

```
>>> str0 = """Hello
... World
... """
>>> str0
'Hello\nWorld\n'
>>> print(str0)
Hello
World
```

- str0 str0[0] str0[4]
- str0[1:4] str0[1:4:2]
- str0[-1] str0[-3:-1]

- 字符串拼接 +
- "Hello" + " " + "World!"

- 字符串格式化

```
>>> str1 = "Your %s is %.2f" % ('GPA', 3.0)
>>> str1
'Your GPA is 3.00'
>>> str2 = '{0}, {1}, {2}'.format('a', 'b', 'c')
>>> str2
'a, b, c'
```

- 参考
- <https://docs.python.org/3.7/library/string.html?highlight=string#module-string>

数据结构

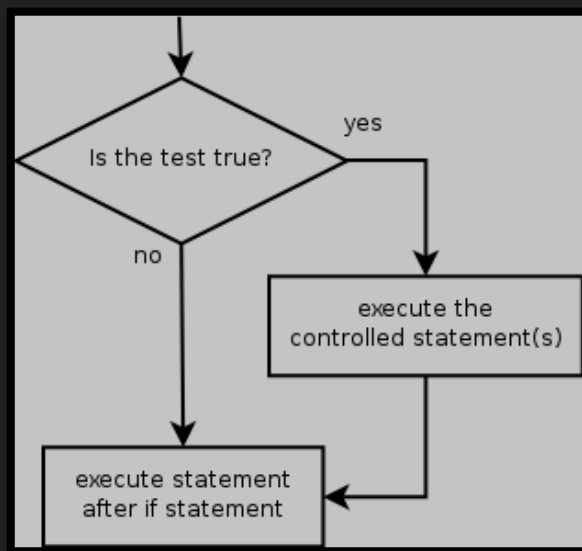
- **List** 是有序且可变的列表
 - `L = [1, 2, 3]`
- **Tuple** 是有序但不可变的列表
 - `T = ("good", "bad")`
- **Set**是无序无标签的集合。
 - `S = {89, 90}`
- **Dict**是无序可变有标签的集合。
 - `D = {"John":56, "Robert":78}`
- 取长度 `len()`
 - `print(len(S))`

Tuples vs. Lists

- Lists slower but more powerful than tuples
 - Lists can be modified and they have many handy operations and methods
- Tuples are immutable & have fewer features
 - Sometimes an immutable collection is required (e.g., as a hash key)
 - Tuples used for multiple return values and parallel assignments
 - `x,y,z = 100,200,300`
 - `old,new = new,old`
- Convert tuples and lists using `list()` and `tuple()`:
 - `mylst = list(mytup); mytup = tuple(mylst)`

条件判断

```
if condition:  
    statements
```



```
if condition1:  
    statements1  
elif condition2:  
    statements2  
else:  
    statements3
```

例子:

```
If age > 60:  
    print("old")  
elif age > 18:  
    print("middle")  
else:  
    print("young")
```

没有大括号
只有缩进，严格意义的缩进
但千万别忘了冒号：：

循环

```
for val in values:  
    do_something(val)
```

```
while condition:  
    do_something()
```

例子:

```
for val in range(5):  
    print(val*val)
```

输出:

0
1
4
9
16

例子:

```
i=0  
while i < 5:  
    print(i * i * i)  
    i = i + 1
```

输出:

0
1
8
27
64

模块安装

- 内建模块与外部模块
 - 内建: math, os, sys 等
 - 外部: flask, opencv, tensorflow
- 如何在anaconda里面加入新模块
 - `conda install module_name`
- 特殊的库
 - `conda install -c conda-forge opencv`
- 已经安装的模块
 - `conda list`
 - `pip list`
- 用pip安装 (Anaconda包括pip)
 - `pip install flask`
- 删除模块
 - `conda remove -n myenv scipy`
 - `pip uninstall mypackage`

内建模块

- Math 模块
- `from math import *`

Constant	Description
e	2.7182818...
pi	3.1415926...

Command name	Description
<code>abs(value)</code>	absolute value
<code>ceil(value)</code>	rounds up
<code>cos(value)</code>	cosine, in radians
<code>floor(value)</code>	rounds down
<code>log(value)</code>	logarithm, base e
<code>log10(value)</code>	logarithm, base 10
<code>max(value1, value2)</code>	larger of two values
<code>min(value1, value2)</code>	smaller of two values
<code>round(value)</code>	nearest whole number
<code>sin(value)</code>	sine, in radians
<code>sqrt(value)</code>	square root

虚拟环境

- Python应用程序通常会使用不在标准库内的软件包和模块。应用程序有时需要特定版本的库，因为应用程序可能需要修复特定的错误，或者可以使用库的过时版本的接口编写应用程序。
- 这意味着一个Python安装可能无法满足每个应用程序的要求。如果应用程序A需要特定模块的1.0版本但应用程序B需要2.0版本，则需求存在冲突，安装版本1.0或2.0将导致某一个应用程序无法运行。
- 这个问题的解决方案是创建一个虚拟环境 virtual environment，一个目录树，其中安装有特定Python版本，以及许多其他包。

虚拟环境 (以anaconda为例)

- 默认是名为base的虚拟环境
- 查看安装了哪些虚拟环境
 - `conda env list`
- 创建一个虚拟环境
 - `conda create -n env_name python=3.6`
 - `conda create -n env_name numpy python=2.7`
- 激活虚拟环境
 - Linux
 - `source activate your_env_name`
 - windows
 - `activate your_env_name`
- 退出虚拟环境
 - Linux
 - `source deactivate your_env_name`
 - windows
 - `deactivate your_env_name`
- 删除环境
 - `conda remove -n your_env_name --all`
- 删除虚拟环境中的包
 - `conda remove --name your_env_name package_name`

模块

- Anaconda模块列表
- https://docs.anaconda.com/anaconda/packages/old-pkg-lists/5.1.0/py3.6_win-64/

Packages included in Anaconda 5.1.0 for 64-bit Windows with Python 3.6

Platform: Windows 64-bit

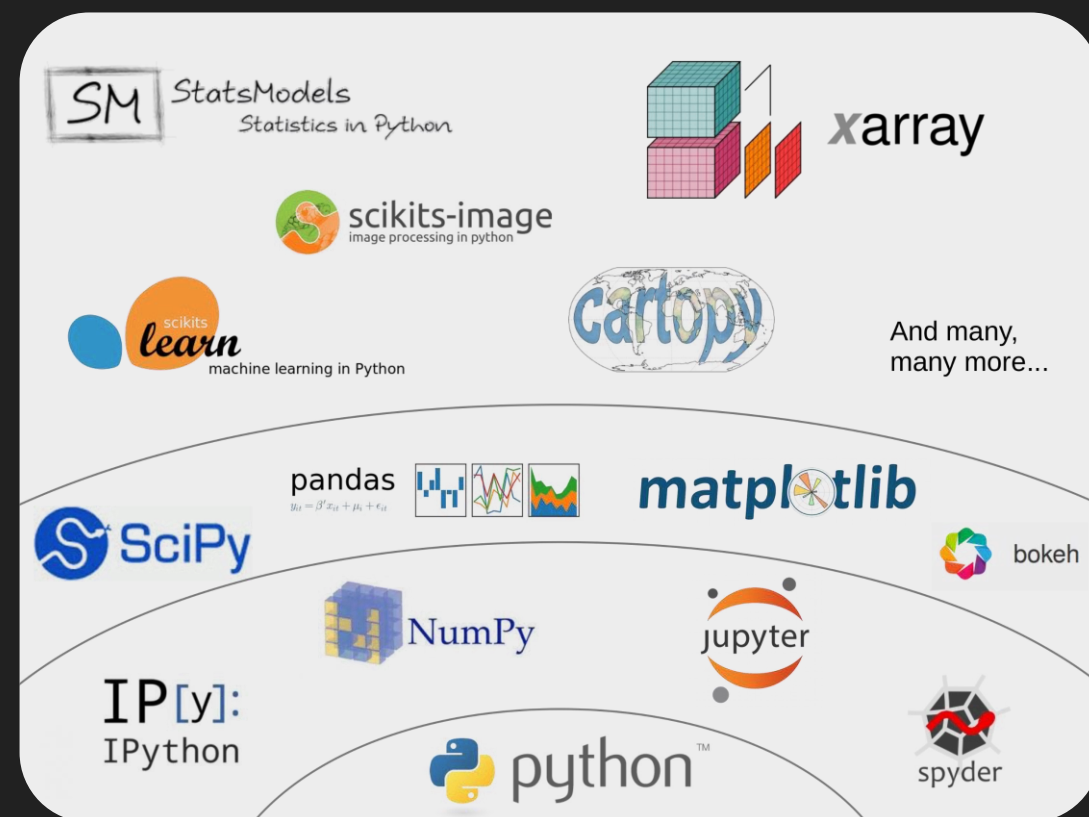
Python version: 3.6

Number of supported packages: 444

Name	Version	Summary / License	In Installer
<code>_ipyw_jlab_nb_ext_conf</code>	0.1.0	A configuration metapackage for enabling Anaconda-bundled jupyter extensions / BSD	✓
<code>_nb_ext_conf</code>	0.4.0	A configuration metapackage for enabling Anaconda-bundled jupyter extensions / BSD	
<code>affine</code>	2.1.0	Matrices describing affine transformation of the plane. / BSD 3-Clause	
<code>aiohttp</code>	2.3.9	Async http client/server framework (asyncio) / Apache 2.0	
<code>alabaster</code>	0.7.10	Configurable, Python 2+3 compatible Sphinx theme. / BSD 3-Clause	✓
<code>anaconda-clean</code>	1.1.0	Delete Anaconda configuration files / BSD	
<code>anaconda-client</code>	1.6.9	Anaconda Cloud command line client library / BSD 3-Clause	✓

常用模块

- 图像处理
 - PIL, OpenCV
- GUI
 - PyQt4/5, tkinter, pygtk
- 网络相关
 - Requests, flask, urllib, beautifulsoup
- 游戏相关
 - Pygame
- 更详尽的列表, 请见
- <https://www.zhihu.com/question/24590883>



如何学习Python?

- 在“做”中“学”
 - 10 ways to learn Python
 - <https://simpleprogrammer.com/get-started-learning-python/>
- 读好的代码，去github上找
- 养成良好的编码习惯
 - google的Python编码规范
 - https://zh-google-styleguide.readthedocs.io/en/latest/google-python-styleguide/python_style_rules/
 - PEP8
 - <https://legacy.python.org/dev/peps/pep-0008/>
- 能用库的，不要自己造轮子



课程

○ 课后作业

- 安装Python (推荐安装anaconda)
- 安装Python IDE (推荐Pycharm)
- 尝试使用Jupyter
 - anaconda下输入jupyter notebook来打开Jupyter, 然后点Upload把ipynb文件导入
- 练习本节课的知识点
 - http://staff.ustc.edu.cn/~zhyuey/python_basics.zip
 - 解压后有ipynb文件, 用Jupyter打开
 - 在每个代码框中, 按Shift+Enter来运行
- 自学git, 了解版本控制系统
- 在github, gitee上各开一个账号

○ 下节课

- 介绍大作业
- 介绍Python的其他高级特性