

# Day 01. Python 环境配置 & 基础语法

## Day 01. Python 环境配置 & 基础语法

### 1. 安装与环境配置

Python 版本区别

安装 Python

安装 PyCharm

Anaconda 的安装与使用

Python 工作原理

安装 Anaconda

Anaconda 环境配置

管理虚拟环境

PyCharm 配置

Anaconda 命令列表

### 2. 基础语法

### 3. 字符串

字符串运算

字符串格式化

字符串内建函数

### 4. 作业

## 1. 安装与环境配置

### Python 版本区别

#### 1. Windows help file

帮助文档

#### 2. Windows x86-64 embeddable zip file

嵌入式版本

#### 3. Windows x86-64 executable installer

64 位可执行文件

#### 4. Windows x86-64 web-based installer

5. [Windows x86 embeddable zip file](#)
6. [Windows x86 executable installer](#)
7. [Windows x86 web-based installer](#)

## 安装 Python

1. [Windows x86-64 executable installer](#)
2. check [Add Python 3.7 to PATH](#)
3. select [Customize installation](#)
4. select [Disable path length limit](#)
5. open CMD [REPL](#) Read - Evaluate - Print Loop

```
1. C:\Users\suzu>py
2. Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:57:15) [MSC v.1915 64
   bit (AMD64)] on win32
3. Type "help", "copyright", "credits" or "license" for more information.
4. >>> print('Hello, Python!')
5. Hello, Python!
6. >>> exit()
7.
8. C:\Users\suzu>
```

## 安装 PyCharm

- 略

## Anaconda 的安装与使用

[ANACONDA - The Most Popular Python Data Science Platform](#)

### Python 工作原理

1. Python 解释器

```
1. PATH_TO_PATHON
2. python.exe
```

2. Python 包集合

```
1. PATH_TO_PATHON
2. Lib
3. site-packages
```

## 安装 Anaconda

### 1. Anaconda Navigator

用于管理工具包和环境的图形用户界面，后续涉及的众多管理命令也可以在 Navigator 中手工实现

### 2. Jupyter notebook

基于 web 的交互式计算环境，可以编辑易于人们阅读的文档，用于展示数据分析的过程

### 3. qtconsole

一个可执行 IPython 的仿终端图形界面程序，相比 Python Shell 界面，qtconsole 可以直接显示代码生成的图形，实现多行代码输入执行，以及内置许多有用的功能和函数

### 4. spyder

一个使用 Python 语言进行跨平台科学运算集成开发环境

## Anaconda 环境配置

- 环境变量

```
d:\Anaconda3;
```

```
d:\Anaconda3\Library\mingw-w64\bin;
```

```
d:\Anaconda3\Library\usr\bin;
```

```
d:\Anaconda3\Library\bin;d:\Anaconda3\Scripts
```

- 测试安装结果

```
1. C:\Users\Administrator>conda --version
2. conda 4.5.11
```

- 升级现有包

```
1. C:\Users\Administrator>conda upgrade --all
```

## 管理虚拟环境

- `conda env list` 查看虚拟环境

```
1. C:\Users\Administrator>conda env list
2. # conda environments:
3. #
4. base * d:\Anaconda3
```

```
1. C:\Users\Administrator>activate
2.
3. (base) C:\Users\Administrator>python
4. Python 3.7.1 (default, Oct 28 2018, 08:39:03) [MSC v.1912 64 bit (AMD64)] :: Anaconda, Inc. on win32
5. Type "help", "copyright", "credits" or "license" for more information.
6. >>>
```

```
1. C:\Users\Administrator>deactivate
```

- `conda create` 创建虚拟环境

```
1. C:\Users\Administrator>conda create -n demo python=3
```

- `activate` 切换虚拟环境

```
1. C:\Users\Administrator>conda env list
2. # conda environments:
3. #
4. base * d:\Anaconda3
5. demo d:\Anaconda3\envs\demo
```

```
1. C:\Users\Administrator>activate demo
2.
3. (demo) C:\Users\Administrator>
```

- MAC OSX

```
1. mingfei:~ mingfei$ cd /anaconda3/bin
2. mingfei:bin mingfei$ source activate demo
3. (demo) mingfei:bin mingfei$
```

- 更换源

```
1. C:\Users\Administrator>conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
2.
3. C:\Users\Administrator>conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/
4.
5. C:\Users\Administrator>conda config --set show_channel_urls yes
```

```
1. C:\Users\Administrator>conda config --show
2. ...
3. channels:
4.   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/
5.   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
6.   - defaults
7. ...
```

```
1. conda config --remove channels defaults # 删除默认源
```

- 安装第三方包

```
1. (demo) C:\Users\Administrator>python
2. Python 3.7.1 (default, Nov 24 2018, 22:14:32) [MSC v.1912 64 bit (AMD64)] :: Ana
3. conda, Inc. on win32
4. Type "help", "copyright", "credits" or "license" for more information.
5. >>> import requests
6. Traceback (most recent call last):
7.   File "<stdin>", line 1, in <module>
8. ModuleNotFoundError: No module named 'requests'
9. >>> exit()
10.
11. (demo) C:\Users\Administrator>
```

```
conda install <package-name>
```

```
pip install <package-name>
```

```
1. (demo) C:\Users\Administrator>conda install requests
2. ...
3. Proceed ([y]/n)? y
4. ...
```

```
1. (demo) C:\Users\Administrator>python
```

```
2. Python 3.7.1 (default, Nov 24 2018, 22:14:32) [MSC v.1912 64 bit (AMD64)] :: Anaconda, Inc. on win32
3. Type "help", "copyright", "credits" or "license" for more information.
4. >>> import requests
5. >>>
```

- 卸载第三方包

```
conda remove requests
```

```
pip uninstall requests
```

- 查看包信息

```
conda list
```

```
1. (demo) C:\Users\Administrator>conda list
2. ...
```

- 导出 / 导入环境

`yaml` 文件

YAML /'jæməɪl/, 是一個可讀性高, 用來表達資料序列的格式

[YAML 语言教程](#)

- 导出

```
conda env export > demo.yaml
```

```
1. (demo) C:\Users\Administrator>conda env export > demo.yaml
```

- demo.yaml

```
1. name: demo
2. channels:
3.   - defaults
4. dependencies:
5.   - asn1crypto=0.24.0=py37_0
6.   - ca-certificates=2018.03.07=0
7.   - certifi=2018.10.15=py37_0
8.   - cffi=1.11.5=py37h74b6da3_1
9.   - chardet=3.0.4=py37_1
10.  - cryptography=2.4.1=py37h7a1dbc1_0
11.  - idna=2.7=py37_0
12.  - openssl=1.1.1a=he774522_0
13.  - pip=18.1=py37_0
```

```
14. - pycparser=2.19=py37_0
15. - pyopenssl=18.0.0=py37_0
16. - pysocks=1.6.8=py37_0
17. - python=3.7.1=he44a216_5
18. - requests=2.20.1=py37_0
19. - setuptools=40.6.2=py37_0
20. - six=1.11.0=py37_1
21. - urllib3=1.23=py37_0
22. - vc=14.1=h0510ff6_4
23. - vs2015_runtime=14.15.26706=h3a45250_0
24. - wheel=0.32.3=py37_0
25. - win_inet_pton=1.0.1=py37_1
26. - wincertstore=0.2=py37_0
27. prefix: d:\Anaconda3\envs\demo
```

- 导入

```
conda env create -f demo.yaml
```

## PyCharm 配置

File | Setting | Project | Project Interpreter

Add System Interpreter

Location: `PATH_ANACONDA_DIR/envs/deom/python.exe`

## Anaconda 命令列表

1. `activate`

切换到 `base` 环境

2. `conda create -n demo python=3`

创建一个名为 `demo` 的环境并指定 `python` 版本为3 的最新版本

3. `activate demo` `deactivate`

切换 / 退出 `demo` 环境

4. `conda env list`

列出 `conda` 管理的所有环境

5. `conda list`

列出当前环境的所有包

6. `conda install requests`

安装 `requests` 包

7. `conda remove requests`

卸载 `requests` 包

8. `conda remove -n demo --all`

删除 `demo` 环境及下属所有包

9. `conda update requests`

更新 `requests` 包

10. `conda env export > demo.yaml`

导出当前环境的包信息

11. `conda env create -f demo.yaml`

用配置文件创建新的虚拟环境

## 2. 基础语法

- comment

```
1. # line comment
2.
3. '''
4. multiple line comment
5. '''
6.
7. """
8. multiple line comment
9. """
```



- data type

```
1. # 整数 int
2.
3. # HEX
4. print(0xff)
5.
6. # 浮点数 float
7.
8. print(1.234e4)
9.
10. # 复数 complex 实部和虚部都是浮点数
11.
12. print('complex', 1+2j)
13.
14. print(complex(1, 2))
15.
16. print('complex', complex(1, 2).conjugate()) # 共轭 ['kōndzūgeit]
17.
18. # 字符串 String
19.
20. print('abc')
21. print("xyz")
22.
23. # \ 转义
24.
25. print('Tom\'s name')
26. print(r'Tom\'s name')
27.
28. # \n 换行
29.
30. print('line\nanother line')
31.
32. print('''line
33. another line''')
34.
35. print(r'''line \n
36. another line''')
```

- output `print`

```
1. print('test')
2.
3. print('a', 'b', 'c')
4.
5. print(1 + 1)
```

- input `input()`

```
1. name = input()
2.
3. print(name)
```

## 3. 字符串

### 字符串运算

1. `[]`

```
1. s = 'Hello, World!'
2.
3. print(s[0])
```

2. `+`

```
1. txt1 = 'Hello, '
2. txt2 = 'World!'
3.
4. print(txt1 + txt2)
```

3. `*`

```
1. s = 'Hello, World!'
2.
3. print(s*3)
```

4. `[::step]`

```
1. s = 'Hello, World!'
2.
3. print(s[4:8])
```

5. `in`

```
1. s = 'Hello, World!'
2.
3. print('H' in s)
```

6. `not in`

```
1. s = 'Hello, World!'
2.
3. print('H' not in s)
```

7. `r/R`

```
1. # r - raw string
```

8. `%`

## 字符串格式化

## 字符串内建函数

1. `capitalize()`

```
1. s = 'hello, World!'
2.
3. print(s.capitalize())
```

2. `center(width[, fillchar])` `zfill()`

```
1. s = 'hello, World!'
2.
3. print(s.center(20, '-'))
```

3. `str.count(sub, start= 0,end=len(string))`

```
1. s = 'hello, World!'
2.
3. print(s.count('l'))
```

4. `str.endswith(suffix[, start[, end]])` `str.startswith(prefix[, start[, end]])`

```
1. s = 'Hello, World!'
2.
3. print(s.endswith('!'), 10, 13)
```

5. `str.find(str, beg=0, end=len(string))`

```
1. s = 'Hello, World!'
2.
```

```
3. print(s.find(',', 10, 13))
```

6. `str.expandtabs(tabsize=8)`

```
1. s = 'H\tello, World!'
2.
3. print(s.expandtabs(4))
```

7. `isalnum()` 字母或数字

```
1. s = 'H\tello, World!'
2.
3. print(s.isalnum())
```

8. `isalpha()`

```
1.
```

9. `isdigit()`

```
1.
```

10. `islower()` `str.isupper()`

```
1.
```

11. `isnumeric()`

```
1.
```

12. `isspace()`

```
1.
```

13. `str.join(sequence)`

```
1.
```

14. `len(string)`

```
1.
```

15. `lower()` `upper()` `swapcase()`

```
1.
```

16. `lstrip()` `rstrip()` `strip()`

```
1.
```

17. `max(str)`

```
1.
```

18. `min(str)`

```
1.
```

19. `replace(old, new [, max])`

```
1.
```

20. `str()`

```
1.
```

## 4. 作业

1. 按以下格式输出：

```
1.      登 鹤 雀 楼
2.      - 【唐】王之涣
3.
4.      白日依山尽，
5.      黄河入海流。
6.      欲穷千里目，
7.      更上一层楼。
```

2. 输出 Python 的版本号

3. 格式化输出当前时间：年-月-日 时:分:秒

4. 用户输入圆形半径，编写程序输出圆形面积

## 5. `isdigit()` `isnumeric()` `isdecimal()` 的区别