Python 基础

Day 07. Python 模块与常用库

1. 模块

模块的概念

- 一个 `.py` 文件就是一个模块 `module`

包 `package`

- 为了便于组织与管理，可以把模块放在不同的包里
- 不能与 Python 自带模块名冲突，在 Python 交互环境执行 `import abc` ，若成功则说明系统存在此模块

```
com/
    example/
        __init__.py  模块名 com.example
        test.py      模块名 com.example.test
    demo.py          模块名 com.demo
    __init__.py      模块名 com
```

`__init__.py`

- 每个包里都有一个 `__init__.py` 文件，这个文件本身也是一个模块，模块名是包名

```
def fn():
    print(1)  # 2, 3, 4
```

```
# from com import fn
#
# fn()

# from com.example import fn
#
# fn()

# from com.demo import fn
#
# fn()

# from com.example.test import fn
#
# fn()
```

2. 使用模块

- 定义模块 `com/example/test.py`

```
#!/usr/bin/evn python
# -*- coding: UTF-8 -*-

""" a test module... """

__author__ = 'tom'
```

```python
import sys


def _fn_private():
    print('this is a private function...')


def fn_test():
    """ test function... """
    print(sys.argv)
    _fn_private()


if __name__ == '__main__':
    fn_test()
```

- `#!usr/bin/evn python`

> #! 是特殊的表示符，其后面跟的是解释此脚本的 shell 的路径 shebang
>
> /usr/bin/env python
>
> 从环境变量中寻找 python，该文件在 windows / Linux / Mac 上均可运行

- `# -*- coding: UTF-8 -*-`

> Python 2 默认以 ASCII 编码，处理非 ASCII 字符需要指定字符集
>
> Python 3 默认以 UTF-8 编码，无需指定

- `if __name__ == '__main__':`

> 运行当前模块文件时，Python 解释器把一个特殊变量 `__name__` 置为 `__main__`
>
> 如果在其他地方导入该模块时，if 判断将失败
>
> 这种 if 测试可以让一个模块通过命令行运行时执行一些额外的代码，最常见的就是运行测试

- `def _fn_private():`

> 不 **需要** 外部引用的函数隐藏 `封装`

- Python File 模版

> File - Settings - Editor - File and Code Templates - Python Script

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-

# @Time     : ${DATE} ${TIME}
# @Author   : mingfei.net@gmail.com
# @FileName : ${NAME}.py
# @GitHub   : https://github.com/thu/${PROJECT_NAME}

#[[]]#
```

- 使用模块

```
# import com.example.test
#
# com.example.test.test()

from com.example.test import fn_test

fn_test()
```

- 别名

```
# 为了防止名称冲突，通过 `as` 起一个别名
from some_module import test as new_test


def test():
    print("local test function...")


test()   # local
new_test()   # import
```

3. 安装模块

命令行

```
pip install module_name
pip uninstall module_name
```

PyCharm

`File | Settings | Project Structure for Windows/Linux`
`PyCharm | Preferences | Project for macOS`

> Project Interpreter `+`

- Python 源国内镜像
  1. http://pypi.douban.com/
  2. http://mirrors.aliyun.com/pypi/simple/
  3. https://pypi.tuna.tsinghua.edu.cn/simple/

ANACONDA

`[,ænə'kɒndə]`

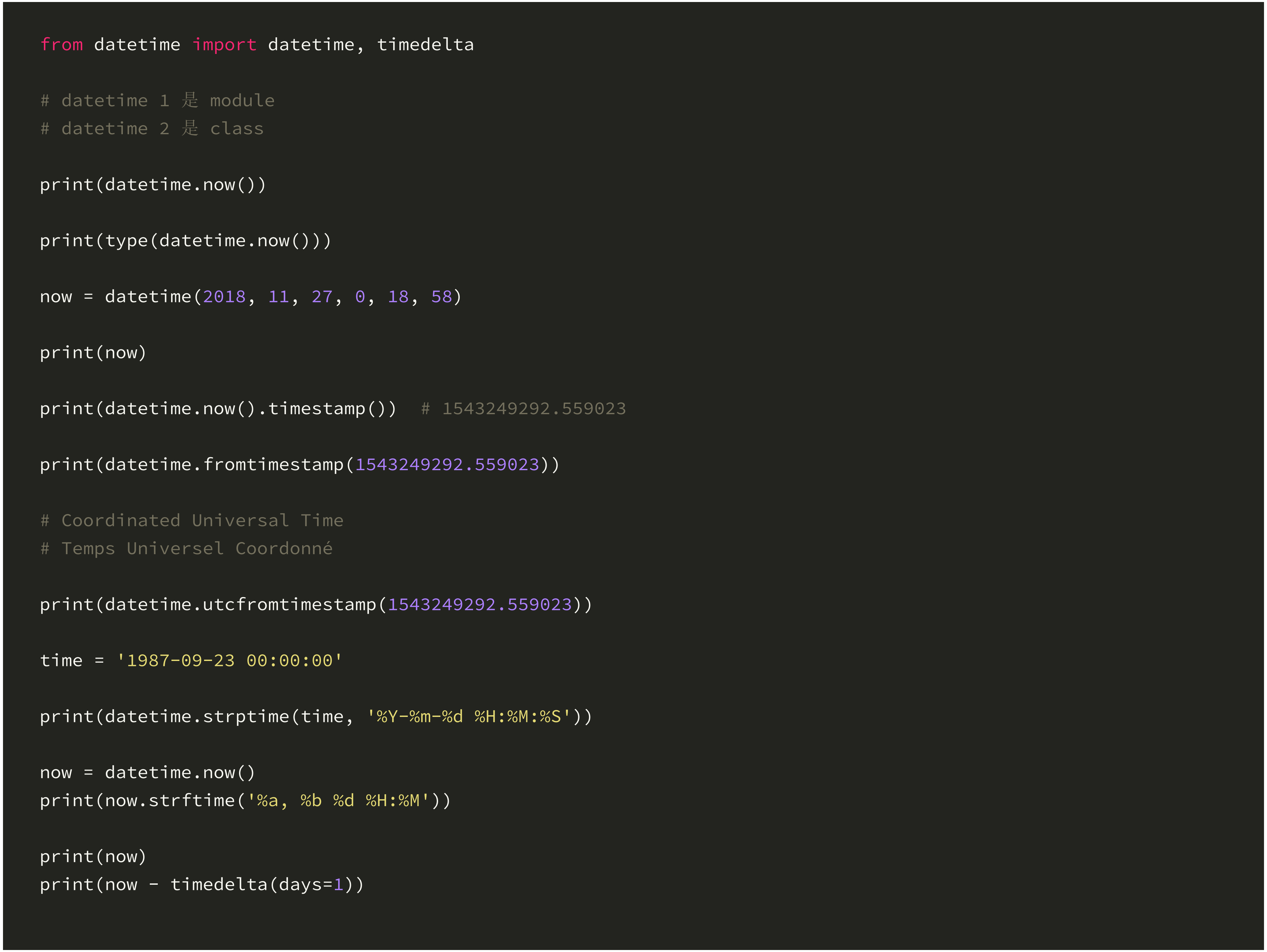
4. 内建模块

Python 标准库

```
# help('modules')

# help('datetime')
```

`datetime`

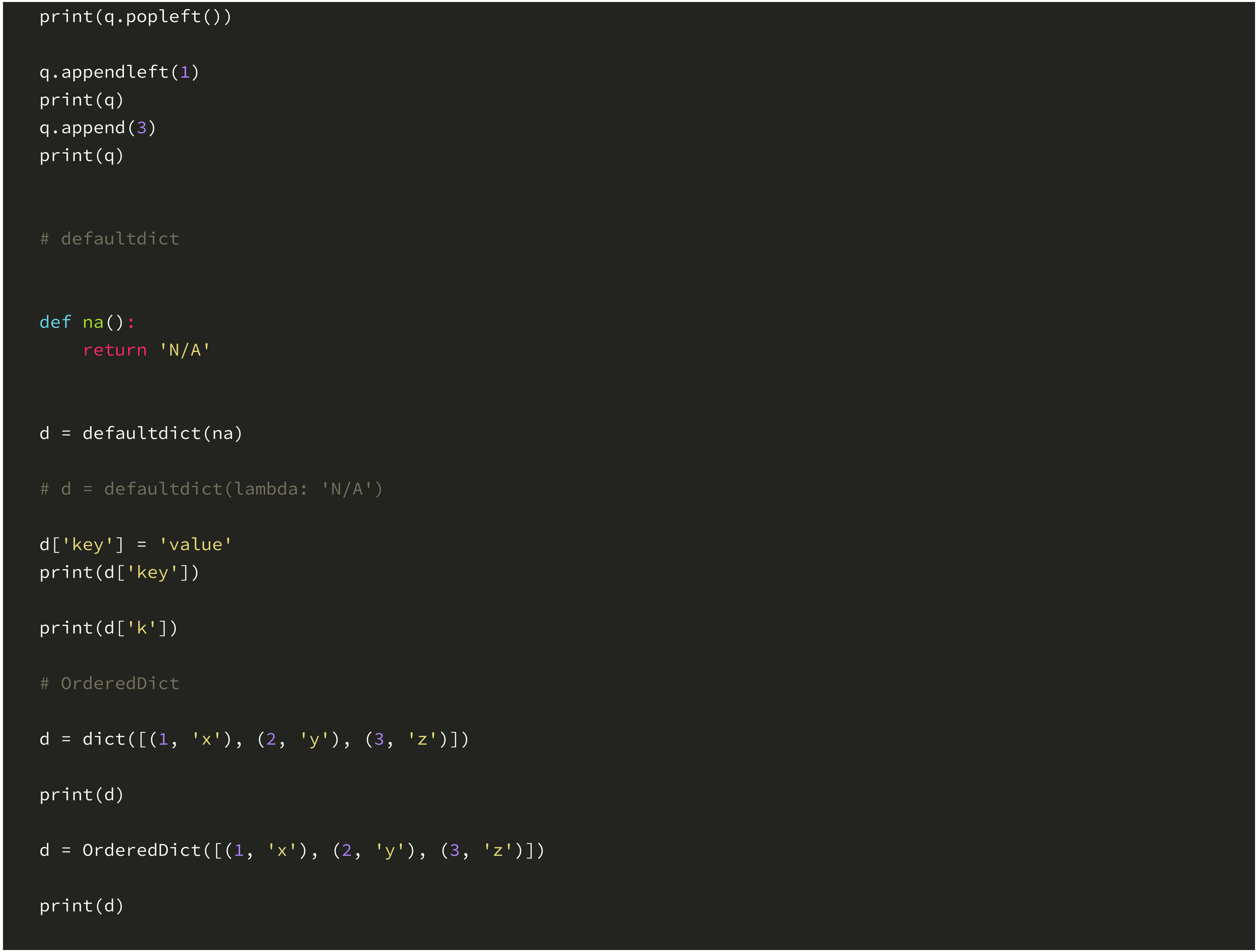
```
#!/usr/bin/env python
```



```python
from datetime import datetime, timedelta

# datetime 1 是 module
# datetime 2 是 class

print(datetime.now())

print(type(datetime.now()))

now = datetime(2018, 11, 27, 0, 18, 58)

print(now)

print(datetime.now().timestamp())  # 1543249292.559023

print(datetime.fromtimestamp(1543249292.559023))

# Coordinated Universal Time
# Temps Universel Coordonné

print(datetime.utcfromtimestamp(1543249292.559023))

time = '1987-09-23 00:00:00'

print(datetime.strptime(time, '%Y-%m-%d %H:%M:%S'))

now = datetime.now()
print(now.strftime('%a, %b %d %H:%M'))

print(now)
print(now - timedelta(days=1))
```

`collections`

```python
#!/usr/bin/env python

from collections import namedtuple, deque, defaultdict, OrderedDict, Counter

Point = namedtuple('Point', ['x', 'y'])

p = Point(1, 2)

print(p.x)
print(p.y)

print(isinstance(p, Point))
print(isinstance(p, tuple))

Circle = namedtuple('Circle', ['x', 'y', 'r'])

c = Circle(1, 2, 3)

print(c.x)
print(c.y)
print(c.r)

print(c._asdict())

# c.x = 0

# deque

q = deque([1, 2, 3])

print(q.pop())
```



```
print(q.popleft())

q.appendleft(1)
print(q)
q.append(3)
print(q)


# defaultdict


def na():
    return 'N/A'


d = defaultdict(na)

# d = defaultdict(lambda: 'N/A')

d['key'] = 'value'
print(d['key'])

print(d['k'])

# OrderedDict

d = dict([(1, 'x'), (2, 'y'), (3, 'z')])

print(d)

d = OrderedDict([(1, 'x'), (2, 'y'), (3, 'z')])

print(d)
```

```
d[-1] = 'a'
d[-2] = 'b'
d[-3] = 'c'

print(d)

# Counter

counter = Counter()

for c in 'programming':
    counter[c] += 1

print(counter)

words = ['hello', 'world', 'nice', 'world']
counter = defaultdict(lambda: 0)
for word in words:
    counter[word] += 1

print(counter)
```

base64

```
#!/usr/bin/env python

import base64

s = b'Hello, World!'

print(base64.encodebytes(s))
print(base64.decodebytes(b'SGVsbG8sIFdvcmxkIQ==\n'))
```

hashlib

```
#!/usr/bin/env python

import hashlib

password = '123'

md5 = hashlib.md5()

md5.update(password.encode('utf-8'))

print(md5.hexdigest())

sha512 = hashlib.sha512()

sha512.update(password.encode('utf-8'))

print(sha512.hexdigest())
print(len(sha512.hexdigest()))
```

~~hmac~~

itertools

```
#!/usr/bin/env python

import itertools

numbers = itertools.count(1)

# for number in numbers:
    # print(number)

cycles = itertools.cycle('abc')

# for cs in cycles:
#     print(cs)

repeats = itertools.repeat('a', 10)

for rs in repeats:
    print(rs)
```

5. 第三方模块

`Pillow`

> Pillow

`PIL: Python Imaging Library`

```
#!/usr/bin/env python
```

```python
from PIL import Image, ImageFilter

# thumbnail

image = Image.open('test.png')
w, h = image.size
image.thumbnail((w // 2, h // 2))
image.save('thumbnail.png')

# blur
image = Image.open('test.png')
image_blur = im.filter(ImageFilter.BLUR)
image_blur.save('blur.png')
```

6. 作业

1. 使用 `Pillow` 模块，生成验证码

> CAPTCHA