4 Python快速面面观（下）

> AI领域中的Python开发 --- by 丁宁

@(SIGAI课程录制)

- 上节课：Python程序的原材料已初步介绍完毕
- 接下来：介绍如何对上节课中的原材料进行加工

> **说明**：本节课仅使用**SIGAI**在线编程的**terminal**以及**Text Editor**方式

条件判断

布尔变量的基本判断规则与方法

```
>>> not True
False
>>> not False
True
>>> True and False
False
>>> True or False
```

```
True
>>> True == False
False
>>> True != False
True
>>> True > False
True
>>> True >= False
True
>>> True < False
False
>>> True <= False
False
>>> True is False
False
>>> True is True
True
>>> False < True <= True
True
>>> False < True < True
False
```

数字的基本判断规则与方法

```
>>> 0 and 2
0
>>> 0 or -2
-2
>>> not 2
False
>>> not 0
True
>>> not -1
False
>>> -2 < 3 < 2
False
>>> 0 == False
True
>>> 1 == True
True
>>> 2 == True
False
>>> 0 is False
False
>>> 1 is True
False
```

Python入门小坑之逻辑判断

- `==` `!=` `>` `<` `>=` `<=`：计算用数值，结果用布尔
- `not`：计算用布尔，结果用布尔
- `is` 不计算，只判断
- `链式判断大小关系`：**结果与数学上保持一致**
- `and`：从左向右找 `0` 或者 `False`，找到则立即返回，未找到则返回最后一个
- `or`：从左向右找 `非0数字` 或者 `True`，找到则立即返回，未找到则返回最后一个

深入理解Python中的 `and`

```
>>> 1 and 2 and 3
3
>>> 1 and True and 3
3
>>> 1 and 2 and True
True
>>> 1 and 0 and 3
0
>>> 1 and False and 3
False
>>> 0 and False and 3
0
>>> False and 0 and 3
False
```

深入理解Python中的 `or`

```
>>> 3 or 2 or 1
3
>>> 0 or 2 or 1
2
>>> False or 2 or 0
2
>>> 0 or False or -3
-3
>>> 0 or False or 0
0
>>> 0 or False or False
False
```

if语句

从这一节开始，我们开始使用**SIGAI**在线编程的**Text Editor**模块

1. 进入**SIGAI在线编程**页面
2. 点击**terminal**，使用命令行创建一个名为 `test.py` 的Python脚本

```
sigai@8a5f47e78164:~$ ls
sharedata  workspace
sigai@8a5f47e78164:~$ cd workspace/
sigai@8a5f47e78164:~/workspace$ ls
sigai@8a5f47e78164:~/workspace$ touch test.py
sigai@8a5f47e78164:~/workspace$ ls
test.py
sigai@8a5f47e78164:~/workspace$
```

3. 此时，双击左侧目录栏中的`workspace`文件夹，会看到刚刚创建好的`test.py`
4. 双击`test.py`文件，进入**Text Editor**模式
5. 在脚本编辑器中输入如下代码：

```
age = input("Please input your age: ")
age = int(age)

if age >= 18:
    print('your age is', age)
    print('adult')
else:
    print('your age is', age)
    print('teenager')
```

6. 保存文本
7. 切换到**terminal**模式
8. 执行脚本：

```
sigai@8a5f47e78164:~/workspace$ python test.py
Please input your age: 10
your age is 10
teenager
sigai@8a5f47e78164:~/workspace$ python test.py
Please input your age: 20
your age is 20
adult
sigai@8a5f47e78164:~/workspace$ python test.py
Please input your age: -2
your age is -2.0
teenager
sigai@8a5f47e78164:~/workspace$ python test.py
Please input your age: 50.0
Traceback (most recent call last):
  File "test.py", line 2, in <module>
    age = int(age)
ValueError: invalid literal for int() with base 10: '50.0'
```

9. 将**age**的类型改为**float**类型，并增加正数的判断：

```
age = input("Please input your age: ")
age = float(age)

if age >= 18:
    print('your age is', age)
    print('adult')
elif age >= 0:
    print('your age is', age)
    print('teenager')
else:
    print('input error!')
```

10. 此时再次运行：

```
sigai@8a5f47e78164:~/workspace$ python test.py
Please input your age: 10.0
your age is 10.0
teenager
sigai@8a5f47e78164:~/workspace$ python test.py
Please input your age: 18.1
your age is 18.1
adult
sigai@8a5f47e78164:~/workspace$ python test.py
Please input your age: -2
input error!
sigai@8a5f47e78164:~/workspace$ python test.py
Please input your age: sigai
Traceback (most recent call last):
  File "test.py", line 2, in <module>
    age = float(age)
ValueError: could not convert string to float: 'sigai'
```

关于如何处理接下来的问题，留在工程化中异常处理的章节详细讲解

循环

for循环：逐个取出进行操作

```
sList = ['s', 'i', 'g', 'a', 'i']
s = ''

for x in sList:
    s += x

print(s)
```

```
sigai@8a5f47e78164:~/workspace$ python test.py
sigai
```

```
>>> list(range(5))
[0, 1, 2, 3, 4]
>>> list(range(1,5))
[1, 2, 3, 4]
>>> list(range(1,5,2))
[1, 3]
```

```
nList = list(range(101))
s = 0

for i in nList:
    s += i

print(s)
```

```
sigai@8a5f47e78164:~/workspace$ python test.py
5050
```

while循环：只要满足条件则持续操作

```
n = 1
s = 0
while n < 101:
    s += n
    n += 1

print(s)
```

break语句：满足条件则跳出循环

```
n = 1
s = 0
while True:
    s += n
    n += 1
    if n > 100:
        break

print(s)
```

continue语句：满足条件则直接进入下个循环

假如现在只想计算1-100之中偶数的和：

```
n = 0
s = 0
while True:
    n += 1
    if n % 2 == 1:
        continue
    if n > 100:
        break

    s += n

print(s)
```

函数

调用内置函数

```
>>> abs(-2)
2
>>> abs(3)
3
>>> int('-2')
-2
>>> str(-2)
'-2'
>>> float(-2)
-2.0
>>> int(-2.0)
-2
>>> bool(-2)
True
>>> int(True)
1
>>> max([5,3,6,4,7])
7
>>> min([5,3,6,4,7])
3
>>> my_func = max
>>> my_func([5,3,6,4,7])
7
```

自定义函数

可以先定义一个函数，而不去实现它

```
def my_max(a, b):
    pass

if __name__ == '__main__':
    assert my_max(1,2) == 2
    assert my_max(4,3) == 4
```

```
sigai@8a5f47e78164:~/workspace$ python test.py
Traceback (most recent call last):
  File "test.py", line 5, in <module>
    assert my_max(1,2) == 2
AssertionError
```

```
def my_max(a, b):
    return 2

if __name__ == '__main__':
    assert my_max(1,2) == 2
    assert my_max(4,3) == 4
```

```
sigai@8a5f47e78164:~/workspace$ python test.py
Traceback (most recent call last):
  File "test.py", line 6, in <module>
    assert my_max(4,3) == 4
AssertionError
```

```
def my_max(a, b):
    if a > b:
        return a
    else:
        return b

if __name__ == '__main__':
    assert my_max(1,2) == 2
    assert my_max(4,3) == 4
    print("OK")
```

```
sigai@8a5f47e78164:~/workspace$ python test.py
OK
```

函数可以返回多个值

```python
def my_sort(a, b):
    if a > b:
        return a, b
    else:
        return b, a

if __name__ == '__main__':
    assert my_sort(1,2) == (2, 1)
    assert my_sort(4,3) == (4, 3)
    print("OK")
```

```
sigai@8a5f47e78164:~/workspace$ python test.py
OK
```

函数的默认参数

```python
def my_max(a, b=0):
    if a > b:
        return a
    else:
        return b

if __name__ == '__main__':
    assert my_max(1,2) == 2
    assert my_max(4,3) == 4
    assert my_max(3) == 3
    assert my_max(-1) == 0
    print("OK")
```

Python入门小坑之默认参数的记忆性

> 默认参数在函数定义时已被计算并冻结，因此默认参数一般指向不变对象

```python
>>> def add_sigai(L = []):
...     L.append('sigai')
...     return L
...
>>> add_sigai([1,2,3])
[1, 2, 3, 'sigai']
>>> add_sigai()
['sigai']
>>> add_sigai()
['sigai', 'sigai']
>>> add_sigai()
['sigai', 'sigai', 'sigai']
```

上述例子只需将默认参数用 `None` 替换，然后判断输入是否为 `None` 即可

传入任意个数的参数

```
def calc_sum(*numbers):
    sum = 0
    for n in numbers:
        sum += n
    return sum

if __name__ == '__main__':
    test_func = calc_sum
    assert test_func() == 0
    assert test_func(3) == 3
    assert test_func(1,3,5) == 9
    assert test_func(-1,1) == 0
    print("OK")
```

Python中的面向对象

分析场景，抽象逻辑，设计Class，创建Instance

```
class Student(object):

    def __init__(self, name, score=-1):
        self.__name = name
        self.__score = score
        self.say_hi()

    def name(self):
        return self.__name

    def say_hi(self):
        if self.__score < 0:
            print("{}: Hi, my name is {}. I'm a new
student.".format(self.__name, self.__name))
        else:
            print("{}: Hi, my name is {}. My score is
{}.".format(self.__name, self.__name, self.__score))

    def get_score(self, teacher, score):
        self.score = score
        print("{}: teacher {} just gave me a {}".format(self.__name,
teacher, score))

class Teacher(object):

    def __init__(self, name):
        self.__name = name
        self.say_hi()
    def say_hi(self):
        print("{}: Hi, my name is {}. I'm a teacher at
SIGAI".format(self.__name, self.__name))

    def score(self, student, score):
        student.get_score(self.__name, score)
        print("{}: I just gave {} a {}".format(self.__name, student.name(),
score))

if __name__ == '__main__':
    studentA = Student("A")
    teacherB = Teacher("B")
    teacherB.score(studentA, 80)
```

关于继承，多态，元类，属性装饰器等内容在后续课程中详细讲解

Python代码的组织-模块

- 包 -> 模块 -> 类或功能函数
- 每个包里面都含有一个 `__init__.py` 文件，而且必须存在，用以区分普通目录还是包
- 创建包或者模块的时候，不可与系统自带的包或者模块重名

使用现成的模块封装一个自己的模块

```
'first module'

__author__ = 'sigai'

import sys

def say_hi():
    args = sys.argv
    if len(args)==1:
        print('Welcome to SIGAI online programming platform!')
    elif len(args)==2:
        print('Hi, %s, Welcome to SIGAI online programming platform!!' %
args[1])
    else:
        print('Too many arguments!')

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

看一下结果：

```
sigai@aa15df80b209:~/workspace$ python test.py
Welcome to SIGAI online programming platform!

sigai@aa15df80b209:~/workspace$ python test.py Tom
Hi, Tom, Welcome to SIGAI online programming platform!!

sigai@aa15df80b209:~/workspace$ python test.py Tom Jack
Too many arguments!
```

包管理工具pip

首先查看一下**在线编程平台**上的**pip**版本和位置

```
sigai@8a5f47e78164:~/workspace$ pip -V
pip 10.0.1 from /usr/local/lib/python3.5/dist-packages/pip (python 3.5)
sigai@8a5f47e78164:~/workspace$ which pip
/usr/local/bin/pip
```

然后查看系统中已安装的第三方包及其版本

```
sigai@8a5f47e78164:~/workspace$ pip list
Package                   Version
-------------------- ---------
absl-py                   0.2.2
astor                     0.7.0
backcall                  0.1.0
bleach                    1.5.0
...
requests                  2.18.4
scikit-image              0.14.0
scikit-learn              0.19.1
scipy                     1.1.0
...
tensorboard               1.8.0
tensorflow-gpu            1.8.0
...
torch                     0.4.0
torchvision               0.2.1
tornado                   5.0.2
...
Werkzeug                  0.14.1
wheel                     0.31.1
You are using pip version 10.0.1, however version 18.0 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

升级**pip**到18.0版本

```
sigai@8a5f47e78164:~/workspace$ pip install --upgrade pip
```

```
/usr/local/lib/python3.5/dist-
packages/pip/_internal/commands/install.py:199: UserWarning: Disabling all
use of wheels due to the use of --build-options / --global-options / --
install-options.
  cmdoptions.check_install_build_global(options)
Looking in indexes: https://pypi.douban.com/simple
Collecting pip
  Downloading
https://pypi.doubanio.com/packages/69/81/52b68d0a4de760a2f1979b0931ba788920
2f302072cc7a0d614211bc7579/pip-18.0.tar.gz (1.2MB)
    100% |████████████████████████████████| 1.3MB 31.0MB/s
  Installing build dependencies ... done
Skipping bdist_wheel for pip, due to binaries being disabled for it.
Installing collected packages: pip
  Found existing installation: pip 10.0.1
    Uninstalling pip-10.0.1:
Could not install packages due to an EnvironmentError: [Errno 13]
Permission denied: '/usr/local/bin/pip'
Consider using the `--user` option or check the permissions.

You are using pip version 10.0.1, however version 18.0 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
sigai@8a5f47e78164:~/workspace$ pip install --upgrade pip --user
/usr/local/lib/python3.5/dist-
packages/pip/_internal/commands/install.py:199: UserWarning: Disabling all
use of wheels due to the use of --build-options / --global-options / --
install-options.
  cmdoptions.check_install_build_global(options)
Looking in indexes: https://pypi.douban.com/simple
Collecting pip
  Downloading
https://pypi.doubanio.com/packages/69/81/52b68d0a4de760a2f1979b0931ba788920
2f302072cc7a0d614211bc7579/pip-18.0.tar.gz (1.2MB)
    100% |████████████████████████████████| 1.3MB 25.0MB/s
  Installing build dependencies ... done
Skipping bdist_wheel for pip, due to binaries being disabled for it.
Installing collected packages: pip
  Running setup.py install for pip ... done
Successfully installed pip-18.0
sigai@8a5f47e78164:~/workspace$ pip -V
pip 18.0 from /home/sigai/.local/lib/python3.5/site-packages/pip (python
3.5)
```

使用`pip install 包名称 (-U/--user)`安装包

```
sigai@8a5f47e78164:~/workspace$ pip install selenium
/home/sigai/.local/lib/python3.5/site-
packages/pip/_internal/commands/install.py:206: UserWarning: Disabling all
use of wheels due to the use of --build-options / --global-options / --
install-options.
  cmdoptions.check_install_build_global(options)
Looking in indexes: https://pypi.douban.com/simple
Collecting selenium
  Downloading
https://pypi.doubanio.com/packages/6d/4b/30b28589f2b6051b04d6f8014537749dc0
8fa787a5569cebb33e892d34d3/selenium-3.13.0.tar.gz (852kB)
    100% |████████████████████████████████| 860kB 331kB/s
Skipping bdist_wheel for selenium, due to binaries being disabled for it.
Installing collected packages: selenium
  Running setup.py install for selenium ... done
Successfully installed selenium
```

Python入门小坑之解释器默认搜索路径

sys.path

```
>>> import sys
>>> print('\n'.join(sys.path))

/usr/lib/python35.zip
/usr/lib/python3.5
/usr/lib/python3.5/plat-x86_64-linux-gnu
/usr/lib/python3.5/lib-dynload
/home/sigai/.local/lib/python3.5/site-packages
/usr/local/lib/python3.5/dist-packages
/usr/lib/python3/dist-packages
```

```
sigai@8a5f47e78164:~/workspace$ pwd
/home/sigai/workspace
```

```
>>> sys.path.append('/home/sigai/workspace')
>>> print('\n'.join(sys.path))

/usr/lib/python35.zip
/usr/lib/python3.5
/usr/lib/python3.5/plat-x86_64-linux-gnu
/usr/lib/python3.5/lib-dynload
/home/sigai/.local/lib/python3.5/site-packages
/usr/local/lib/python3.5/dist-packages
/usr/lib/python3/dist-packages
/home/sigai/workspace
```