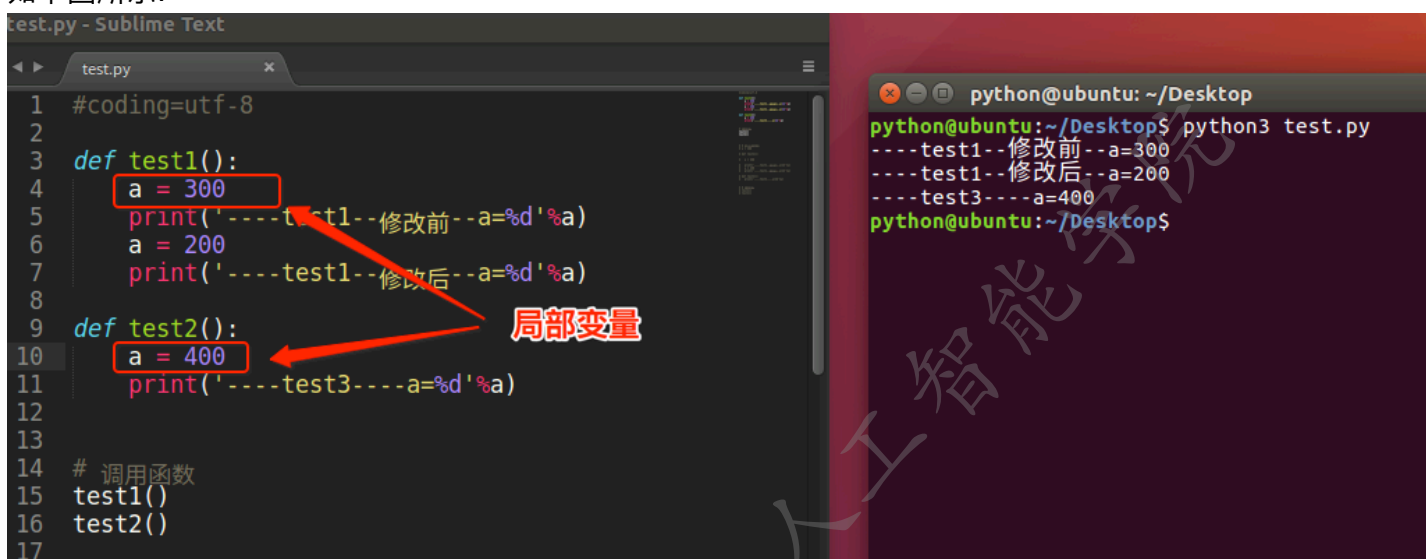


## 局部变量

- 局部变量，就是在函数内部定义的变量
- 其作用范围是这个函数内部，即只能在这个函数中使用，在函数的外部是不能使用的
- 因为其作用范围只是在自己的函数内部，所以不同的函数可以定义相同名字的局部变量（打个比方，把你、我是当做成函数，把局部变量理解为每个人手里的手机，你可有个iPhone8，我当然也可以有个iPhone8了，互不相关）
- 局部变量的作用，为了临时保存数据需要在函数中定义变量来进行存储
- 当函数调用时，局部变量被创建，当函数调用完成后这个变量就不能够使用了

如下图所示：



```
test.py - Sublime Text
1 #coding=utf-8
2
3 def test1():
4     a = 300
5     print('----test1--修改前--a=%d'%a)
6     a = 200
7     print('----test1--修改后--a=%d'%a)
8
9 def test2():
10    a = 400
11    print('----test3----a=%d'%a)
12
13
14 # 调用函数
15 test1()
16 test2()
17
```

```
python@ubuntu: ~/Desktop
python@ubuntu:~/Desktop$ python3 test.py
----test1--修改前--a=300
----test1--修改后--a=200
----test3----a=400
python@ubuntu:~/Desktop$
```

## 全局变量

如果一个变量，既能在一个函数中使用，也能在其他的函数中使用，这样的变量就是全局变量

打个比方：有2个兄弟 各自都有手机，各自有自己的小秘密在手机里，不让另外一方使用（可以理解为局部变量）；但是家里的电话是2个兄弟都可以随便使用的（可以理解为全局变量）

```
# 定义全局变量
a = 100

def test1():
    print(a) # 虽然没有定义变量a但是依然可以获取其数据

def test2():
    print(a) # 虽然没有定义变量a但是依然可以获取其数据

# 调用函数
test1()
test2()
```

运行结果:

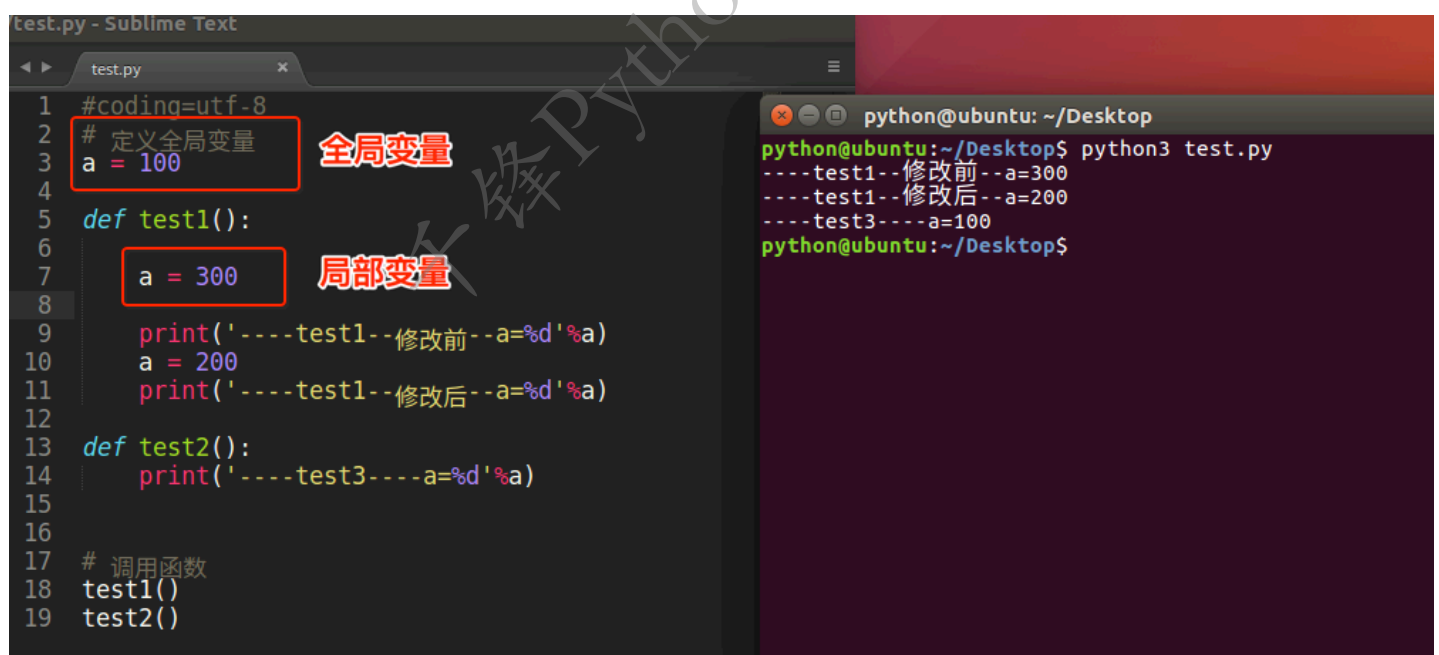
```
python@ubuntu:~/Desktop$ python3 test.py
100
100
```

## 总结1:

- 在函数外边定义的变量叫做 全局变量
- 全局变量能够在所有的函数中进行访问

## 全局变量和局部变量名字相同问题

看如下代码:



```
test.py - Sublime Text
1 #coding=utf-8
2 # 定义全局变量
3 a = 100
4
5 def test1():
6     a = 300
7
8     print('----test1--修改前--a=%d'%a)
9     a = 200
10    print('----test1--修改后--a=%d'%a)
11
12 def test2():
13     print('----test3----a=%d'%a)
14
15
16
17 # 调用函数
18 test1()
19 test2()
```

```
python@ubuntu: ~/Desktop
python@ubuntu:~/Desktop$ python3 test.py
----test1--修改前--a=300
----test1--修改后--a=200
----test3----a=100
python@ubuntu:~/Desktop$
```

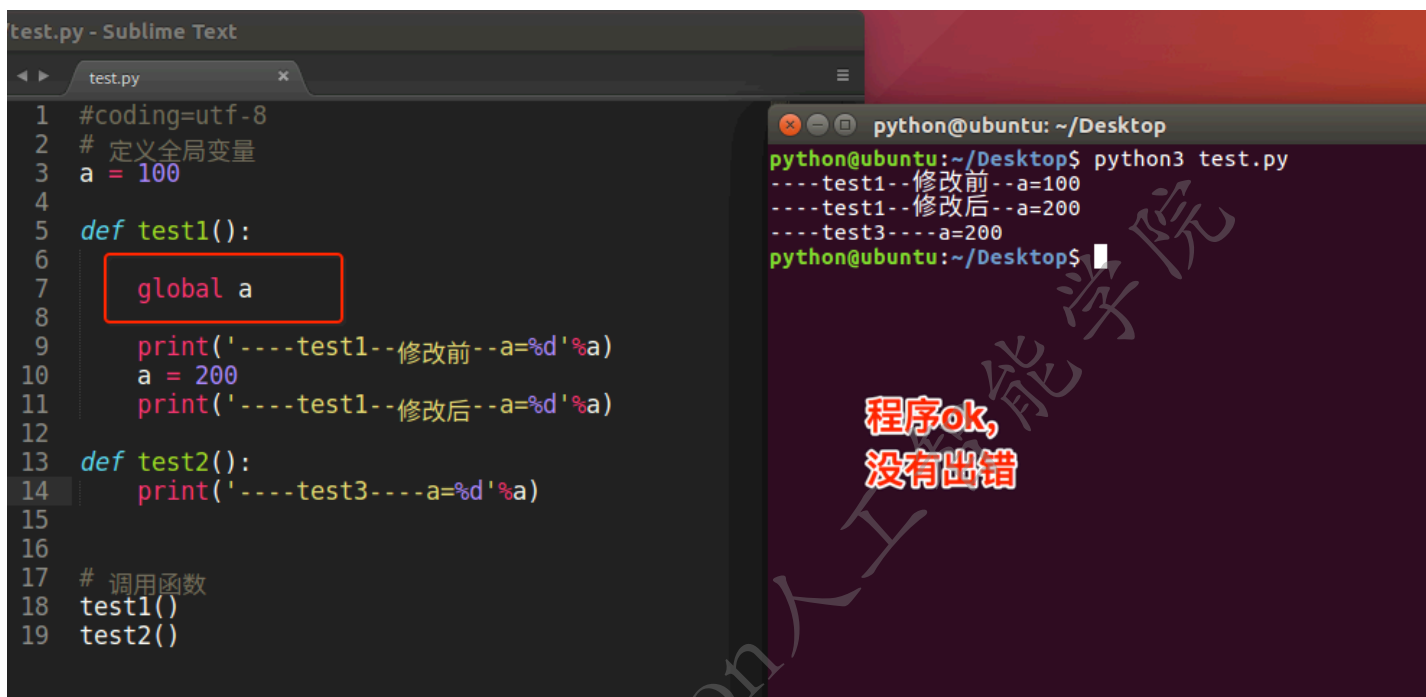
## 总结2:

- 当函数内出现局部变量和全局变量相同名字时，函数内部中的 `变量名 = 数据` 此时理解为定义了一个局部变量，而不是修改全局变量的值

## 修改全局变量

函数中进行使用时可否进行修改呢？

代码如下：



```
test.py - Sublime Text
1 #coding=utf-8
2 # 定义全局变量
3 a = 100
4
5 def test1():
6     global a
7
8     print('----test1--修改前--a=%d'%a)
9     a = 200
10    print('----test1--修改后--a=%d'%a)
11
12 def test2():
13    print('----test3----a=%d'%a)
14
15
16
17 # 调用函数
18 test1()
19 test2()
```

```
python@ubuntu: ~/Desktop
python@ubuntu:~/Desktop$ python3 test.py
----test1--修改前--a=100
----test1--修改后--a=200
----test3----a=200
python@ubuntu:~/Desktop$
```

程序ok,  
没有出错

## 总结3:

- 如果在函数中出现 `global 全局变量的名字` 那么这个函数中即使出现和全局变量名相同的 `变量名 = 数据` 也理解为对全局变量进行修改，而不是定义局部变量
- 如果在一个函数中需要对多个全局变量进行修改，那么可以一次性全部声明，也可以分开声明

```
# 可以使用一次global对多个全局变量进行声明
global a, b
# 还可以用多次global声明都是可以的
# global a
# global b
```

## 查看所有的全局变量和局部变量

Python提供了两个内置函数globals()和locals()可以用来查看所有的全局变量和局部变量。

```
def test():
    a = 100
    b = 40
    print(locals()) # {'a': 100, 'b': 40}

test()

x = 'good'
y = True
print(globals()) # {'__name__': '__main__', '__doc__': None, '__package__': None,
 '__loader__': <_frozen_importlib_external.SourceFileLoader object at 0x101710630>
, '__spec__': None, '__annotations__': {}, '__builtins__': <module 'builtins' (built-in)>,
 '__file__': '/Users/jiangwei/Desktop/Test/test.py', '__cached__': None, 'test': <function test at 0x101695268>, 'x': 'good', 'y': True}
```

千锋Python人工智能学院