python in action

Lesson 5 - 分支控制和函数初步

凤凰山 github.com/gxtrobot/pyinaction

2019-11

Lesson5 - 分支控制和函数初步

课程内容

今天的课程将介绍 Python 的逻辑操作符，以及分支控制语句 if 的用法，包括 if, else, elif 的几种形式. 然后介绍函数的初步知识，如何定义函数.

课程目标

- 了解并使用 Python 的逻辑操作符
- 了解使用 if 语句
- 了解使用 if...else...语句
- 了解使用 if...elif...else 语句
- 了解如何定义函数
- 编写一个函数来对一个列表的所有值求和

逻辑操作符

用来组合条件 (bool 值) 的操作符

- and ， or， not
- 接收一个或 2 个 bool 值, 返回一个 bool 值
- A and B， 两个条件须同时成立， 结果为真， 否则为假
- A or B， 两个条件有一个成立， 结果为真， 否则为假
- not A， A 为真返回假， A 为假返回真
- and， or 具有短路特性， 意思就是如果只看第一个值就能确定最终结果的时候直接返回， 不会看第二个值

IDLE 玩一玩

在 IDLE 试着使用逻辑操作符，并查看结果

```
a = 5
b = 3
c = 10

a > b and c > b
a > b or b > c
not a > b

b > a and print(1)
b < a and print(1)

c > a or print(1)
c < a or print(1)
```

使用 if 语句

单独使用 if

可能执行一个语句块或跳过

```
score = 70
if score >= 60:
    print('passed')
```

- if 后接一个条件判断 bool 值表达式
- 当条件为真时，语句块就执行，继续执行语句块后语句
- 当条件为假时，跳过语句块，继续执行语句块后语句
- if 语句第一行以':'结束，后面跟缩进语句块

使用 if 语句

使用 if...else...

两个语句块，二选一执行

```
score = 70
if score >= 60:
    print('passed')
else:
    print('not passed')
```

- if 后跟一个条件判断 bool 值表达式，然后接一个语句块
- else 后直接跟一个语句块
- if 后条件为真，执行 if 语句块，条件为假，执行 else 语句块
- 两个语句块必然有一个会执行

使用 if 语句

使用 if...elif...else

多个语句块，N 选一执行

```
score = 80
if score >= 90:
    print('very good ')
elif score >= 60:
    print('passed')
else:
    print('not passed')
```

- if 后跟一个条件判断 bool 值表达式，然后接一个语句块
- 然后可以跟若干个 elif 语句，每个都接一个 bool 表达式和一个语句块
- 最后可以接一个 else 语句，没有 bool 值表达式，直接跟一个语句块
- 从最开始的 if 开始，按顺序判断每个 bool 表达式，如果为真就执行相应语句块，然后结束整个 if 控制语句，继续后面语句执行
- 如果没有一个条件为真，则执行 else 对应语句块

IDLE 玩一玩

测 试 以 上 的 各 种 `if` 语 句

定义函数

```
def getsum(score_list):
    pass
```

- def 语句定义一个函数
- def 后接函数名，函数名需要满足 python 的命名规则，且不能和系统函数同名，否则会覆盖系统函数
- 函数名后用 ()，圆括号内为逗号分隔的参数列表
- 下面跟一个缩进语句块，代表函数具体执行内容
- pass 可以代表一个空语句块，可以用来临时占位使用，不会报错
- 如果需要返回值，可以用 return 语句后接返回的值，return 执行后立即结束函数并返回值
- 如果没有 return 语句，函数默认返回 None 值
- 调用函数使用函数名 ()，圆括号内用逗号分隔要传输的参数列表

IDLE 玩一玩

```
a = [1,2,3,4]
def getsum(score_list):
    total = 0
    for score in score_list:
        total = total + score
    return total

total = getsum(a)
print(total)

def print_list(score_list):
    for score in score_list:
        print('score:', score)

res = print_list(a)
print('res:', res)
```

课后练习

- 编写一个函数，getavg(score_list)，求列表的平均值
- 编写一个函数，getrate(score)，求分数的分级（>=90 为 A，75 到 90 为 B，60 到 75 为 C，低于 60 为 D）

问题