

流程控制

条件分支流程

- if-else结构

```
1. if
    if 布尔表达式:
        代码

2. if-else
    if 布尔表达式:
        代码
    else:
        代码

    代码块

3. if-elif-else:
    if 布尔表达式:
        代码
    elif 布尔表达式:
        代码
    ...
    else:
        代码

4. if的嵌套使用
    每一个结构下（代码块）都可以嵌套if的任何结构
    可以嵌套多层
```

断言

- assert

```
assert : 关键字
语法 :
1. assert 布尔表达式
    如果布尔表达式为真，程序正常执行，不受影响
    如果布尔表达式为假，程序会抛出异常---AssertionError
```

循环流程

- while循环

重复执行某些代码

语法：

1.

```
while 布尔表达式：
    循环体
```

2. 注意：

避免无用的死循环（服务器）
死循环比一定都是坏的---看需求
控制好判断条件（布尔表达式）

3. 适用于不知道循环次数

- for循环

也称为计数循环

1. 语法

```
for 目标 in 可迭代对象：
    循环体
```

目标：变量

可迭代对象：字符串，列表

2. 可以调用函数

`range()`

- `range()`

1. `range(stop)`

2. `range([start], stop, [step])`

start:默认0开始

step:步长，默认是1

```
##for i in 'hello world':
```

```
##    print(i)
```

```
##
```

```
##for i in [1,2,3]:
```

```
##    print(i)
```

```
##for i in range(10):
```

```
##    print(i)
```

```
##
```

```
##for i in range(3,11,3):
```

```
##    print(i)
```

```
##
```

```
##for i in range(10,1,-1):
```

```
##    print(i)
```

- break

跳出当前循环，跳出后后面的循环不再执行

```
for i in range(10):
    if i==3:
        break
    else:
        print(i)
```

- continue

跳过本次循环，跳过后，后面的循环继续执行

```
for i in range(10):
    if i==3:
        continue
    else:
        print(i)
```

- 循环嵌套

循环之中还可以有循环

```
for j in range(3): # 有3行---行数
    for i in range(3): # 第n行---有3列---列数
        print('* ',end='')
    print()
```

```
# 边长为3的正方形
for i in range(3):
    for j in range(3):
        print('*',end=' ')
    print()

# 打印三角形
for i in range(1,4): # 1,2,3
    for j in range(i):
        print('*',end=' ')
    print()
```