

Loop

Python 基础入门

回顾

捕捉 异常

完善 代码

你 了解 我吗

生肖 换算

条件 很重要！



循环

课堂上 老师 问

100 个 5 相加，和是多少

从 1 加到 100，和是多少

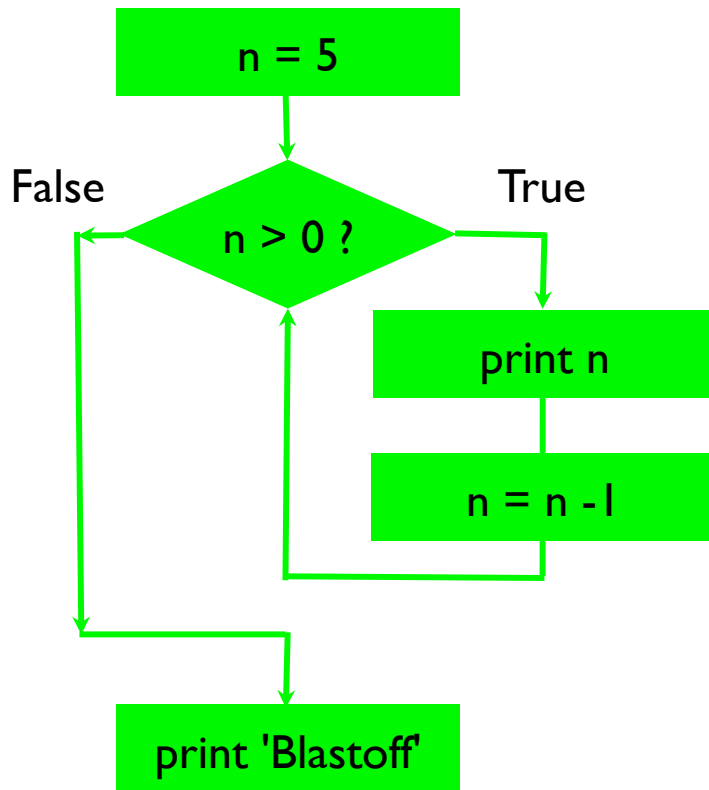
编程中，对于一些 重复类似 的操作
使用 循环 可以大大提高编程的 效率



while 循环

条件 成立 则循环

- ✓ 进行 初始化
- ✓ 判断条件 是否成立
- ✓ 若 成立 则执行循环中的代码
- ✓ 若 不成立 则结束循环



while 循环

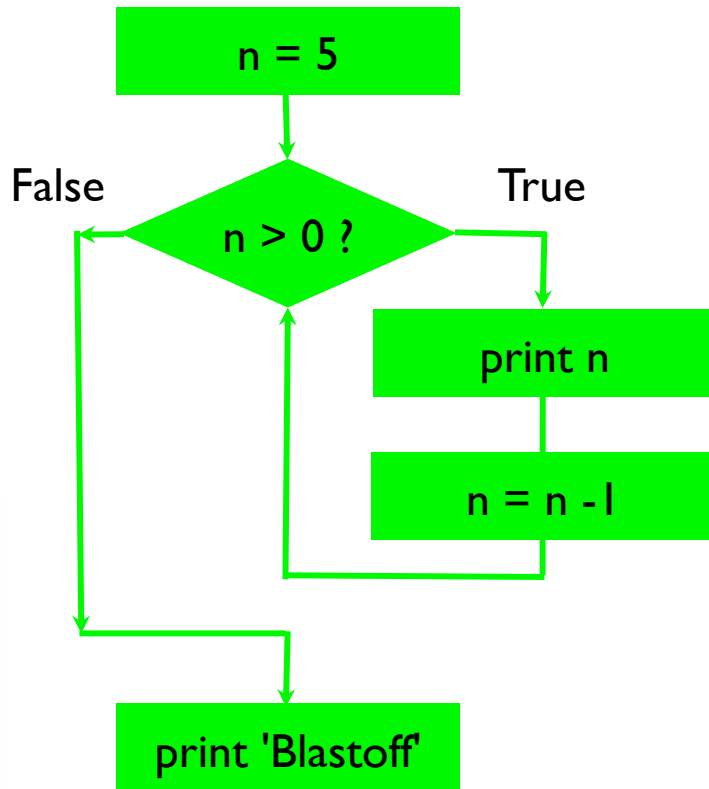
条件 成立 则循环

```
n = 5
```

```
while n > 0:  
    print(n)  
    n = n - 1
```

```
print("Blastoff!")  
print(n)
```

```
5  
4  
3  
2  
1  
Blastoff!  
0
```



■ 加点延时

使用 `time` 模块

`time` 模块提供了和 时间 相关的操作

使用 `import` 加载模块

`sleep()` 可以让程序 停止 一段时间

可以设置 `int` 或 `float`, 单位为 秒

```
import time

n = 5
while n > 0:
    print(n)
    time.sleep(1)
    n = n - 1

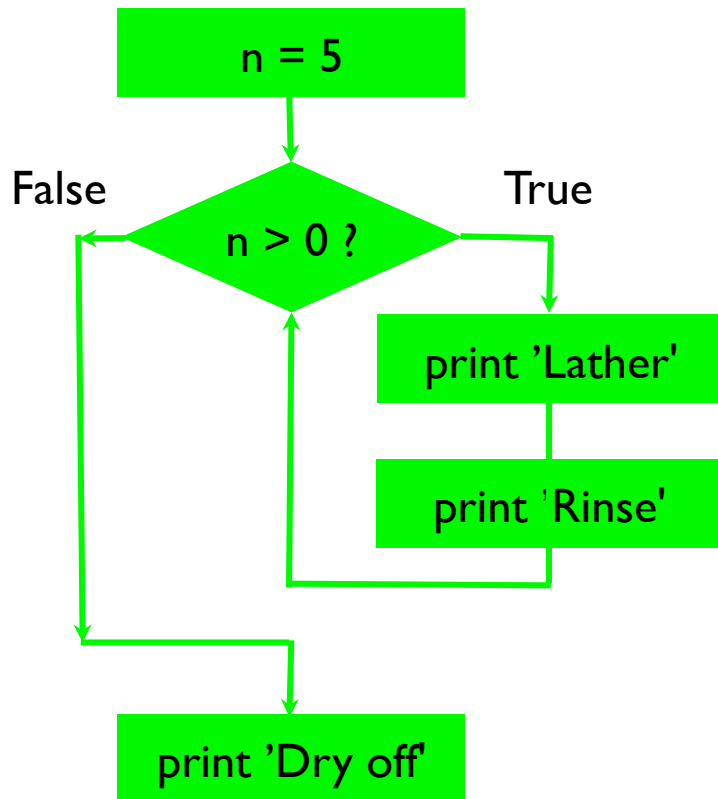
print("Blastoff")
print(n)
```

死循环

以下循环 有何问题

```
n = 5  
  
while n > 0:  
    print("Lather")  
    print("Rinse")  
  
print("Dry off!")
```

注意：在循环中，修改 条件变量

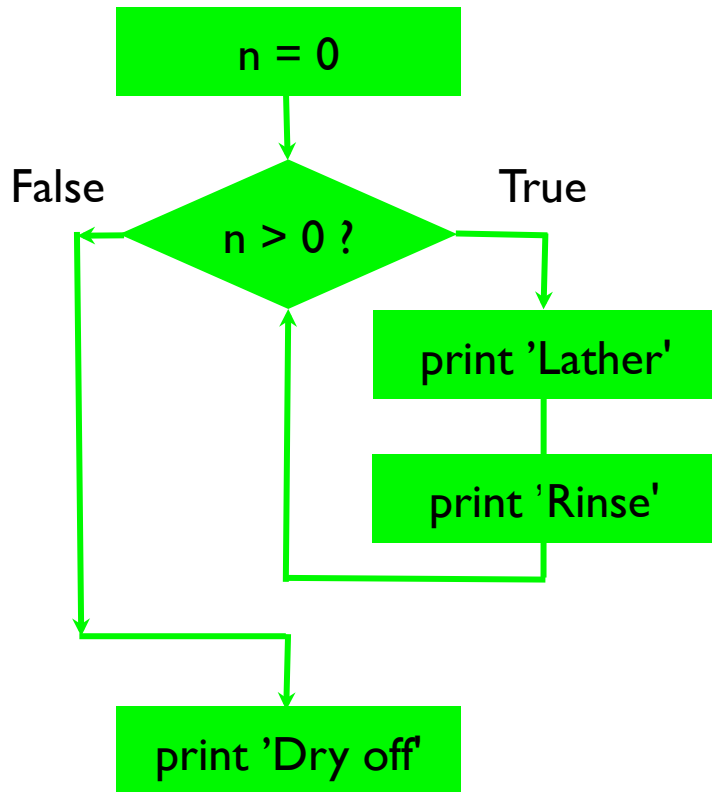


■ 另一个循环

以下循环 会执行 吗

```
n = 0  
  
while n > 0:  
    print("Lather")  
    print("Rinse")  
  
print("Dry off!")
```

注意：合理设置条件变量的 初始值



■ 循环控制

使用 `break` 结束 整个循环

```
while True:
    line = input('> ')
    print(line)
print('out of the while')
```

```
while True:
    line = input('> ')
    if line == 'quit':
        break
    print(line)
print('out of the while')
```

■ 循环控制

使用 `continue` 结束 本轮循环

不再执行 循环中的 后续代码

并回到 条件判断 处

```
while True:
    line = input('> ')
    if line == 'quit':
        break
    if line == 'sleep':
        continue
    print(line)

print('out of the while')
```

练习

1 计算 累加和

- ✓ 使用 `input()` 输入一个 正整数
- ✓ 计算从 1 到 该整数 之间，所有整数的 和
- ✓ 包括 1 和 该整数
- ✓ 例如输入 5，则输出 $1 + 2 + 3 + 4 + 5$ 的计算结果



练习

2 计算 阶乘

- ✓ 使用 `input()` 输入一个 正整数
- ✓ 输出该整数的 阶乘
- ✓ 例如输入 5，则输出 $1 \times 2 \times 3 \times 4 \times 5$ 的计算结果



随机整数

使用 `random` 产生 随机数

`randrange(start, stop, step)`

`start`: 最小值, 包含在范围内

`stop`: 最大值, 不包含在范围内

`step`: 递增的步长

```
>>> import random
>>> x = random.randrange(100)
>>> print(x)
4
>>> y = random.randrange(100, 200)
>>> print(y)
154
>>> z = random.randrange(0, 4, 2)
>>> print(z)
2
>>>
```

■ 作业

1 猜数 游戏

- ✓ 自由设定一个 范围，例如 0 至 100
- ✓ 在 设定范围 中随机产生一个 整数
- ✓ 让玩家 不断地 来猜这个数
 - 如果 猜大了，则提示猜大了
 - 如果 猜小了，则提示猜小了
 - 如果 猜对了，则提示 正确，并输出 猜的次数
- ✓ 挑战*：猜中之后，让玩家选择是否 重来一局

■ 作业

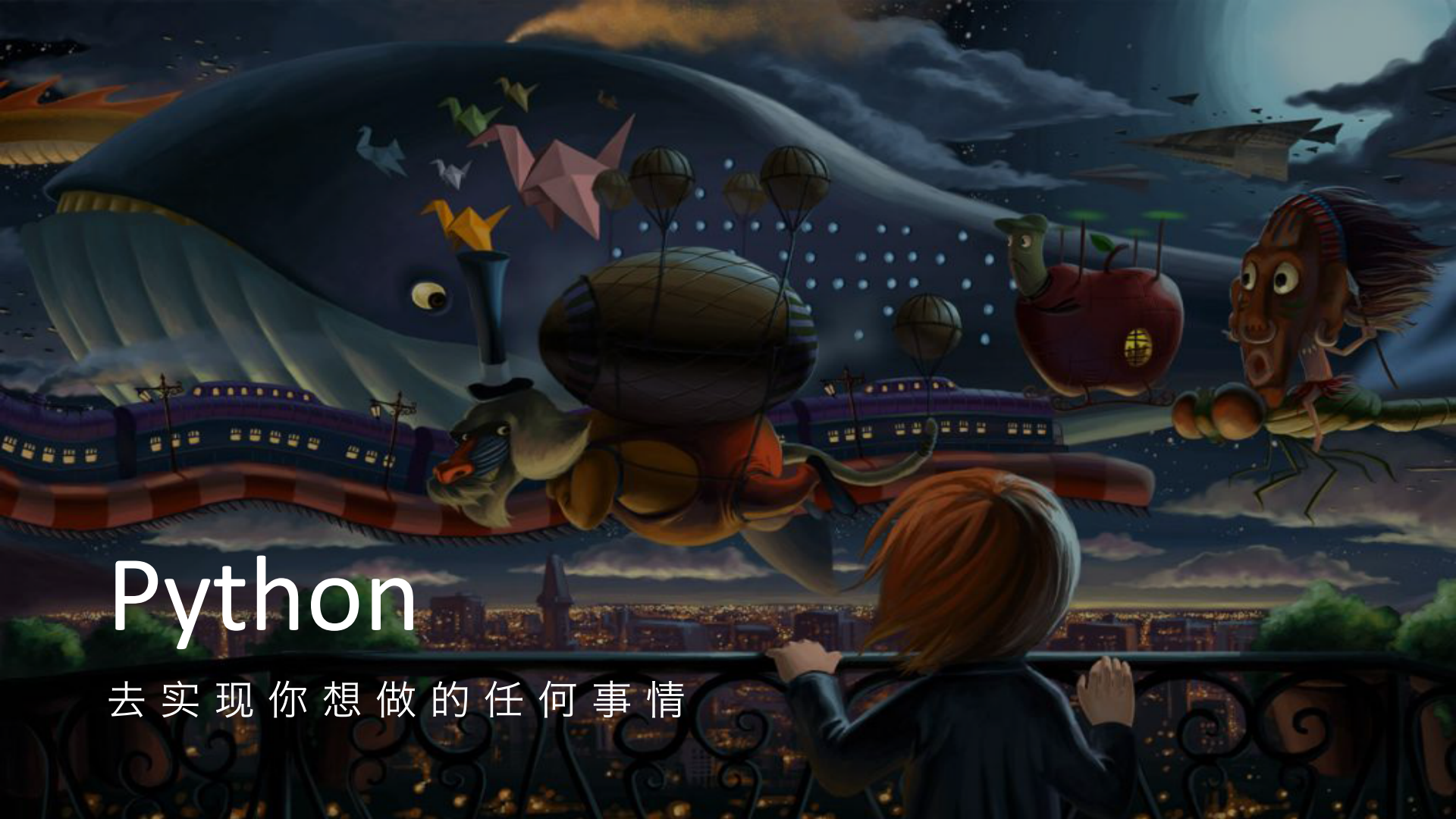
2 水仙花数

- ✓ 找到所有的 水仙花数
- ✓ 水仙花数是 三位数
- ✓ 水仙花数 每位上数字 的 三次方之和 等于它 自身
- ✓ 例如, $153 = 1^3 + 5^3 + 3^3$

■ 作业

3 斐波那契 数列

- ✓ 斐波那契数列是这样的 一系列数
- ✓ $1, 1, 2, 3, 5, 8, 13, 21, \dots$
- ✓ 即 后一个 数字, 等于 前两个 数字的 和
- ✓ 输出斐波那契数列的 前30个 数字



Python

去实现你想做的任何事情