
Condition
Python 基础入门

回顾

变量的 类型

获取 类型

类型 转换

用户 输入

格式化

注释

条件

课堂上 老师 说

如果 待会不下雨，体育课就正常上

否则 就呆在教室里，上自习

编程中，往往需要考虑到 多种 可能的情况

一条路走到底的编程是 不完善 的

使用条件

通过 条件 判断

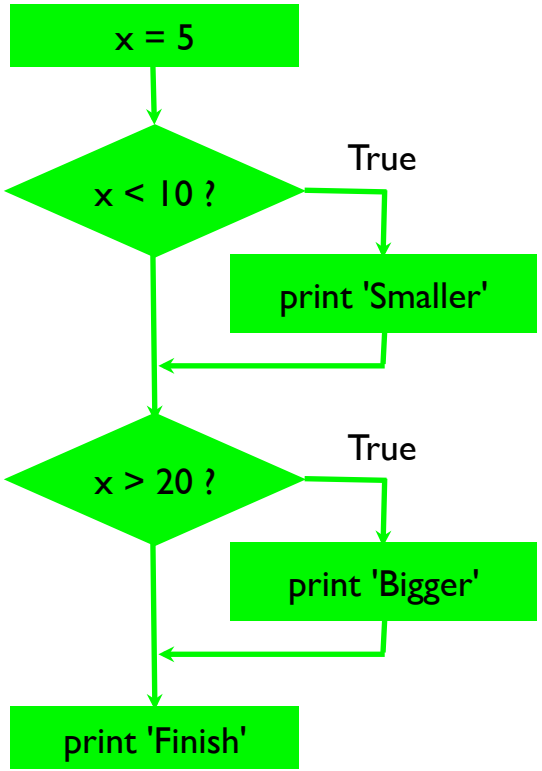
如果满足 某个条件，则运行 对应的代码

```
x = 5

if x < 10:
    print("Smaller")

if x > 20:
    print("Bigger")

print("Finish")
```



布尔类型

新的类型 布尔

布尔 类型只有两种取值：True 和 False

True 表示 条件成立，False 表示 不成立

已学的类型：int、float、string、bool

注意 大小写！

```
>>> passed = True
>>> type(passed)
<class 'bool'>
>>> passed = False
>>> type(passed)
<class 'bool'>
```

```
>>> passed = TRUE
Traceback (most recent call last):
  File "<pyshell#316>", line 1, in <module>
    passed = TRUE
NameError: name 'TRUE' is not defined
```

比较运算符

通过 比较 得出 真假

布尔表达式 使用 比较运算符

并产生一个 True 或 False

```
>>> print(1 > 2)
False
>>> print(2 > 1)
True
```

<	小于
<=	小于或等于
==	等于
>=	大于或等于
>	大于
!=	不等于

比较运算

对变量进行 比较

So easy

```
>>> x = 5
>>> print(x == 5)
True
>>> print(x > 4)
True
>>> print(x >= 5)
True
>>> print(x < 6)
True
>>> print(x <= 5)
True
>>> print(x != 6)
True
```

和 算术运算符 一样

比较运算符 不会改变 变量的值

```
>>> print(x)
5
```

比较 + 条件

将比较 结果 作为 条件

比较结果为 True 则 条件成立

并执行对应的 代码

比较结果为 False 则 条件不成立

不执行对应的 代码

```
x = 5
if x == 5:
    print("Equal 5")
if x > 4:
    print("Greater than 4")
if x >= 5:
    print("Greater than or Equal 5")
if x < 6:
    print("Less than 6")
if x <= 5:
    print("Less than or Equal 5")
if x != 6:
    print("Not Equal 6")
```

缩进

缩进 决定了 代码结构

一层缩进即一个 tab，或四个 空格

在 if 的冒号之后，增加 一层缩进
表示代码块的 开始

保持 缩进，表示代码块的 继续

空行 将被 忽略，不会影响缩进

减少 一层缩进，表示代码块的 结束

```
x = 5
print("Before 5")
if x == 5:
    print("Is 5")
    print("Is Still 5")

    print("Third 5")
print("After 5")

print("Before 6")
if x == 6:
    print("Is 6")
    print("Is Still 6")
    print("Third 6")
print("After 6")
```

思考

这些代码能 正确运行 吗

```
x = 5
    print(x)
```

```
x = 5
print("Before 5")
if x == 5:
    print("Is 5")

    print("Is Still 5")
```

```
x = 5
print("Before 5")
if x == 5:
    print("Is 5")
        print("Is Still 5")
```

```
x = 5
print("Before 5")
if x == 5:
    print("Is 5")
    print("Is Still 5")

    print("Third 5")
print("After 5")

print("Before 6")
if x == 6:
    print("Is 6")
    print("Is Still 6")
print("Third 6")
print("After 6")
```

想象力

```
x = 5
print("Before 5")
if x == 5:
    print("Is 5")
    print("Is Still 5")

    print("Third 5")
print("After 5")

print("Before 6")
if x == 6:
    print("Is 6")
    print("Is Still 6")
    print("Third 6")
print("After 6")
```

```
x = 5
print("Before 5")
if x == 5:
    print("Is 5")
    print("Is Still 5")

    print("Third 5")
print("After 5")

print("Before 6")
if x == 6:
    print("Is 6")
    print("Is Still 6")
print("Third 6")
print("After 6")
```

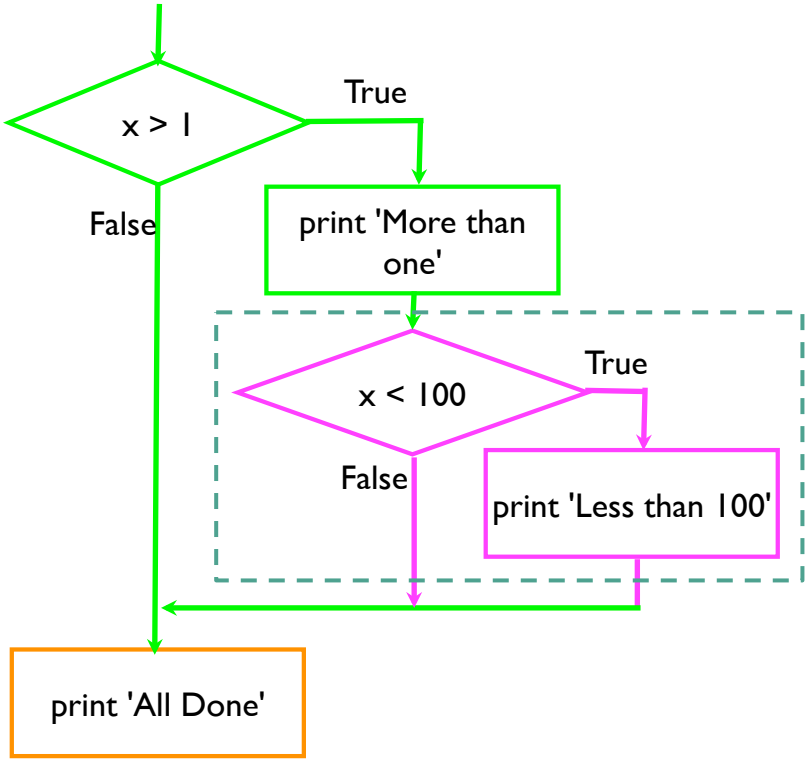
嵌套条件

```
x = 42

if x > 1:
    print("More than one")
    if x < 100:
        print("Less than 100")

print("All done")
```

```
More than one
Less than 100
All done
```



双路判断

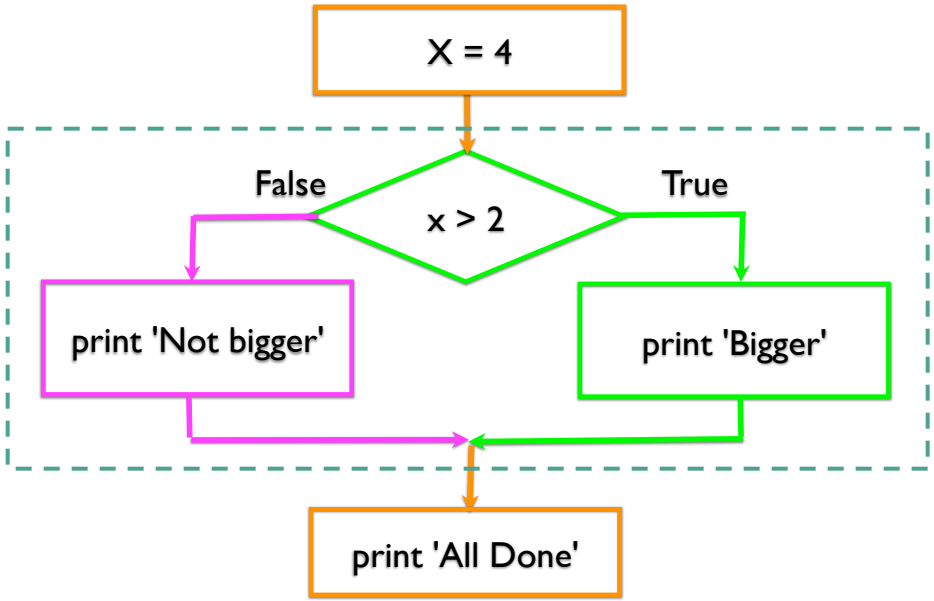
使用 else

处理 条件不成立 的情况

```
x = 4

if x > 2:
    print("Bigger")
else:
    print("Not Bigger")

print("All done")
```



多路判断

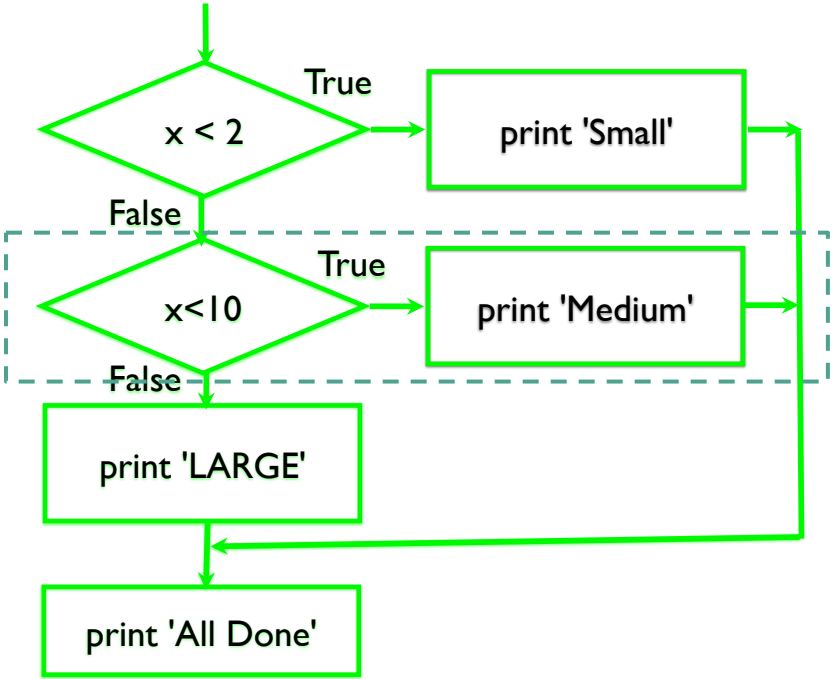
使用 elif

第一个 条件不成立，继续判断 其它条件

```
x = 4

if x < 2:
    print("Small")
elif x < 10:
    print("Medium")
else:
    print("Large")

print("All done")
```



多路判断

哪些 print() 永不执行

好的 条件组合，应当依次 不重叠地 涵盖 所有的 情况

```
if x < 2:
    print("Smaller than 2")
elif x >= 2:
    print("Two or more")
else:
    print("Something else")
```

```
if x < 2:
    print("Smaller than 2")
elif x < 20:
    print("Smaller than 20")
elif x < 10:
    print("Smaller than 10")
else:
    print("Something else")
```

练习

评分等级

- ✓ 使用 input() 输入一个 整数
- ✓ 根据以下 评分规则，print() 对应的 评分等级
  - 90 ~ 100: Excellent
  - 80 ~ 89: Good
  - 70 ~ 79: Medium
  - 60 ~ 69: Passed
  - 0 ~ 59: Failed
  - other: Wrong Score
- ✓ 使用 Python脚本 存储代码

作业

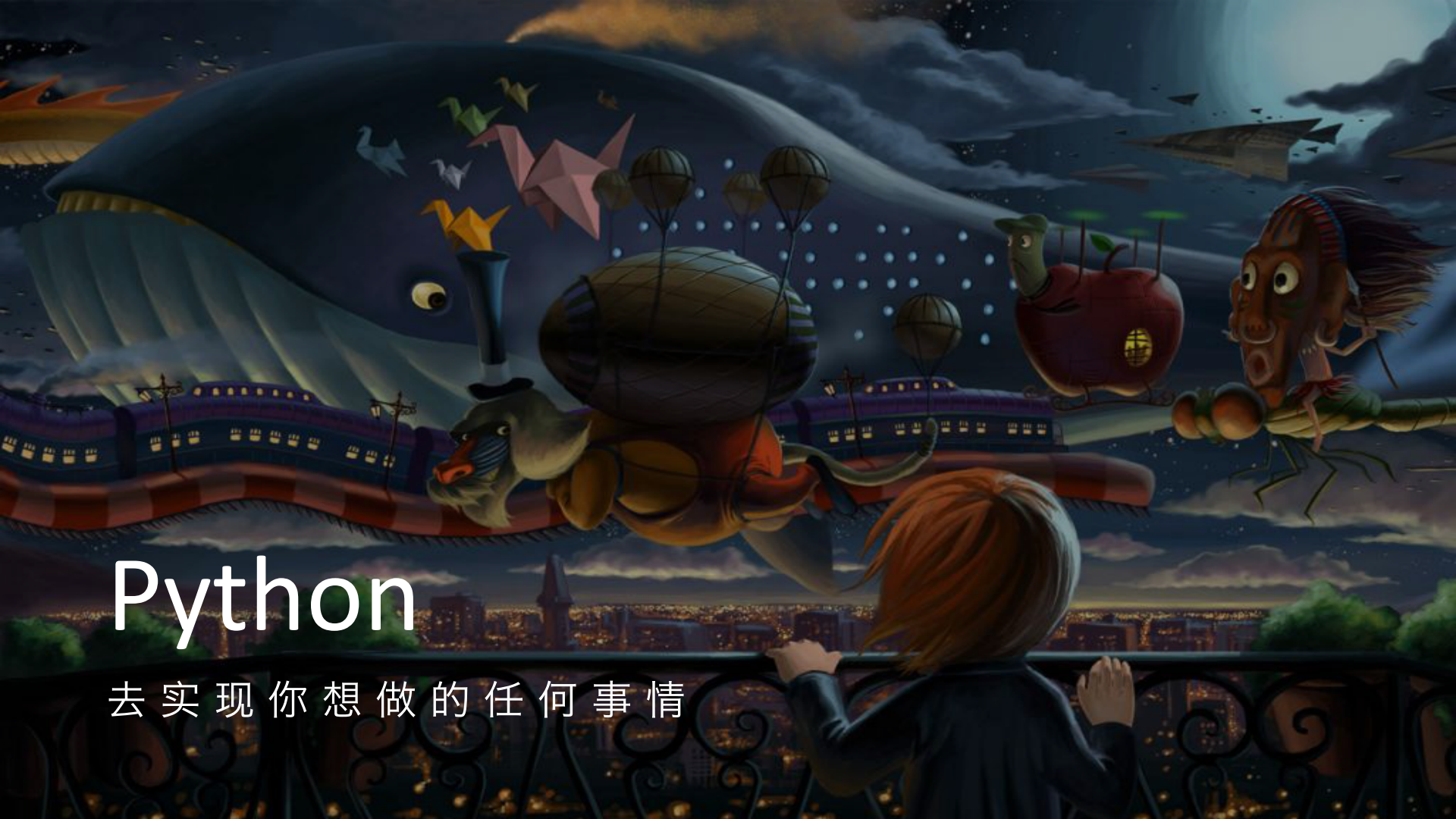
1 改写 评分等级

- ✓ 从下往上进行判断
- ✓ 即先处理 小于 0，然后 0~59，依次类推
- ✓ 最后处理 大于 100

作业

2 简易 计算器

- ✓ 使用 input() 输入第一个数字
- ✓ 使用 input() 输入第二个数字
- ✓ 使用 input() 输入运算符，例如 + - * / **
- ✓ 判断 运算符类别，并进行相应的计算
- ✓ 输出 计算结果


Python
去 实 现 你 想 做 的 任 何 事 情