

第三课 Python的流程控制语句

课时介绍

◆ 分支语句的简单书写

◆ 多条件下的分支语句

◆ 分支语句的应用

◆ 循环结构介绍

◆ 循环的执行和跳出、
终止

◆ 循环语句的应用

课程目标

◆ 掌握分支结构的使用方法

◆ 掌握循环结构的使用方法

流程控制语句

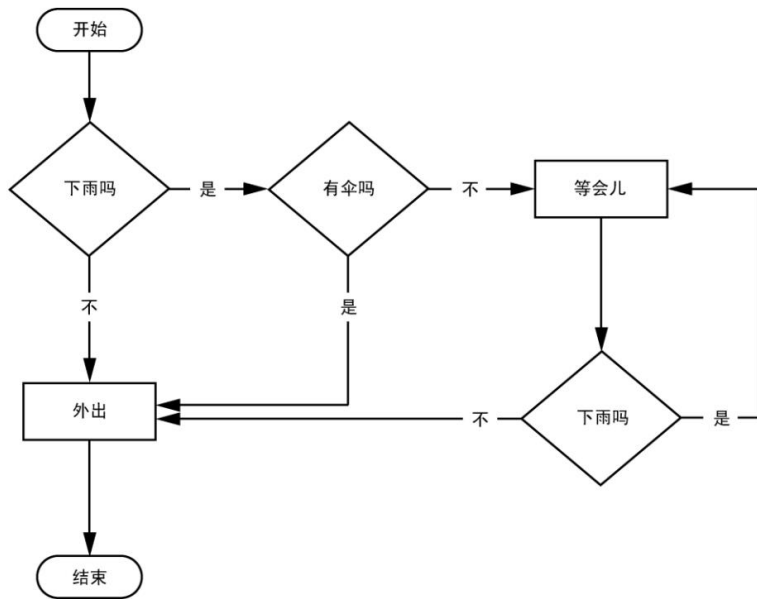
◆ 顺序执行

◆ 分支语句

◆ 循环语句

分支语句

- ◆ 工作生活无时不在的判断与选择，在编程中被称为“分支语句”



分支语句基本形式

◆ 分支语句基本书写形式

if 判断条件:

条件成立时执行语句块

else:

条件不成立时执行语句块

判断条件的书写规则

- ◆ 每个if语句的核心都是判断条件的编写
- ◆ 判断条件必须是返回True(成立)或者False(不成立)的**表达式**
- ◆ 我们通常称返回布尔类型的表达式为 **“布尔表达式”**

判断条件的书写规则

result = 1 + 1

result = "How are you ?" + "Fine,thank you."

result = (1+1)>3

result = 18 * 671 + (3321 - 871) % 31

等值判断

- ◆ 判断条件中使用 `==` 符号，来决定前后两值是否相等
- ◆ 判断条件中使用 `!=` 符号，来判断前后两值是否不相等
- ◆ 等值判断可以用于数字、字符串，且字符串区分大小写

比较运算符

运算符	描述
==	等于
!=	不等于
>	大于
<	小于
>=	大于等于
<=	小于等于

逻辑运算符

职位信息

1. 大专及以上学历，要求掌握PhotoShop
2. 掌握Java、Python、C任意一门语言优先录用
3. 非计算机相关专业不予考虑

逻辑运算符

- ◆ 逻辑运算符是指在多个条件组合判断时使用的运算符
- ◆ 逻辑运算符优先级为 not > and > or

运算符	描述	示例
and (与)	前后条件都成立,结果才成立	1==1 and 2>1 输出 True
or (或)	前后条件有一个成立, 结果就成立	1 > 2 or 2 > 1 输出True
not (非)	对结果取反	not 1==1 输出 False

综合训练-血压评估

正常血压范围

低压： $60\text{mmHg} < \text{舒张压} < 90\text{mmHg}$

高压： $90\text{mmHg} < \text{收缩压} < 140\text{mmHg}$

多分支语句

- ◆ 多分支语句是在原有基础上增加**elif**，进行多重判断

if 判断条件1:

 条件1成立时执行语句块.....

elif 判断条件2:

 条件2成立时执行语句块.....

elif ...

else:

 其他情况下时执行语句块.....

BMI指数计算器

- ◆ 身体质量指数 (Body Mass Index, 简称BMI), 亦称克托莱指数
- ◆ BMI指数 = 体重(kg) / 身高(m)的平方

分类	BMI范围
偏瘦	≤ 18.4
正常	18.5 ~ 23.9
过重	24.0 ~ 27.9
肥胖	≥ 28.0

分支语句嵌套

- ◆ 分支语句嵌套是指分支语句中再次使用if进行二次判断

```
has_ticket = input("乘客是否购买车票(1-是, 0-否):")
knife_length = int(input("请输入刀具长度 (公分,0-未携带) "))
if has_ticket:
    print("车票检查通过, 准备开始安检")
    if knife_length < 20:
        print("刀具未超过20厘米, 允许上车")
    else:
        print("管制刀具, 车站没收")
else:
    print("没有车票, 不允许进站")
```


分支语句嵌套

- ◆ 分支语句嵌套是指分支语句中再次使用if进行二次判断

```
has_ticket = input("乘客是否购买车票(1-是, 0-否):")
knife_length = int(input("请输入刀具长度 (公分,0-未携带) "))
if has_ticket and knife_length < 20:
    print("刀具未超过20厘米, 允许上车")
elif has_ticket and knife_length >= 20:
    print("管制刀具, 车站没收")
else:
    print("没有车票, 不允许进站")
```

流程控制语句

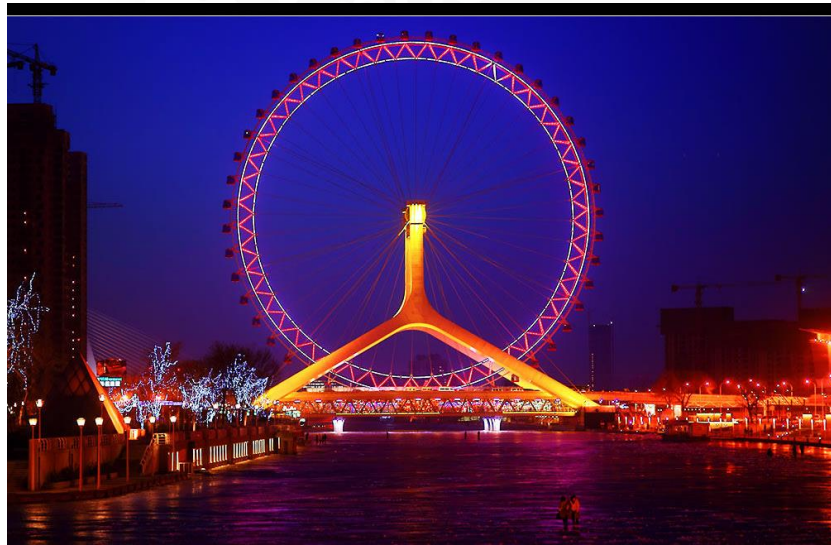
◆ 顺序执行

◆ 分支语句

◆ 循环语句

什么是循环

- ◆ 循环是在满足条件下周而复始的运行的情况



while循环

- ◆ while 关键字用于创建循环，在满足条件时，将循环执行语句块

while 循环执行条件：

被循环执行的代码块

修改循环的判断条件

计数器

- ◆ 通常循环有明确的执行次数，计数器便是保存循环次数的变量
- ◆ 计数器常用字母i (**i**ndex) 来命名

```
#计数器
```

```
i = 0
```

```
while i < 5:
```

```
    print("Python is the best language")
```

```
    i = i + 1
```

死循环

- ◆ 当循环执行条件永远成立时，没有终点的循环被称为“死循环”

```
i = 0
```

```
while i <= 10:
```

```
    print("Python is the best language")
```

```
    i = i + 1
```

综合训练：阶乘计算器

- ◆ 要求1：计算20的阶乘 ($1 \times 2 \times 3 \times \dots \times 20$)
- ◆ 要求2：当前阶乘数能被5整除，则打印中间结果
- ◆ 要求3：能够输入自定义的数值 (1-100)

continue与break

- ◆ continue 用于跳过当前循环的剩余语句
- ◆ break关键字用来终止循环语句

循环嵌套

- ◆ 循环嵌套是指循环中出现循环的使用情况

```
while 条件1:
```

```
    while 条件2:
```

```
        内部循环语句
```

```
    外部循环语句
```

列出1000以内的质数



第三课 Python的流程控制语句

课程总结

- ◆ 分支结构用于处理程序中判断与选择
- ◆ 循环结构用于重复执行特定的代码

流程控制语句

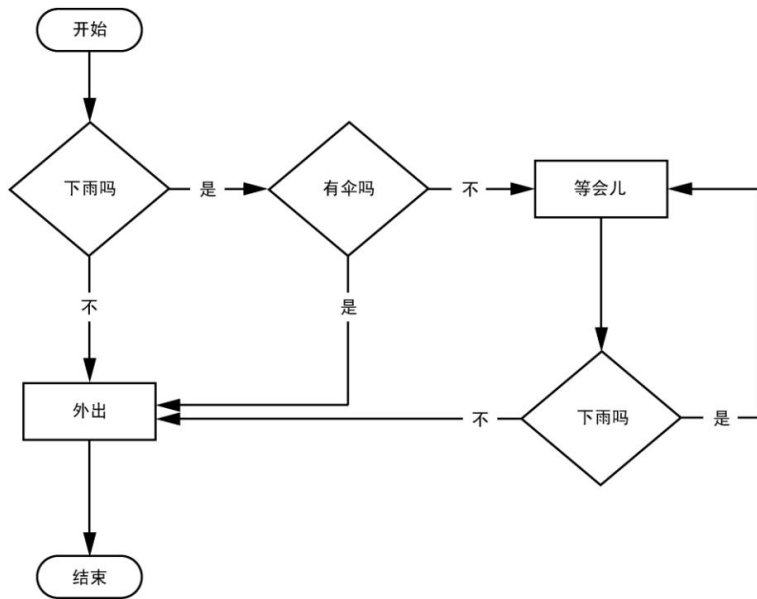
◆ 顺序执行

◆ 分支语句

◆ 循环语句

分支语句

- ◆ 工作生活无时不在的判断与选择，在编程中被称为“分支语句”



比较运算符

运算符	描述
==	等于
!=	不等于
>	大于
<	小于
>=	大于等于
<=	小于等于

逻辑运算符

- ◆ 逻辑运算符是指在多个条件组合判断时使用的运算符
- ◆ 逻辑运算符优先级为 not > and > or

运算符	描述	示例
and (与)	前后条件都成立,结果才成立	1==1 and 2>1 输出 True
or (或)	前后条件有一个成立, 结果就成立	1 > 2 or 2 > 1 输出True
not (非)	对结果取反	not 1==1 输出 False

多分支语句

- ◆ 多分支语句是在原有基础上增加**elif**，进行多重判断

if 判断条件1:

 条件1成立时执行语句块.....

elif 判断条件2:

 条件2成立时执行语句块.....

elif ...

else:

 其他情况下时执行语句块.....

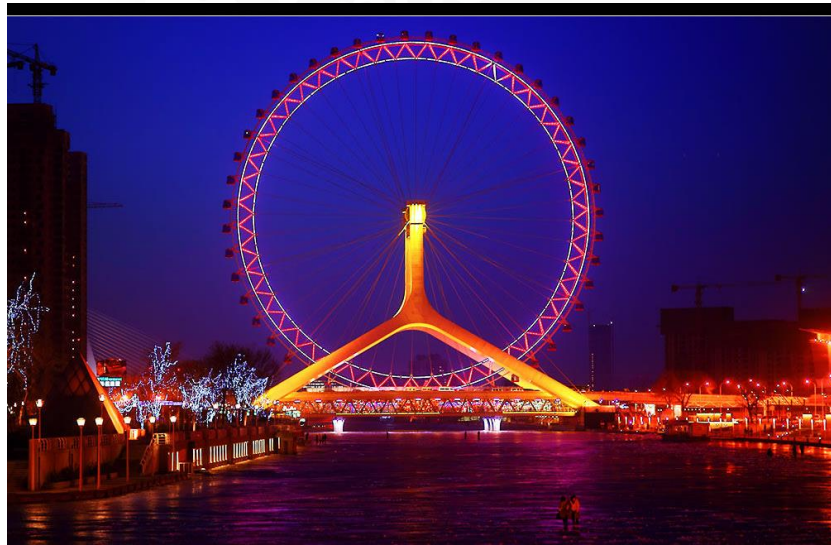
分支语句嵌套

- ◆ 分支语句嵌套是指分支语句中再次使用if进行二次判断

```
has_ticket = input("乘客是否购买车票(1-是, 0-否):")
knife_length = int(input("请输入刀具长度 (公分,0-未携带) "))
if has_ticket:
    print("车票检查通过, 准备开始安检")
    if knife_length < 20:
        print("刀具未超过20厘米, 允许上车")
    else:
        print("管制刀具, 车站没收")
else:
    print("没有车票, 不允许进站")
```

什么是循环

- ◆ 循环是在满足条件下周而复始的运行的情况



while循环

- ◆ while 关键字用于创建循环，在满足条件时，将循环执行语句块

while 循环执行条件：

被循环执行的代码块

修改循环的判断条件

计数器

- ◆ 通常循环有明确的执行次数，计数器便是保存循环次数的变量
- ◆ 计数器常用字母i (**i**ndex) 来命名

```
#计数器
```

```
i = 0
```

```
while i < 5:
```

```
    print("Python is the best language")
```

```
    i = i + 1
```

死循环

- ◆ 当循环执行条件永远成立时，没有终点的循环被称为“死循环”

```
i = 0
```

```
while i <= 10:
```

```
    print("Python is the best language")
```

```
    i = i + 1
```

continue与break

- ◆ continue 用于跳过当前循环的剩余语句
- ◆ break关键字用来终止循环语句

循环嵌套

- ◆ 循环嵌套是指循环中出现循环的使用情况

```
while 条件1:
```

```
    while 条件2:
```

```
        内部循环语句
```

```
    外部循环语句
```


列出1000以内的质数

