



第3章 流程控制

授课老师：刘国旭

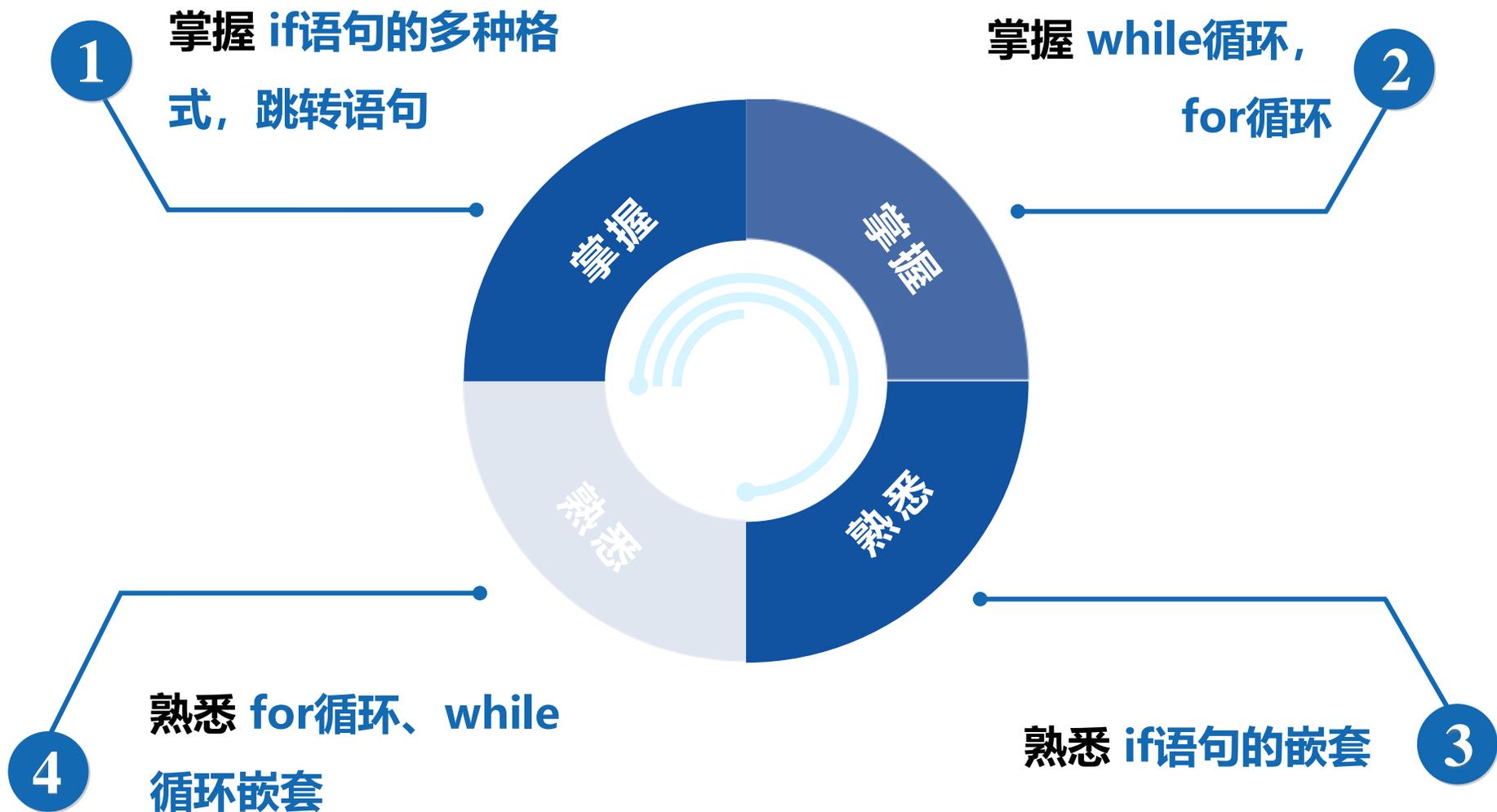
潍坊科技学院



- if语句
- if语句的嵌套
- 循环语句
- 循环嵌套
- 跳转语句



学习目标





目录页



3.1 条件语句

3.2 实训案例

3.3 循环语句

3.4 实训案例

3.5 跳转语句

3.6 阶段案例——房贷计算器



目录页



潍坊科技学院
Weifang University of Science and Technology



3.1 条件语句

3.2 实训案例

3.3 循环语句

3.4 实训案例

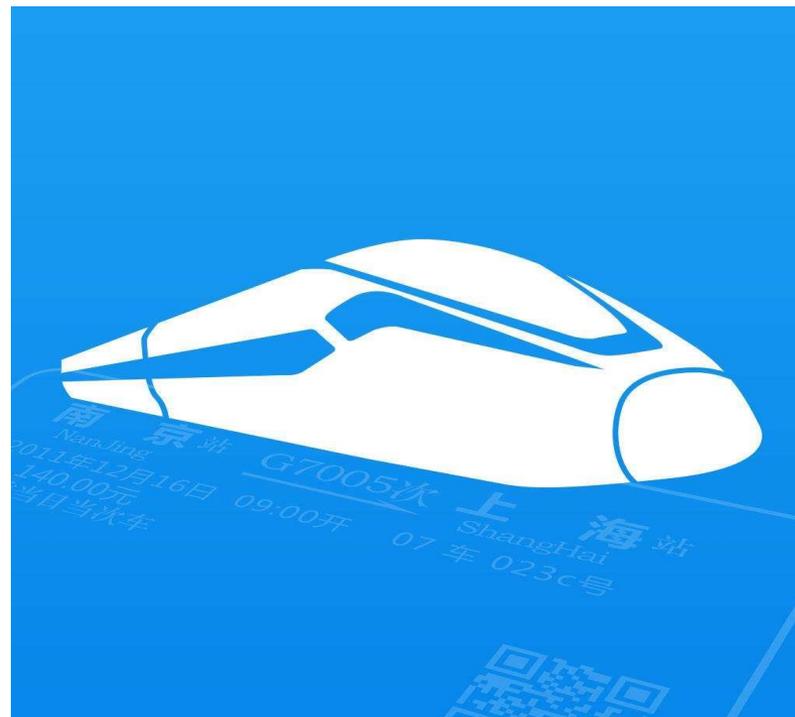
3.5 跳转语句

3.6 阶段案例——房贷计算器



3.1 条件语句

现实生活中，大家在12306网站购票时需要先验证身份，验证通过后可进入购票页面，验证失败则需重新验证。在代码编写工作中，大家可以使用**条件语句**为程序**增设条件**，使程序**产生分支**，进而有选择地执行不同的语句。





3.1.1 if语句



if语句由关键字**if**、**判断条件**和**冒号**组成，if语句和从属于该语句的代码段可组成选择结构。

格式

if 条件表达式:
 代码块

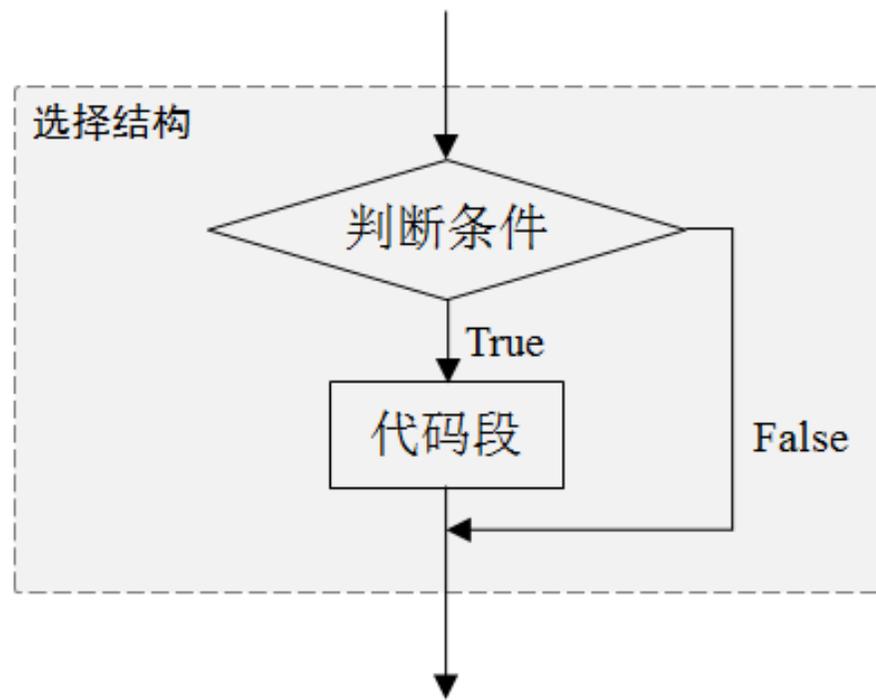




3.1.1 if语句



执行if语句时，若if语句的**判断条件成立**（判断条件的布尔值为**True**），执行之后的代码段；若if语句的**判断条件不成立**（判断条件的布尔值为**False**），跳出选择结构，继续向下执行。



if语句的执行流程



3.1.1 if语句



```
score = 88  
if score >= 60:  
    print( "考试及格! " )
```



3.1.2 if-else语句



一些场景不仅需要处理满足条件的情况，也需要对不满足条件的情况做特殊处理。因此，Python提供了可以同时处理满足和不满足条件的if-else语句。

格式

```
if 判断条件:  
    代码块1  
else:  
    代码段2
```

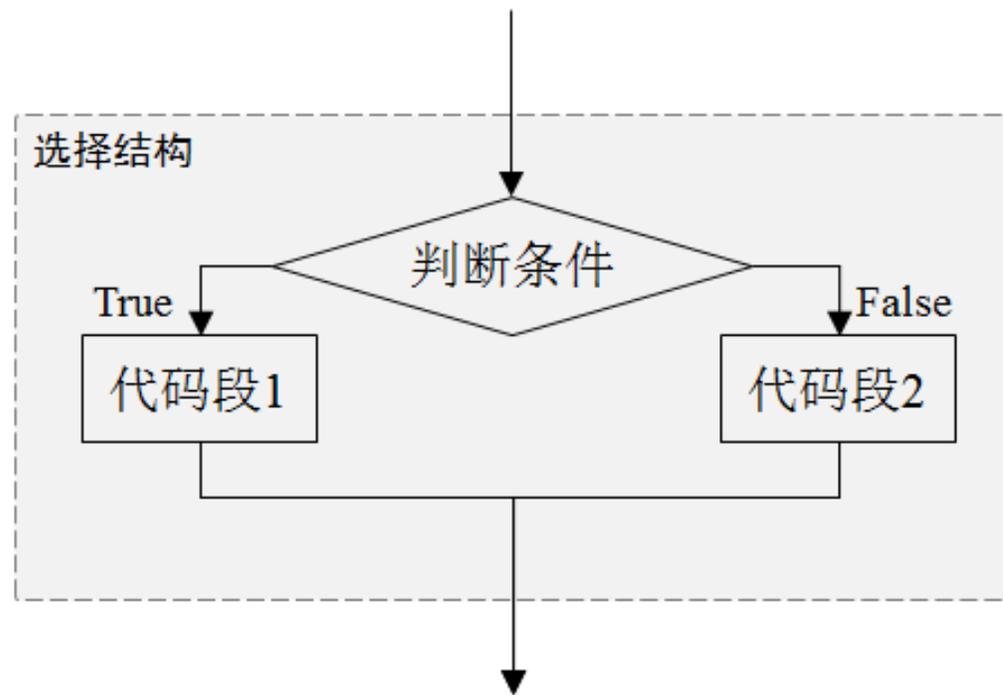




3.1.2 if-else语句



执行if-else语句时，若判断条件**成立**，执行if语句之后的**代码段1**；若判断条件**不成立**，执行else语句之后的**代码段2**。



if-else语句的执行流程



3.1.2 if-else语句



```
score = 88
if score >= 60:
    print( "考试及格! " )
else:
    print( "考试不及格! " )
```



3.1.3 if-elif-else语句



Python除了提供单分支和双分支条件语句外，还提供多分支条件语句**if-elif-else**。多分支条件语句用于处理单分支和双分支无法处理的情况。

```
if 判断条件1:  
    代码段1  
elif 判断条件2:  
    代码段2  
elif 判断条件3:  
    代码段3  
...  
else:  
    代码段n
```

格式

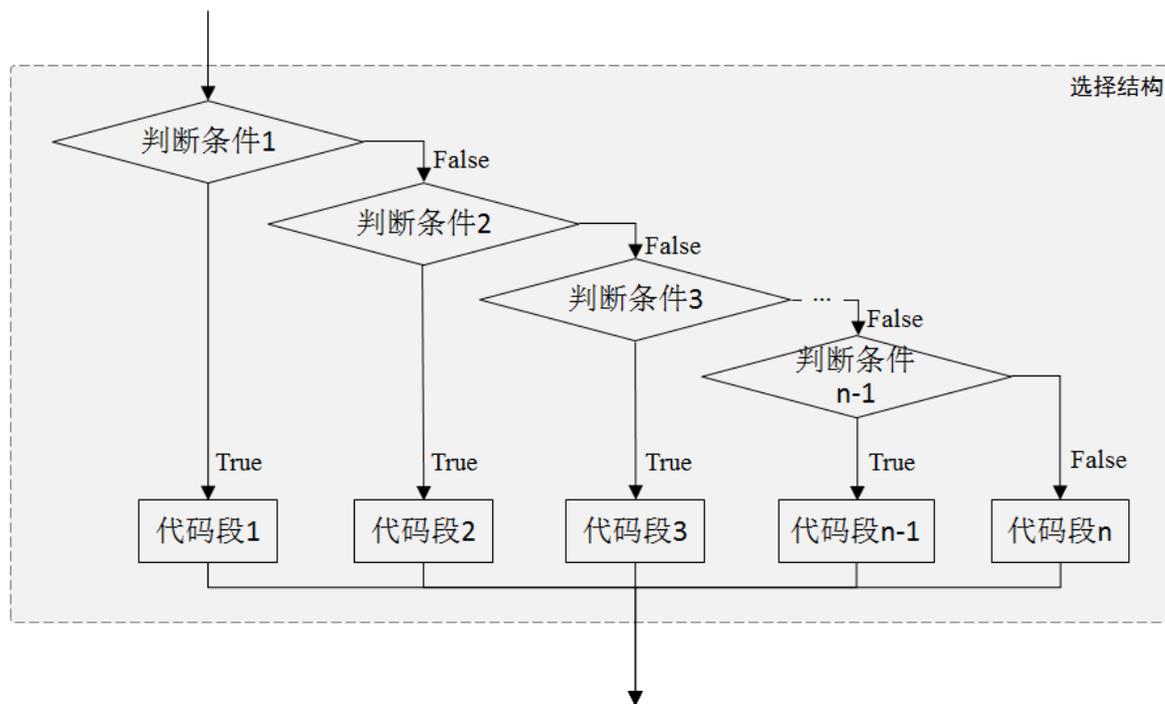




3.1.3 if-elif-else语句



执行if-elif-else语句时，若if条件成立，执行if语句之后的代码段1；若if条件不成立，判断elif语句的判断条件2：条件2成立则执行elif语句之后的代码段2，否则继续向下执行。以此类推，直至所有的判断条件均不成立，执行else语句之后的代码段。



if-elif-else语句的执行流程



3.1.3 if-elif-else语句



```
score = 88
if score >= 85:
    print( "优秀" )
elif 75 <= score < 85:
    print( "良好" )
elif 60 <= score < 75:
    print( "中等" )
else:
    print( "差" )
```



3.1.4 if嵌套



大家在某些火车站乘坐高铁出行时需要历经检票和安检两道程序：检票符合条件后方可进入安检程序，安检符合条件后方可进站乘坐列车。这个场景中虽然涉及两个判断条件，但这两个条件并非选择关系，而是嵌套关系：先判断外层条件，条件满足后才去判断内层条件；两层条件都满足时才执行内层的操作。





3.1.4 if嵌套



Python中通过if嵌套可以实现程序中条件语句的嵌套逻辑。

格式

```
if 判断条件1:           # 外层条件
    代码段1
    if 判断条件2:       # 内层条件
        代码段2
    ...
```

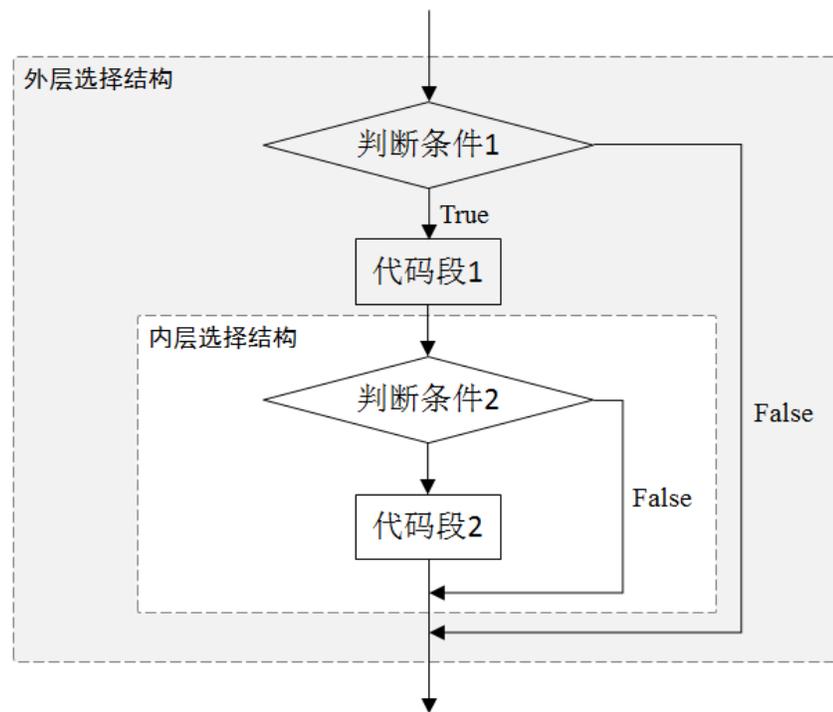




3.1.4 if嵌套



执行if嵌套时，若**外层判断条件**（判断条件1）的值为**True**，执行**代码段1**，并对**内层判断条件**（判断条件2）进行判断：若判断条件2的值为**True**，则执行代码段2，否则跳出内层条件结构，顺序执行外层条件结构中内层条件结构之后的代码；若**外层判断条件**的值为**False**，直接**跳过条件语句**，既不执行代码段1，也不执行内层的条件结构。



if嵌套的执行流程



目录页



潍坊科技学院
Weifang University of Science and Technology



3.1 条件语句

3.2 实训案例

3.3 循环语句

3.4 实训案例

3.5 跳转语句

3.6 阶段案例——房贷计算器



3.2.1 计算器



计算器极大地提高了人们进行数字计算的效率与准确性，无论是超市的收银台，还是集市的小摊位，都能够看到计算器的身影。计算器最基本的功能是四则运算。

本实例要求编写代码，实现计算器的四则运算功能。



3.2.2 猜数字



猜数字由两个人参与，一个人设置一个数字，一个人猜数字，当猜数字的人说出一个数字，由出数字的人告知是否猜中：若猜测的数字大于设置的数字，出数字的人提示“很遗憾，你猜大了”；若猜测的数字小于设置的数字时，出数字的人提示“很遗憾，你猜小了”；若猜数字的人在规定的次数内猜中设置的数字，出数字的人提示“恭喜，猜数成功”。

本实例要求编写代码，实现遵循上述规则的猜数字程序。



目录页



潍坊科技学院
Weifang University of Science and Technology



3.1 条件语句

3.2 实训案例

3.3 循环语句

3.4 实训案例

3.5 跳转语句

3.6 阶段案例——房贷计算器

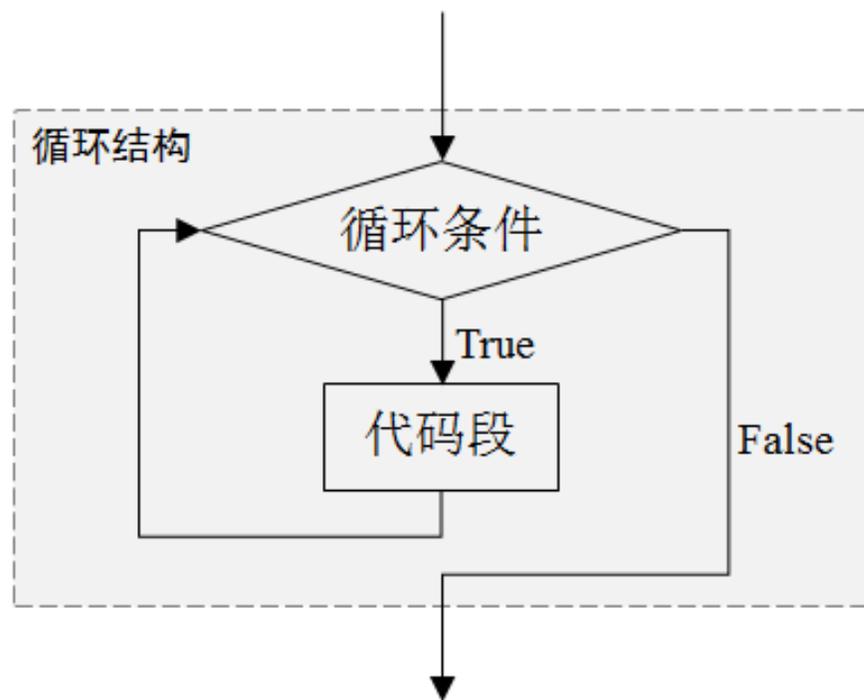
3.3.1 while语句

while语句一般用于实现条件循环，该语句由关键字while、循环条件和冒号组成，while语句和从属于该语句的代码段组成**循环结构**。

while 条件表达式:

代码块

格式





3.3.1 while语句



```
i = 1
result = 0
while i <= 10:
    result += i
    i += 1
print(result)
```



3.3.2 for语句



for语句一般用于实现**遍历循环**。遍历指逐一访问目标对象中的数据，例如逐个访问字符串中的字符；遍历循环指在循环中完成对**目标对象**的遍历。

for 临时变量 in 目标对象:

代码块

格式





3.3.2 for语句



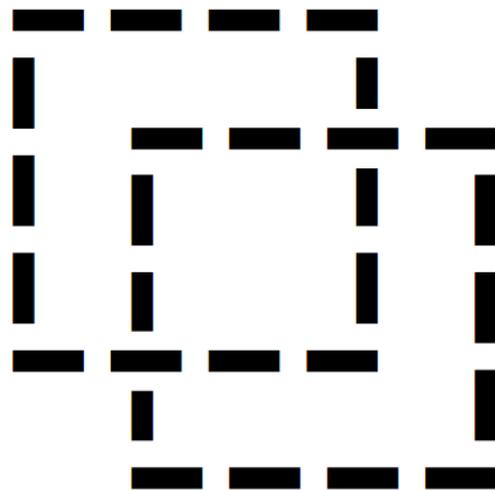
```
for word in "Python" :  
    print(word)
```



3.3.3 循环嵌套



循环之间可以互相嵌套，进而实现更为复杂的逻辑。循环嵌套按不同的循环语句可以划分为 **while** 循环嵌套和 **for** 循环嵌套。





3.3.3 循环嵌套



while循环嵌套是指while语句中嵌套了while或for语句。以while语句中嵌套while语句为例，while循环嵌套的语法格式如下：

```
while 循环条件1:           # 外层循环
    代码段1
    while 循环条件2:       # 内层循环
        代码段2
    .....
```

格式





3.3.3 循环嵌套



```
i = 1
while i < 6:
    j=0
    while j < i:
        print( "*" , end= " ")
        j += 1
    print()
    i += 1
```



3.3.3 循环嵌套



for循环嵌套是指for语句中嵌套了while或for语句。以for语句中嵌套for语句为例，for循环嵌套的语法格式如下：

```
for 临时变量 in 目标对象:           # 外层循环
    代码段1
    for 临时变量 in 目标对象:       # 内层循环
        代码段2
    .....
```

格式





3.3.3 循环嵌套



```
for i in range(1,6):  
    for j in range(i):  
        print( "*" , sep= " " )  
    print()
```



目录页



潍坊科技学院
Weifang University of Science and Technology



3.1 条件语句

3.2 实训案例

3.3 循环语句

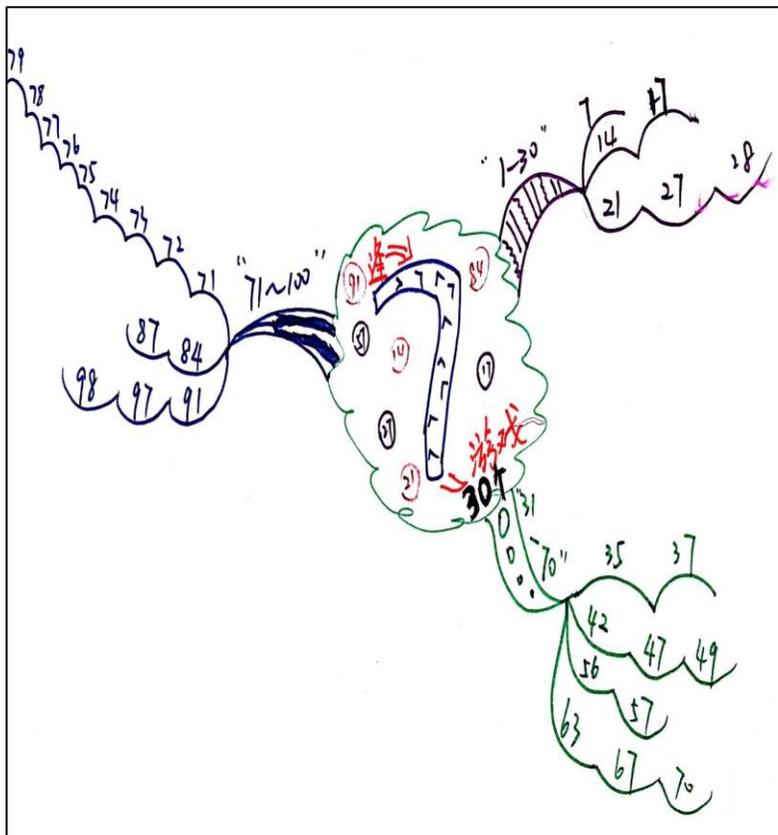
3.4 实训案例

3.5 跳转语句

3.6 阶段案例——房贷计算器



3.4.1 逢7拍手游戏

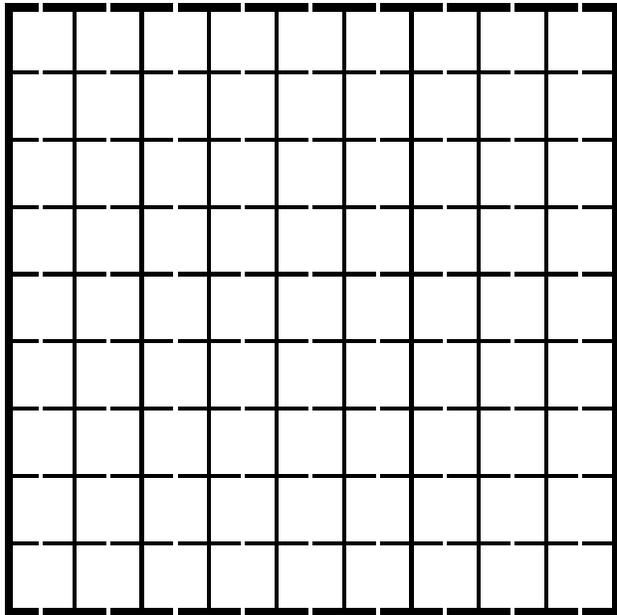


逢7拍手游戏的规则是：从1开始顺序数数，数到有7或者包含7的倍数的时候拍手。

本实例要求编写代码，模拟实现逢七拍手游戏，实现输出100以内需要拍手的数字的程序。



3.4.2 打印五子棋棋盘



五子棋是一种由双人对弈的纯策略型棋类游戏，它使用的棋盘一般由纵横等距的各15条平行线构成，这些线垂直交叉形成的**225个交叉点**为对弈双方的落子点。

本实例要求编写代码，实现按用户要求**打印指定大小的五子棋棋盘**的程序。



目录页



潍坊科技学院
Weifang University of Science and Technology



3.1 条件语句

3.2 实训案例

3.3 循环语句

3.4 实训案例

3.5 跳转语句

3.6 阶段案例——房贷计算器



3.5.1 break语句



break语句用于**结束循环**，若循环中使用了break语句，程序执行到break语句时会结束循环；若**循环嵌套**使用了break语句，程序执行到break语句时会**结束本层循环**。

```
for word in "Python":  
    if (word == 'o'):  
        break  
    print(word, end=" ")
```

示例

Pyth

结果



3.5.1 break语句



break语句用于**结束循环**，若循环中使用了break语句，程序执行到break语句时会结束循环；若**循环嵌套**使用了break语句，程序执行到break语句时会结束**本层循环**。

示例

```
for x in "123":  
    for y in "1234":  
        if (y == '3'):  
            break  
        print(y, end=" ")  
    print(x, end=" ")
```



3.5.2 continue语句



continue语句用于在满足条件的情况下跳出**本次循环**，该语句通常也与**if**语句配合使用。

```
for word in "Python":  
    if (word == 'o'):  
        continue  
    print(word, end=" ")
```

示例

P y t h n

结果



目录页



潍坊科技学院
Weifang University of Science and Technology



3.1 条件语句

3.2 实训案例

3.3 循环语句

3.4 实训案例

3.5 跳转语句

3.6 阶段案例——房贷计算器



3.6阶段案例——房贷计算器



房贷计算 利率表

等额本息 等额本金

每月月供参考(元)
0.00

支付利息(元)	还款总额(元)
0.00	0.00

等额本息还款详情

贷款类型	商业贷款 >
贷款金额(万)	请输入金额
期限(年)	5 >
利率(%)	4.75 >

以上为央行2016年最新公布的贷款基准利率

房贷计算器是支付宝平台中搭载的一款在线计算工具，按用户选择的贷款类型（商业贷款、公积金贷款、组合贷款）、贷款金额（万）、期限（年）、利率（%）可计算得出每月月供参考（元）、支付利息（元）、还款总额（元）这些信息。



3.6阶段案例——房贷计算器



房贷计算公式如下：

- ▶ 每月月供参考 = $\text{贷款金额} \times [\text{月利率} \times (1 + \text{月利率}) ^ \text{还款月数}] \div \{ [(1 + \text{月利率}) ^ \text{还款月数}] - 1 \}$
- ▶ 还款总额 = 每月月供参考 \times 期限 \times 12
- ▶ 支付利息 = 还款总额 - 贷款金额 \times 10000

以上计算方式中月利率（ $\text{月利率} = \text{利率} \div 12$ ）指以月为计息周期计算的利息。不同贷款类型的利率是不同的：对于商业贷款而言，五年以下（含五年）的贷款利率是4.75%，五年以上的贷款利率是4.90%；对于公积金贷款利率而言，五年以下（含五年）的贷款利率是2.75%，五年以上的利率是3.25%。

本案例要求编写程序，根据以上计算方式开发一个房贷计算器。



3.7 本章小结



本章主要讲解了**流程控制**的相关知识，包括**条件语句**、**循环语句**、**跳转语句**，并结合众多实训案例演示了如何利用各种语句实现流程控制。通过本章的学习，希望读者能掌握程序的执行流程和流程控制语句的用法，为后续的学习打好扎实的基础。