Python 程序设计教案

<table>
<tr><td>本次授课内容</td><td colspan="2">4.1 条件表达式
4.2 选择结构
4.3 循环结构
4.4 综合案例解析</td></tr>
<tr><td>本次课的
教学目的</td><td colspan="2">理解条件表达式与 True/False 的等价关系
熟练运用常见选择结构
熟练运用 for 循环和 while 循环
理解带 else 子句的循环结构执行过程
理解 break 和 continue 语句在循环中的作用</td></tr>
<tr><td>本次课教学
重点与难点</td><td colspan="2">for 循环本质
else 在选择结构和循环结构中的用法</td></tr>
<tr><td>教学方法
教学手段</td><td colspan="2">PPT、边讲边练</td></tr>
<tr><td rowspan="3">课堂教学
时间分配</td><td>教学内容</td><td>时间分配（分）</td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td>课堂教学设计</td><td colspan="2">首先介绍条件表达式的概念和计算方法，然后介绍选择结构和循环结构的语法，最后通过几个例题演示选择结构和循环结构的应用。</td></tr>
<tr><td>实　验</td><td colspan="2">选择结构与循环结构</td></tr>
<tr><td>思考题及作业题</td><td colspan="2">课后习题 1-13 题</td></tr>
<tr><td>备　注</td><td colspan="2"></td></tr>
<tr><td>教学后记</td><td colspan="2"></td></tr>
<tr><td></td><td colspan="2">第　一　节　课</td></tr>
</table>

课堂重点内容详解	在选择结构和循环结构中，都要根据条件表达式的值来确定下一步的执行流程。条件表达式的值只要不是 False、0（或 0.0、0j 等）、空值 None、空列表、空元组、空集合、空字典、空字符串、空 range 对象或其他空迭代对象，Python 解释器均认为与 True 等价。
	
	单分支选择结构语法如下所示，其中表达式后面的冒号“:”是不可缺少的，表示一个语句块的开始，并且语句块必须做相应地缩进，一般是以 4 个空格为缩进单位。
	
	`if 表达式:`
	`    语句块`
	
	当表达式值为 True 或其他与 True 等价的值时，表示条件满足，语句块被执行，否则该语句块不被执行，而是继续执行后面的代码（如果有的话），如图 4-1 所示。
	
	图 4-1 单分支选择结构
	
	双分支选择结构的语法为：
	
	`if 表达式:`
	`    语句块 1`
	`else:`
	`    语句块 2`
	
	当表达式值为 True 或其他等价值时，执行语句块 1，否则执行语句块 2。语句块 1 或语句块 2 总有一个会执行，然后再执行后面的代码（如果有的话），如图 4-2 所示。
	
	图 4-2 双分支选择结构
	
	多分支选择结构的语法为：
	
	`if 表达式 1:`
	`    语句块 1`
	`elif 表达式 2:`
	`    语句块 2`
	`elif 表达式 3:`
	`    语句块 3`
	`......`
	`else:`
	`    语句块 n`
	
	其中，关键字 elif 是 else if 的缩写。
	
	选择结构可以进行嵌套，示例语法如下所示：
	
	`if 表达式 1:`
	`    语句块 1`
	`    if 表达式 2:`
	`        语句块 2`
	`    else:`

```
        语句块 3
else:
    if 表达式 4:
        语句块 4
```

Python 主要有 for 循环和 while 循环两种形式的循环结构，多个循环可以嵌套使用，也可以和选择结构嵌套使用来实现复杂的业务逻辑。

在 Python 中，循环结构可以带 else 子句，其执行过程为：如果循环因为条件表达式不成立或序列遍历结束而自然结束时则执行 else 结构中的语句，如果循环是因为执行了 break 语句而导致循环提前结束则不会执行 else 中的语句。while 循环和 for 循环完整语法形式分别为：

```
while 条件表达式:
    循环体
[else:
    else 子句代码块]
```

和

```
for 取值 in 序列或迭代对象:
    循环体
[else:
    else 子句代码块]
```

其中，方括号内的 else 子句可以没有，也可以有，根据要解决的问题来确定。

break 语句和 continue 语句在 while 循环和 for 循环中都可以使用，并且一般常与选择结构或异常处理结构结合使用。一旦 break 语句被执行，将使得 break 语句所属层次的循环提前结束；continue 语句的作用是提前结束本次循环，忽略 continue 之后的所有语句，提前进入下一次循环。

第 二 节 课

例 4-8 输入若干个成绩，求所有成绩的平均分。每输入一个成绩后询问是否继续输入下一个成绩，回答“yes”就继续输入下一个成绩，回答“no”就停止输入成绩。

```
numbers = []
while True:
    x = input('请输入一个成绩：')
    #异常处理结构，用来保证用户只能输入实数
    try:
        #先把 x 转换成实数，然后追加到列表 numbers 尾部
        numbers.append(float(x))
    except:
        print('不是合法成绩')
```

```
    #下面的循环用来限制用户只能输入任意大小写的“yes”或者“no”
    while True:
        flag = input('继续输入吗？（yes/no）').lower()
        if flag not in ('yes', 'no'):
            print('只能输入 yes 或 no')
        else:
            break
    if flag=='no':
        break

#计算平均分
print(sum(numbers)/len(numbers))
```

例 4-9 编写程序，判断今天是今年的第几天。

```
import time

date = time.localtime()                                  #获取当前日期时间
year, month, day = date[:3]                              #获取年、月、日信息
day_month = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
                                                         #一年中每个月的天数
if year%400==0 or (year%4==0 and year%100!=0):   #判断是否为闰年
    day_month[1] = 29                                     #闰年的 2 月是 29 天
if month==1:
    print(day)
else:
    print(sum(day_month[:month-1])+day)            #前面所有月的天数加上
                                                         #本月第几天
```

例 4-10 编写代码，输出由星号*组成的菱形图案，并且可以灵活控制图案的大小。

```
n = int(input('输入一个整数：'))
for i in range(n):
    print(('* '*i).center(n*3))      #center()是字符串排版方法，居中对齐
                                     #其中的参数 n*3 表示排版后字符串长度
for i in range(n, 0, -1):
    print(('* '*i).center(n*3))
```

例 4-11 快速判断一个数是否为素数。

```
n = input("Input an integer:")
n = int(n)
#2 是素数
if n == 2:
    print('Yes')
#除了 2 之外的所有偶数必然不是素数
elif n%2 == 0:
    print('No')
else:
    #大于 5 的素数必然出现在 6 的倍数两侧
    #因为 6x+2、6x+3、6x+4 肯定不是素数，假设 x 为大于 1 的自然数
    m = n % 6
```

```
    if m!=1 and m!=5:
        print('No')
    else:
        #判断整数 n 是否能被 3 到 n 的平方根之间的奇数整除
        for i in range(3, int(n**0.5)+1, 2):
            if n%i == 0:
                print('No')
                break
        else:
            print('Yes')
```

例 4-12 编写程序，计算组合数 C(n,i)，即从 n 个元素中任选 i 个，有多少种选法。

```
def Cni(n, i):
    if not (isinstance(n,int) and isinstance(i,int) and n>=i):
        print('n and i must be integers and n must be >=i.')
        return
    result = 1
    Min, Max = sorted((i,n-i))
    for i in range(n,0,-1):
        if i>Max:
            result *= i
        elif i<=Min:
            result /= i
    return result

print(Cni(6,2))
```

例 4-13 编写程序，输入一个自然数 n，然后计算并输出前 n 个自然数的阶乘之和 1!+2!+3!+...+n!的值。

```
n = int(input('请输入一个自然数：'))
#使用 result 保存最终结果，t 表示每一项
result, t = 1, 1
for i in range(2, n+1):
    #在前一项的基础上得到当前项
    t *= i
    #把当前项加到最终结果上
    result += t
print(result)
```

例 4-14 编写代码，模拟决赛现场最终成绩的计算过程。有至少 3 个评委，打分规则为删除最高分和最低分之后计算剩余分数的平均分。

```
while True:
    try:
        n = int(input('请输入评委人数：'))
        if n <= 2:
            print('评委人数太少,必须多于 2 个人。')
        else:
```

```
            break
    except:
        #pass 是空语句，表示什么也不做
        pass

for i in range(n):
    #这个 while 循环用来保证用户必须输入 0 到 100 之间的数字
    while True:
        try:
            score = input('请输入第{0}个评委的分数：'.format(i+1))
            #把字符串转换为实数
            score = float(score)
            assert 0<=score<=100
            scores.append(score)
            #如果数据合法，跳出 while 循环，继续输入下一个评委的分数
            break
        except:
            print('分数错误')

#计算并删除最高分与最低分
highest = max(scores)
lowest = min(scores)
scores.remove(highest)
scores.remove(lowest)
finalScore = round(sum(scores)/len(scores),2)

formatter = '去掉一个最高分{0}\n 去掉一个最低分{1}\n 最后得分{2}'
print(formatter.format(highest, lowest, finalScore))
```

例 4-15 编写程序，实现人机对战的尼姆游戏。

```
from random import randint

n = int(input('请输入一个正整数：'))
while n > 1:
    #人类玩家先走
    print("该你拿了，现在剩余物品数量为：{0}".format(n))
    #确保人类玩家输入合法整数值
    while True:
        try:
            num = int(input('输入你要拿走的物品数量：'))
            #确保拿走的物品数量不超过一半
            assert 1 <= num <= n//2
            break
        except:
            print('最少必须拿走 1 个，最多可以拿走{0}个。'.format(n//2))
    n -= num
    if n == 1:
        print('恭喜,你赢了！')
        break
    #计算机玩家随机拿走一些，randint()用来生成指定范围内的一个随机数
```

```
    n -= randint(1, n//2)
else:
    print('哈哈，你输了。')
```