

创客教育与少儿编程

——帮助孩子赢在未来



创客童年 · Python专题



● 公司简介：

专注于少儿编程、创客教育的课程研发，成立初衷就致力于降低少儿编程、创客教育门槛。在运营线下校区的同时筹备线上云课堂。目前在全网推出的系列课程有：

《陪孩子一起学Arduino》

《陪孩子一起学Python》、

《陪孩子一起学MicroPython与物联网》

目前这些课程在千聊课堂、网易云课堂、腾讯课堂，等各大平台的同类课程中均排名前列，服务老师，学生过万人。

18年12月底向行业开放课程合作，为学校、机构提供视频教程、PPT课件，配套器材等等服务，帮助学校、机构快速有效地开设创客编程课程。

● 创始人简介：

肖华盛，创客童年·创始人，线上云课堂负责人、主讲老师，原长沙“小小创客空间”创始人，从事创客教育近6年，一直负责课程研发，教学，团队师资建设。研发的中小学创客课程有：Arduino机器人、Python编程、Java编程，中鸣机器人竞赛，MicroPython物联网课程等等，总课时达600多课时。

本次培训的目标



- 对Python基础课程的知识体系有系统的认识。
- 学习Python课程知识、程序讲解技巧
- 分享机构、学校快速落地开设创客课程的解决方案

课程整体规划:



Arduino
智能生活系列

四~五年级



Arduino
智能机器人设计



五-六年级



Python编程入门



六-七年级



MicroPython
物联网智能硬件

六--七年级



Python编程进阶



七-九年级

Curriculum learning plan

Python课程大纲:



- 第一章：准备出发
- 第二章：内存与变量
- 第三章：数学运算
- 第四章：数据类型
- 第五章：程序的输入
- 第六章：图形用户界面
- 第七章：选择程序结构
- 第八章：循环程序结构
- 第九章：列表
- 第十章：元组、字典
- 第十一章：函数
- 第十二章：对象
- 第十三章：正则表达式
- 第十四章：模块
- 第十五章：Pygame
- 第十六章：检测
- 第十七章：事件
- 第十八章：声音
- 第十九章：GUI
- 第二十章：文件读写
- 第二十一章：游戏开发

Python课程目录:



第一季: Python入门	第二季: 选择程序结构	第三季: 循环程序结构
1. HellWorld	10. 关系运算符	19. while循环
2. print打印输出	11. if条件语句	20. while循环实例练习
3. 可爱的变量	12. 逻辑运算符	21. 项目-猜数字游戏
4. 基本数学运算	13. 各运算符的优先级	22. for循环
5. 基本数据类型	14. if-else语句	23. break与continue
6. 数据的输入	15. 多重if结构	24. 循环的嵌套
7. 图形交互程序设计(上)	16. if结构的嵌套	25. Pycharm与海龟绘图
8. 图形交互程序设计(下)	17. 实例练习	26. 项目-模拟银行系统(上)
	18. 项目-猜拳小游戏(上)	27. 项目-模拟银行系统(下)
	19. 项目-猜拳小游戏(下)	28. 项目-万年历(上)
		29. 项目-万年历(下)

Python语言的特点？



- Python上手简单，程序容易编写，容易理解
- Python是开源、免费的软件，在网上能够很方便下载到有用，好玩的Python程序与Python工具，很容易找到Python学习交流社区
- 跨平台性，也就是Python语言可以在不同类型的计算机上运行
- Python语言运用人群广泛，目前很多大公司都在使用Python，包括NASA（美国航空航天局）、谷歌等
Python在2017年的编程语言排行榜中一路飙升已成为全球第四大编程语言。



安装Python



安装完成后，测试是否安装成功

- 在桌面开始菜单中找到“运行”，
- 在“运行”中输入cmd进入dos命令行
- 在dos命令行中输入python，然后回车

```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>python
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```


Python入门: print语句



示例: `print(“hello world”)`

- `print()`: 是Python库中提供的一个打印函数。
- `print()` 函数可以打印字符, 整数等, 程序中的字符类型得用 “ ” 或用 ‘ ’ 标注, 否则程序无法识别

print格式化输出



- 转义字符”\”：在程序中用”\”来转义出一些特殊的意义字符。
- 什么地方需要用到转义字符？
例如需要输出以下代码：“I” am jim
- 常见的转义字符有：\t, \n, \\。

\t:空一个制表位，相当于Tab的功能

\n:换行

\\: 转义出本身\

用格式化输出的方式输出以下列表

	语文	数学	英语	计算机	
张三	92	89	80	100	
李四	80	85	91	99	
王五	99	100	98	100	

sep与end参数



● print函数中默认的两个参数sep和end

1. sep分隔符参数

例: `name = "Jim"`

`print("My name is", name)`

输出: `My name is Jim`

`print("My name is", name, sep = "@")`

输出: `My name is@Jim`

说明: 在print内部修改了sep的默认值, 修改@为分隔符

2. end结束符参数

例: `name = "Jim"`

`print("My name is", name, end = ", ")`

`print("I am from ChangSha")`

输出: `My name is Jim, I am from ChangSha`

程序

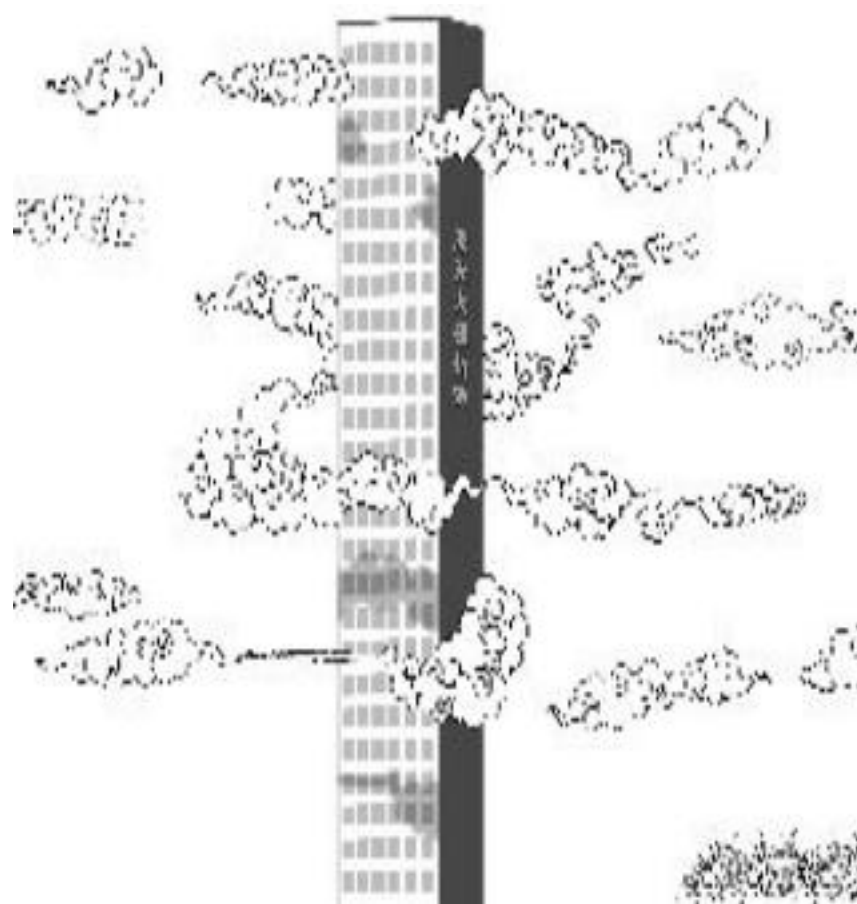


- 组成:



内存与变量

- 计算机为了可以处理大量的来自外部输入的数据，建了一栋大厦（叫内存空间），大厦里面有很多很多小房间（变量）可以用来存储各式各样的数据。



内存地址与变量名



- 内存地址：

计算机按照一定规则给内存大厦里的一个个小房间（即变量）设定的编号，例如：405455088。

- 变量名：

程序员为了方便记忆，给将要存放数据的小房间（即变量）添的一个有意义的标签， 例如：age，这样之后通过索引标签名（age）就可以获取到我们存放的年龄数据了。

变量的赋值



- 变量的赋值：
把数据存放到指定变量名（即房间标签名）的过程叫变量的赋值

- 例：
name = “Jack”
age = 12
height = 1.53

变量的命名



- 命名规则
 - ❑ 必须以字母或下划线“_”开头，不能以数字开头，后面部分可以使用字母、数字或下划线字符。
 - ❑ 变量的名字是区分大小的，例如Apple和apple是不同的。
 - ❑ 除了字母、数字、下划线“_”外，变量名里面，不能含有其他特殊字符

判断正确性

my_mother
student666
6apple
you teather

money_99\$
last-name
City
city



练习1

- 用变量存储姓名，年龄，城市信息，然后再通过变量打印输出。

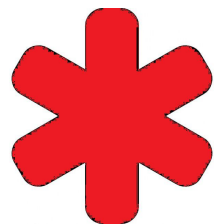
name: Mary

age: 13

city: 长沙



基本数学运算

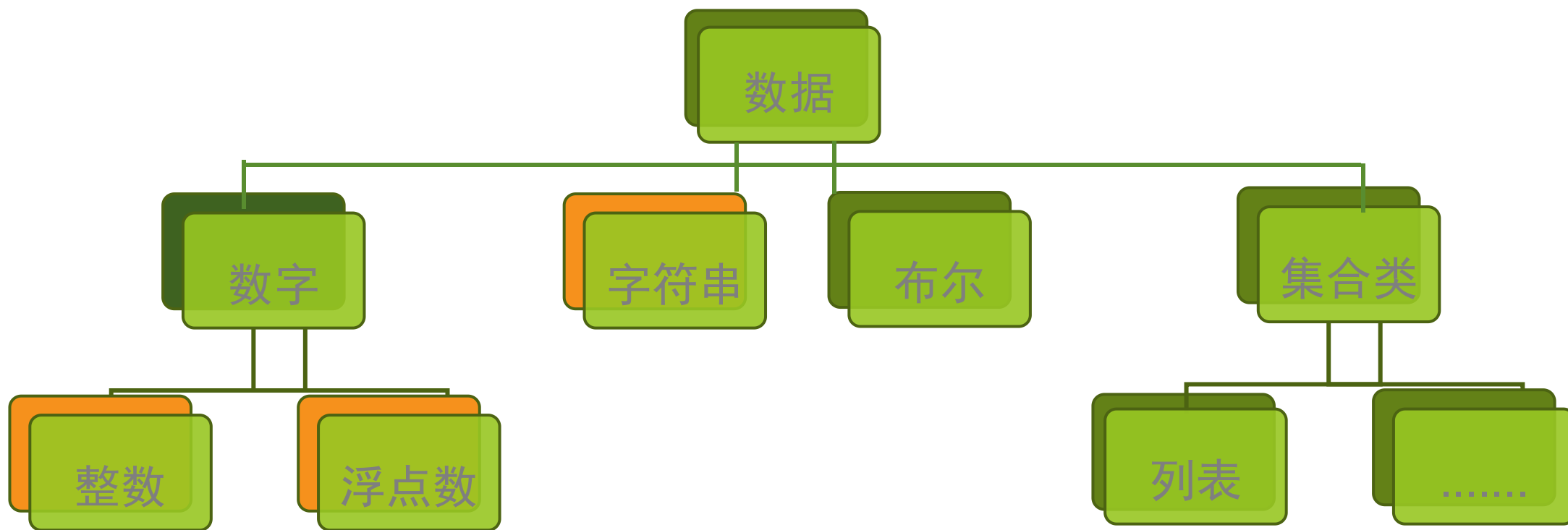


小练习

- 计算87天里，一共包含有多少周，还剩几天？



数据的分类



基本数据类型

● 例：

name = “Jack” → 字符串

age = 10 → 整数

height = 1.53 → 浮点数

number = “3” → ?



数据类型的转换



□ `str()` : 将括号内的数据复制一份加工成一个新的字符串型数据。

□ `int()` : 将括号内的数据复制一份加工成一个新的整数型数据。

□ `float()` : 将括号内的数据复制一份加工成一个新的浮点型数据。



type() 函数

type(): 查看括号内的数据是哪种数据类型?

```
name = "Jack"  
age = 10  
height = 1.53  
n = type(name)  
a = type(age)  
h = type(height)  
print(n)  
print(a)  
print(h)
```

运行结果:

```
=====  
<class 'str'>  
<class 'int'>  
<class 'float'>  
>>>
```



课后任务？

已知变量a中存有字符“123456”，变量b中存有字符“45678”。

任务1：尝试将两个变量中的字符拼接起来，并打印输出，

任务2：将两个变量中的字符转换成整数后，计算他们的和，并打印输出。



数据的输入：input() 函数



- input函数：用于接收来自键盘的数据输入，接收的数据默认是字符串数据类型。

- 例：


```
name = input("请输入你的姓名")  
print("你的姓名是:", name)
```

输出结果：

请输入你的姓名

小明

你的姓名是: 小明



```
name = input()  
这条指令的意  
思是接收数  
据，并把数据  
保存到name中
```

课后练习



有一个农夫家里养了鸡和兔，现在需要你帮助农夫设计一个程序，当农夫数完鸡和兔的个数时，程序能帮助他快速计算出鸡和兔总共共有多少只头？总共共有多少只腿？

运行效果如下

===

请输入鸡的数量

67

请输入兔的数量

34

鸡和兔的头加起来总共是:/101/个

鸡和兔的腿加起来总共是:/270/只



图形用户界面：easygui



- 一、获得easygui模块的方法：

方法1：从<http://easygui.sourceforge.net>下载。

方法2：关注右侧公众号，然后发送关键词“easygui”获取下载链接。



- 二、将下载得到的easygui.py文件，复制到Python安装路径下的Lib文件夹中。

- 三、测试是否安装成功

交互环境中输入命令：`import easygui`

easygui 模块的方法

- ok输入框: `msgbox()`
- 多按钮输入框: `buttonbox()` 方法
- 下拉选择框: `choicebox()`
- 自定义输入框: `enterbox()`
- 整数输入框: `integerbox()`
- 账号密码输入框: `multipasswordbox()`



bool (布尔) 数据类型

- 为什么需要bool数据类型，例：
一件商品是真货还是假货？
小明同学讲的是真话还是假话？
Jack的最后一道题答对了还是答错了？

- 布尔类型的值
True ——> 真
False ——> 假



关系运算符



- Python中常用的关系运算符有：

>: 大于

<: 小于

==: 恒等于

!= : 不等于

>=: 大于等于

<=: 小于等于

- 什么情况下用到关系运算符？

小明的身高>小强的身高吗？ (真/假)

A款鞋价格==B款鞋价格吗？ (真/假)

思考一下，这几个同学那个高，那个矮？



80厘米



1米45厘米



0.95米



1.32米

练习二

- 设计一个程序提示使用者输入一个偶数，然后程序会根据偶数的特征来判别使用者输入的是否为偶数，并输出判断结果。



if 条件语句



● if 语句在程序中的格式

主程序代码

```
if 条件为真:  
    代码....  
    .....
```

主程序代码

.....

运用场景



统计学生成绩.

```
if 成绩>90:
```

```
    评为优秀
```

```
    发放证书
```

```
统计学生成绩
```

```
.....
```

代码块

练习一

- 程序提示使用者输入成绩，如果成绩大于90分，输出“优秀”，然后再输出“请到教务处领取奖状”，默认输出成绩测试。



逻辑运算符



运算符	表达式	说明
and(与)	条件1 and 条件2	只有两个条件都为真， 整体结果才为真
or(或)	条件1 or 条件2	只要1个条件为真， 整体结果就为真
not(非)	not 条件	条件为真时，结果为假 条件为假时，结果为真

课后练习

- 设计一个程序，首先提示用户输入月考总成绩, 然后接着询问是否做完本周作业？（用“是/否”来回答），如果月考总成绩大于280分，并且本周的作业也做完了，就输出“可以玩1个小时游戏”，默认输出“测试是否可以玩游戏”。



运算符的优先级



运算符	运算符符号	归类
小括号	()	
正负号	+x, -x	数学运算
乘法、除法与取余	*, /, %	
加法、减法	+, -	
移位	<< , >>	
按位与	&	位运算
按位异或	^	
按位或		
关系运算	< , <= , > , >= , == , !=	关系运算
同一性测试	Is , is not	
成员测试	In , not in	
逻辑运算（非）	not	逻辑运算
逻辑运算（与）	and	
逻辑运算（或）	or	

练习一

某学校评三好学生的标准是，语数英三门成绩的总分超过280分(含280)，或 语数英三门成绩的总分只超过260分(含260)，但体育成绩超过90分（含90）。请为该学校编写一个测试程序，要求提示学生输入语、数、英、体育四门成绩，成绩达标的则输出“你被评为三好学生，请到教务处领取奖状”，默认输出“评优测试”。



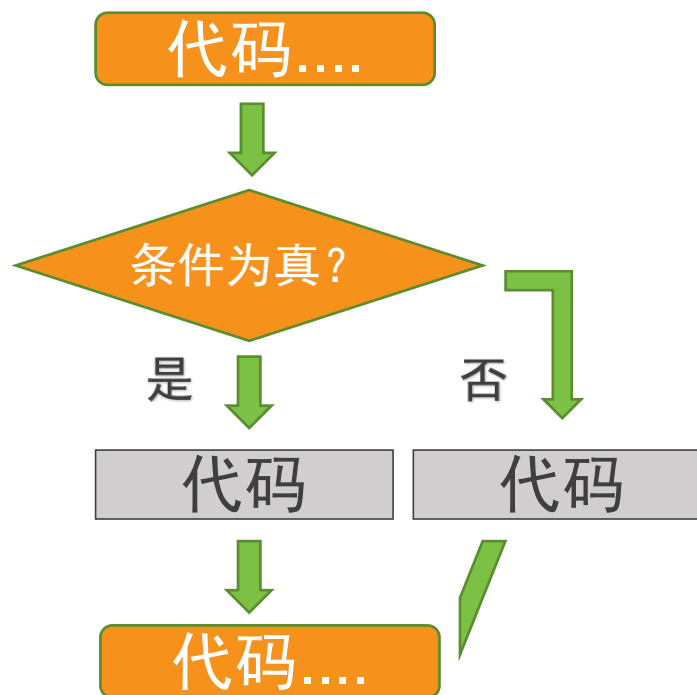
if-else条件语句



- if-else语句格式

```
统计学生成绩.  
if 成绩>90:  
    评为优秀  
    发放证书  
else:  
    继续努力  
    成绩测试...
```

- if-else程序结构

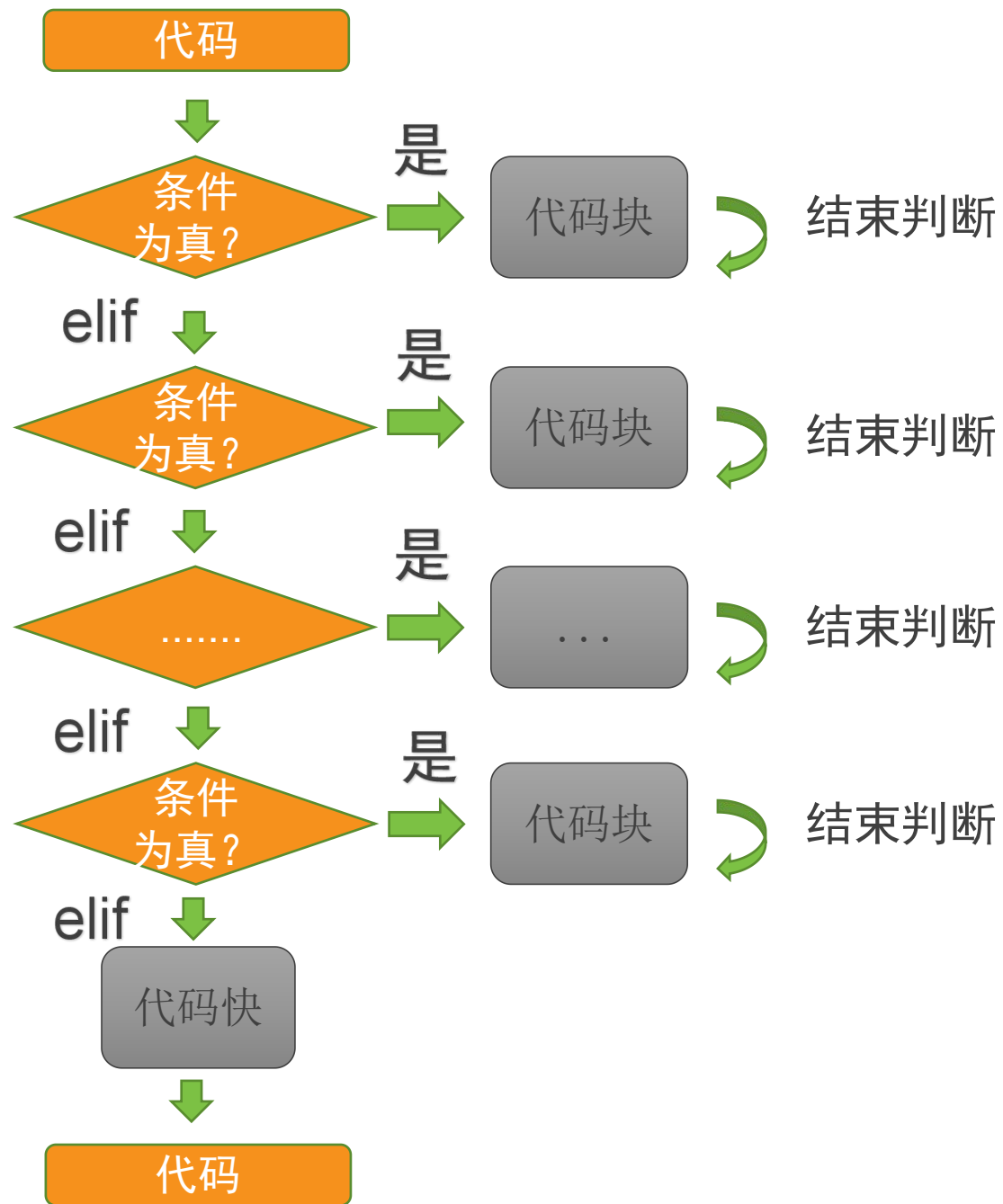


判断闰年

- 设计一个程序，要求任意输入一个年份，自动判断出该年份是否为闰年？已知有满足闰年的条件有两个：
 1. 该年份能够对4整除，但不能对100整除
 2. 该年份直接能被400整除是闰年则输出” 闰年”，不是则输出” 平年”。



多重if结构

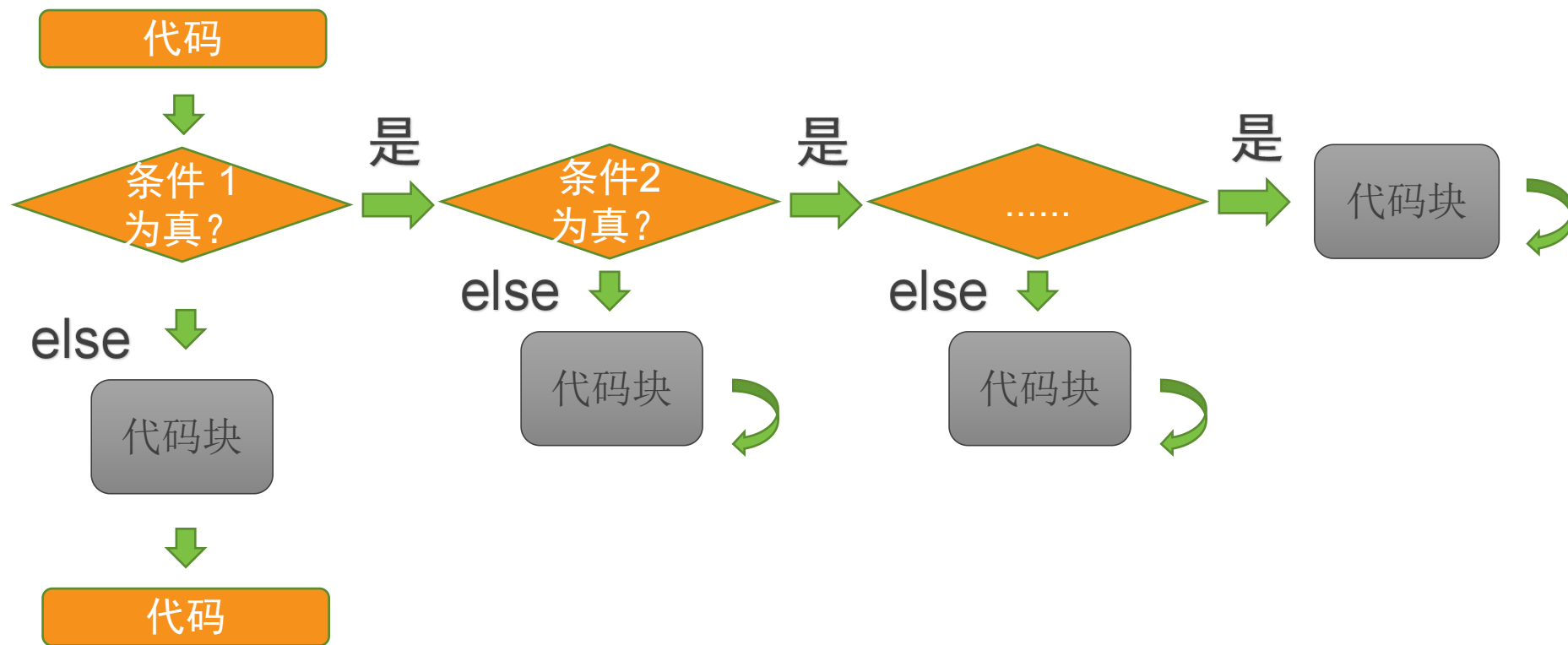


练习

- 一家商场在降价促销。如果购买金额大于500(含500)，则能够享受8折优惠，如果购买金额大于300（含300），则能够享受9折优惠，如果购买金额大于100元（含100），则享受减免10元的优惠。编写一个程序，提示输入购买金额，然后系统显示优惠折扣，和最终需要支付的金额。



if的嵌套结构

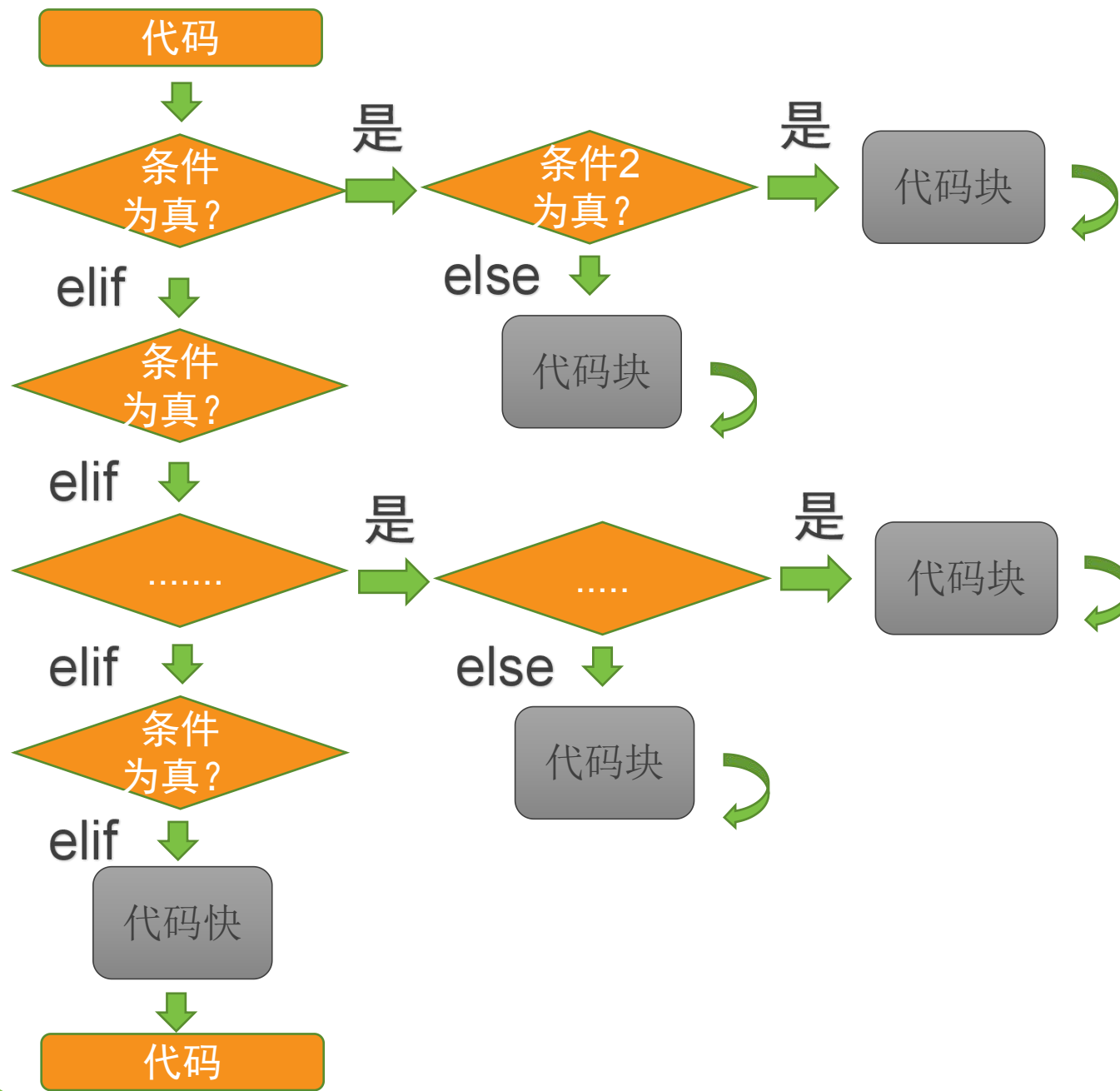


练习



某学校评三好学生的标准是，语数英三门成绩的总分超过280分(含280)，或 语数英三门成绩的总分只超过260分(含260)，但体育成绩超过90分（含90），成绩达标的则输出“你被评为三好学生，请到教务处领取奖状”，否则输出“继续努力”。请为该学校编写一个评三好学生的程序，要求：合理建立程序，如果用户的语数英三门成绩不超过260分，就不用询问体育成绩了。

多重if嵌套结构





项目-猜拳小游戏



程序中的循环结构



- 条件循环：
重复执行至发生某种情况时结束的循环，称为条件循环。
程序用到了” while” 关键字，又称为

while循环

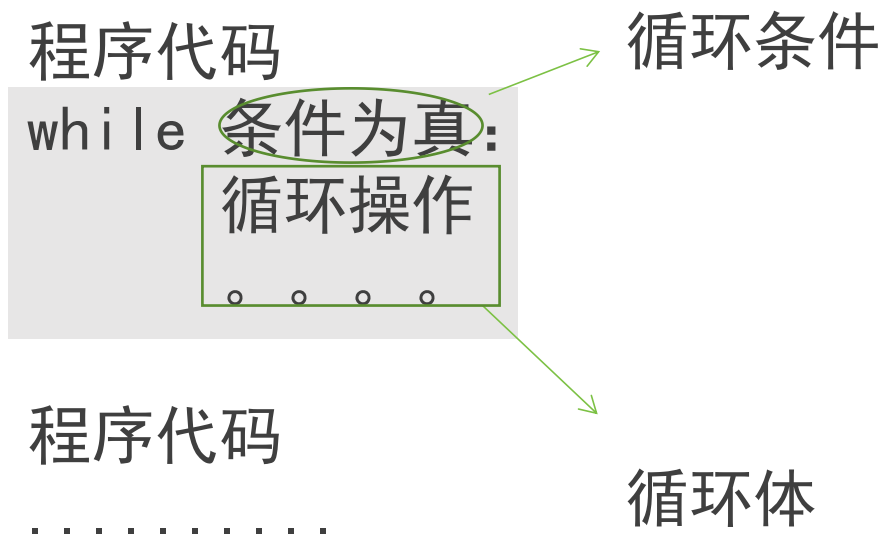
- 计数循环：
重复一定次数的循环，称为计数循环，用了” for” 关键字，又被
称为

for循环

while循环



- while循环在程序中的格式



运用



```
i = 0  
while i < 100:  
    print("好好学习")  
    i = i + 1  
print("程序结束")
```

练习一



- 提示用户输入密码，密码正确则输出“密码正确”，否则提示“密码错误”，“请重新输入”。密码设置为“abc”。

- 输出结果

```
===== RESTART: E:\Python_file\lesson19\tes1.py
请输入密码a
密码错误
请重新输入t
密码错误
请重新输入gff
密码错误
请重新输入abc
密码正确
>>> |
```

练习二



- 求 $1+2+3+4+5+\dots+100=?$

- 输出结果

```
===== RESTART: E:\Python_file\lesson19\test2.py  
求1+2+3+4+5+....+100 的和为 5050  
>>>
```

总结：while循环



● while循环的三要素：循环变量的初值，循环条件，循环变量的改变

`i = 0`#1. 循环变量的初值

`while i < 100:`#2. 循环的条件

`print("好好学习");`

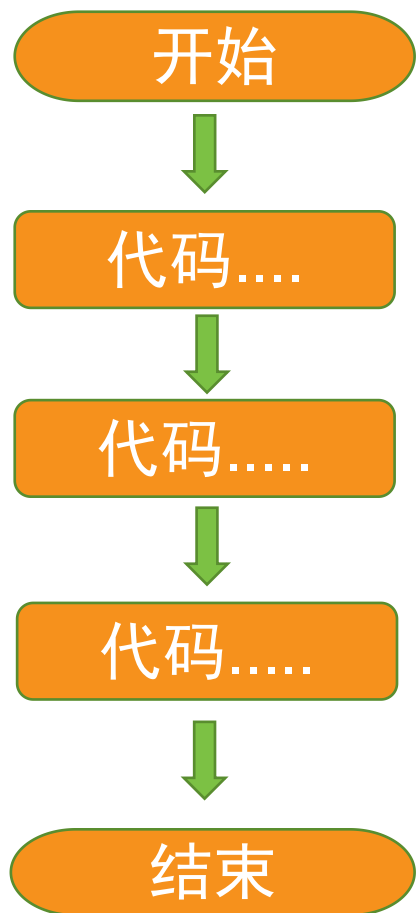
`i = i + 1`#3. 循环变量的增量

● 循环在哪里使用？

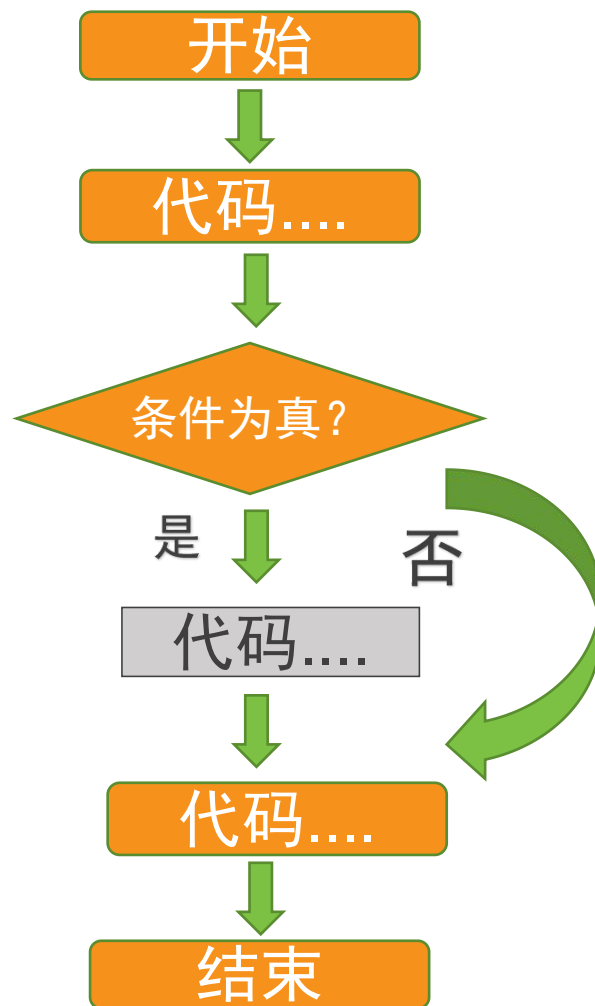
1. 需要重复执行的代码

2. 相近或相似的代码，其中变化是有规律的

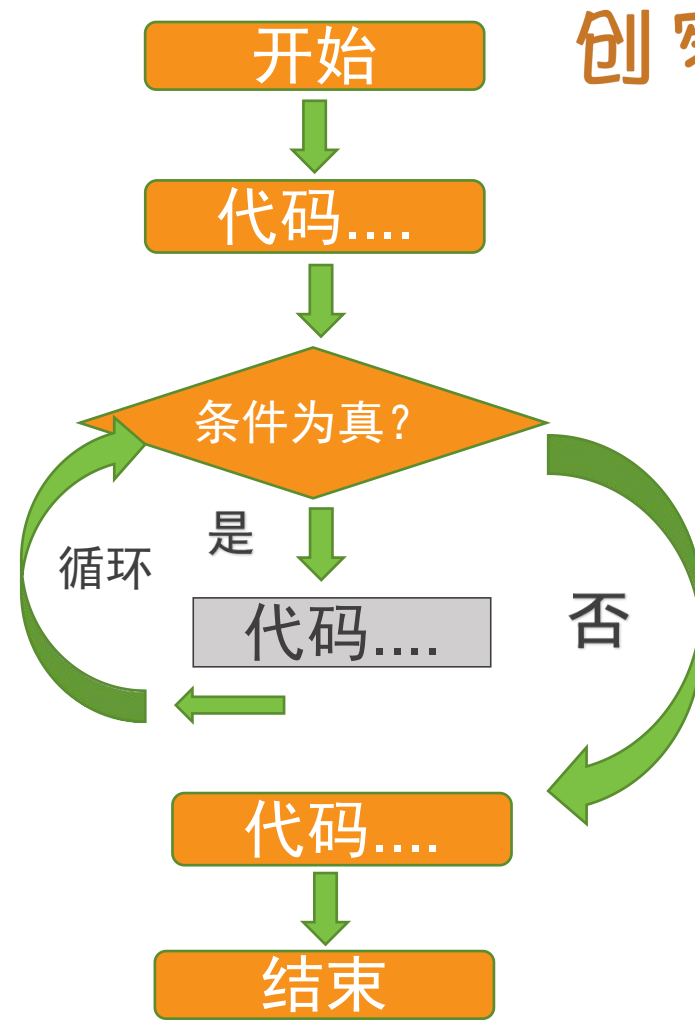
三种程序结构



顺序程序结构



选择程序结构



循环程序结构



项目-猜数字游戏



for循环



- 重复一定次数的循环

- 格式

for 变量 in 可迭代对象: ———> 列表、字符串、元组、字典、集合
循环体代码

- 例

```
for i in [1, 2, 3, 4, 5]:  
    print("hello world")
```

- 运行结果

```
===== RESTART: C:/Users/Administrator/Desktop/test2.py  
hello world  
hello world  
hello world  
hello world  
hello world  
>>>
```

range() 函数



- 功能：生成一定范围的整数

- 格式：

range(start, end, step)

start: 可选 起始值（包含），默认值为0

end: 必选 结尾值（不包含）

step: 可选 步长，默认值为1

- 与for循环配合

```
for i in range(100): #变量i的值循环地从0取到99
    print(i)
```

练习一：打印9的乘法表



● 运行效果

```
===== RESTART: C:/Users/Administrator/Desktop/test.py
1 * 9 = 9
2 * 9 = 18
3 * 9 = 27
4 * 9 = 36
5 * 9 = 45
6 * 9 = 54
7 * 9 = 63
8 * 9 = 72
9 * 9 = 81
>>>
```

课后练习



- 求 $1-2+3-4+5+\dots+999$ 的结果?

- 运行结果

```
===== RESTART: E:\Python_file\lesson22\homework.py
1-2+3-4+...+99= 500
>>>
```

for循环 VS while循环



●for循环：次数循环

1. 场景：常用于事先已经知道循环次数的循环。
2. 优势：程序结构清晰简洁。
3. 劣势：不能做条件循环使用。

●while循环：条件

1. 场景：常用于事先不知道循环次数的循环。
2. 优势：可用于任何循环场景。
3. 劣势：程序结构不如for循环清晰、简洁。

break



- 功能:

break语句可用于while和for循环中，用来终止整个循环。
当有嵌套循环时，break语句只能跳出最近一层的循环。

- 例

```
while True:
```

```
    a = input("请输入一个字符（输入Q或q结束）")
```

```
    if a == "Q" or a == "q":
```

```
        break
```

```
    print(a)
```

```
print("循环结束")
```



continue



- 功能:

continue可用于while和for循环中，用来提前结束一次循环。
当有嵌套循环时，continue只针对最近一层的循环。

- 例

```
for i in range(10):  
    print("测试")  
    if i%2 == 0:  
        continue  
    print(i)
```



else语句



- while、for循环可以附带一个else语句（可选）。如果for、while循环没有被break语句结束，则会执行else语句，否则不执行。

- 语法格式如下：

```
while 条件表达式:  
    循环体
```

```
else:  
    语句块
```

或者：

```
for 变量 in 可迭代对象:  
    循环体
```

```
else:  
    语句块
```

练习二



- 循环录入学员的成绩，如果录入成绩小于0则无效(不参与人数计算和平均分计算)，输入q或Q则直接退出录入系统，最后打印输出学员人数和所有学员的平均成绩。
- 运行效果

```
===== RESTART: C:/Users/Administrator/Desktop/test2.py
请输入学员成绩(输入q或Q则退出)88
请输入学员成绩(输入q或Q则退出)98
请输入学员成绩(输入q或Q则退出)77
请输入学员成绩(输入q或Q则退出)-5
录入的成绩无效
请输入学员成绩(输入q或Q则退出)q
录入完成，退出
一共录入了 3 个学员
学员的平均成绩为: 87.66666666666667
>>>
```

循环的嵌套



- 定义：
一个循环体内可以嵌入另一个循环，一般称为“嵌套循环”，或者“多重循环”。

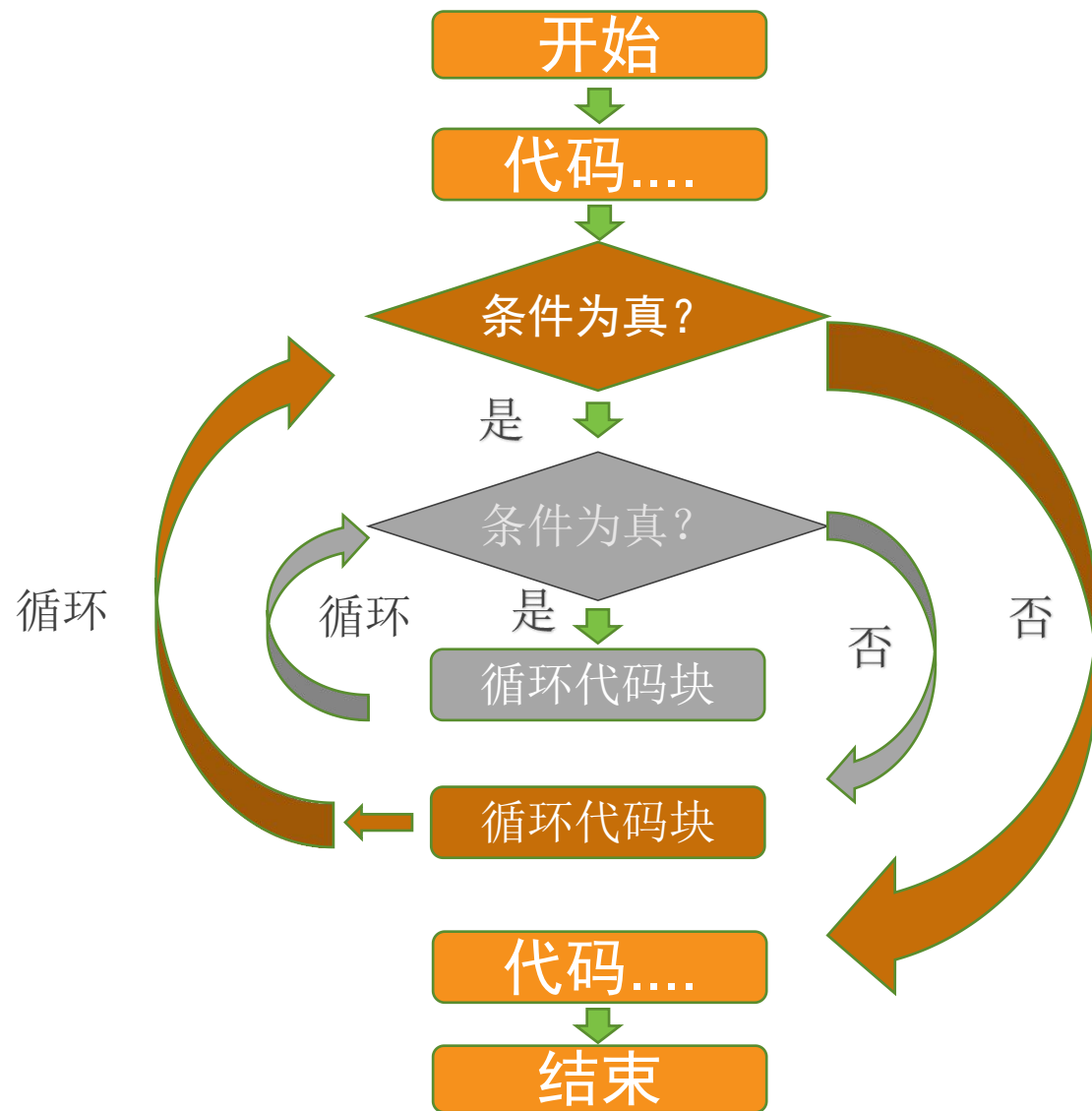
- 例：

```
day = 1  
while day < 8:  
    print("星期", day)
```



```
day += 1
```

循环的嵌套



嵌套循环结构

练习二：



● 打印九九乘法表

1*1=1									
2*1=2	2*2=4								
3*1=3	3*2=6	3*3=9							
4*1=4	4*2=8	4*3=12	4*4=16						
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25					
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36				
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49			
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64		
9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81	

拓展:海龟绘图



- 导入模块:

```
import turtle
```

- 常用方法:

<code>turtle.forward(100)</code>	#画笔向前:100
<code>turtle.left(90)</code>	#设置画笔左转90度
<code>turtle.right(90)</code>	#设置画笔右转90度
<code>turtle.goto(0,50)</code>	#将画笔移动到指定坐标位置
<code>turtle.penup()</code>	#抬笔
<code>turtle.pendown()</code>	#落笔
<code>turtle.circle(100)</code>	#画圆, 半径:100
<code>turtle.write("创客")</code>	#写操作
<code>turtle.color("red")</code>	#设置画笔颜色:红色
<code>turtle.speed(10)</code>	#设置画笔速度 0:最快 10:快 6:中等 3:慢 1:最慢
<code>turtle.width(10)</code>	#设置画笔宽度:10
<code>turtle.fillcolor("red")</code>	#设置填充颜色
<code>turtle.begin_fill()</code>	#开始填充
<code>turtle.end_fill()</code>	#结束填充
<code>turtle.hideturtle()</code>	#隐藏画笔

项目-模拟银行系统



- 设计卡号，密码登录窗口。
- 业务办理：存钱、取钱、查看余额。
- 处理多个账户问题。
- 连续3次登录失败后，系统自锁。

项目-万年历



- 计算1900年1月1日-2019年1月1日一共有多少天？
- 计算1900年1月1日-2019年10月1日一共有多少天？
- 计算1900年1月1日-year年month月1日一共有多少天？
- 根据windows系统日历效果制作万年历。

课程地址：



千聊课堂

网易云课堂



腾讯课堂

爱奇艺课堂



微信公众号

联系我们：



创客童年·公众号



创客童年·肖老师



创客童年·周老师



18900744765 (肖老师)
15115376510 (周老师)

创客教育&少儿编程

——帮助孩子赢在未来

