

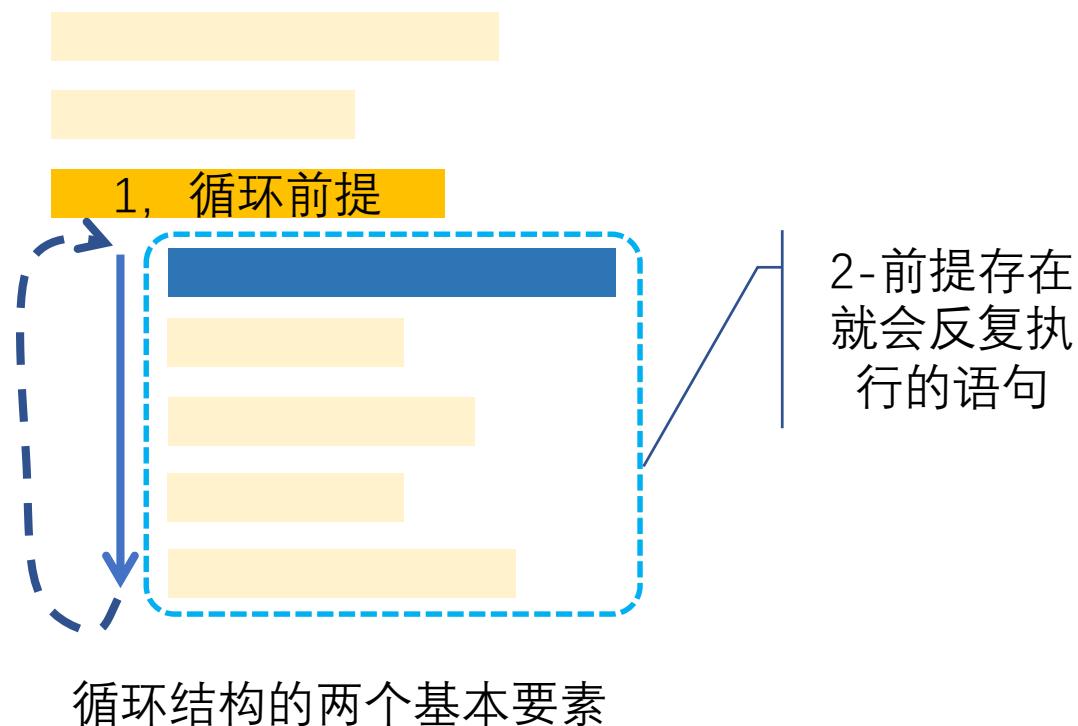
Python程序设计04-基本控制流程

北京大学 陈斌

2019.03.18

目录

- 重复：循环结构
- 迭代循环：for语句
- 常用的迭代数据集
- 绘制数学曲线
- 条件分支：if语句
- Online Judge练习
- 条件循环：while语句
- break和continue



Python语言的几个要件

数据对象和组织

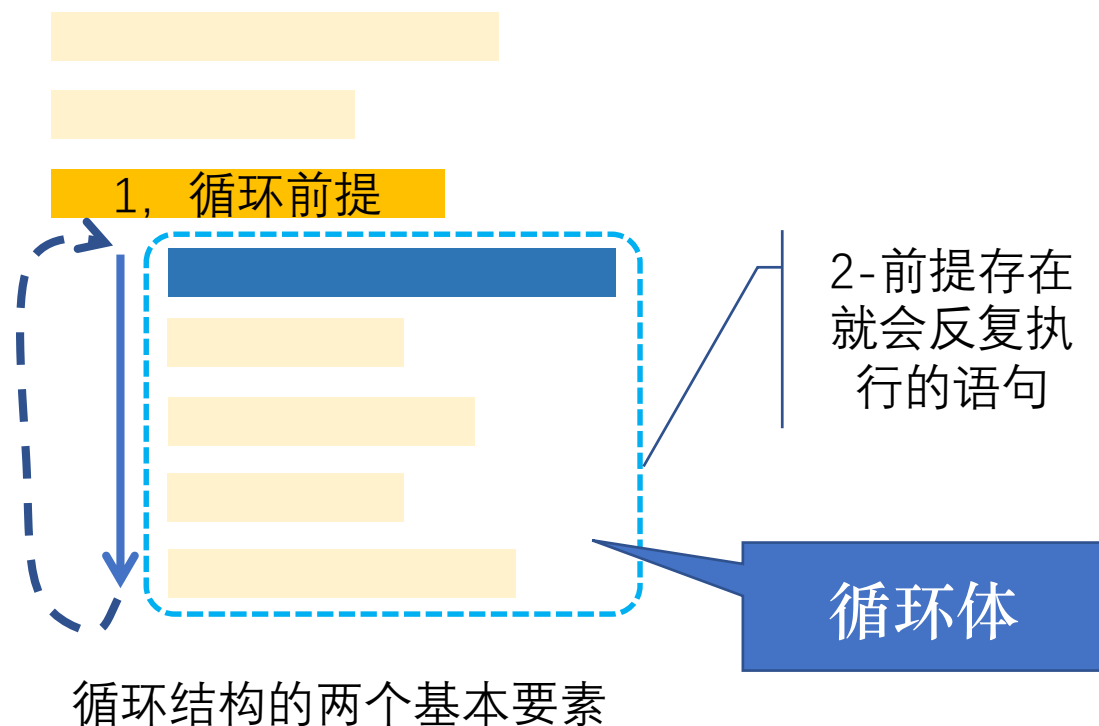
- 对现实世界实体和概念的抽象
- 分为简单类型和容器类型
- 简单类型用来表示值
 - 整数int、浮点数float、复数complex、逻辑值bool、字符串str
- 容器类型用来组织这些值
 - 列表list、元组tuple、集合set、字典dict
- 数据类型之间几乎都可以转换

赋值和控制流

- 对现实世界处理和过程的抽象
- 分为运算语句和控制流语句
- 运算语句用来实现处理与暂存
 - 表达式计算、函数调用、赋值
- 控制流语句用来组织语句描述过程
 - 顺序、条件分支、循环
- 定义语句也用来组织语句，描述一个包含一系列处理过程的计算单元
 - 函数定义、类定义

重复：循环结构（loop）

- 我们需要让计算机反复做设定的任务
- 又能在该停止的时候自动停止重复
- 循环结构具有两个要素
 - 一个循环前提
 - 一组重复执行的语句（**循环体**）
- 只要循环前提成立，**循环体**就会被反复执行



迭代循环：for语句

- 迭代循环语句：for语句
- 循环前提：
 - 一个（或一组）**循环变量**
 - 一个**数据对象集**
- for语句每次从对象集中**取出**一个数据对象，**赋值**给循环变量
 - 如果能取到，就执行一次循环体
 - 循环体中可以使用循环变量
 - 如果取完了，就退出循环

```
forst.py - /Users/chenbin/Documents/教学项目/北大附
s = 0
for i in range(1, 101):
    s = s + i
print("sum:1..100:", s)

a, b, c = 10, -5, 0.5
for x in [-25, 0, 10, 35, 100]:
    y = a * x * x + b * x + c
    print(f"f({x})={y}")

for name in ["Tom", "Jerry", "张三"]:
    print("Hello!", name)
```

常用的数据集：range函数

- range函数可以产生连续整数构成的数据集
- range(end)
 - [0, end)
- range(start, end)
 - [start, end)
- range(start, end, step)
 - [start, end) 步长step
 - 如果step小于0则反向取

range()函数产生一个连续整数的数据集

range(end)

range(start, end)

range(start, end, step)

```
1 print("range:5")
2 for i in range(5):
3     print(i)
4
5 print("range:1,5,2")
6 for i in range(1, 5, 2):
7     print(i)
8
9 print("range:4,1,-1")
10 for n in range(4, 1, -1):
11     print(n)
```

常用的数据集：列表list

- 列表是一种容器数据类型，可以包容多个数据对象
- 整数/浮点数列表
 - [1, 3, 5, 35, -10]
 - [1.23, 34.5, 10.0, 245.7]
- 字符串列表
 - ["Tim", "Jay", "Mary"]
- 混合列表
 - ["Hello", True, 12, 34.56]

```
for name in ["Tom", "Jerry", "张三"]:  
    print("Hello!", name)
```

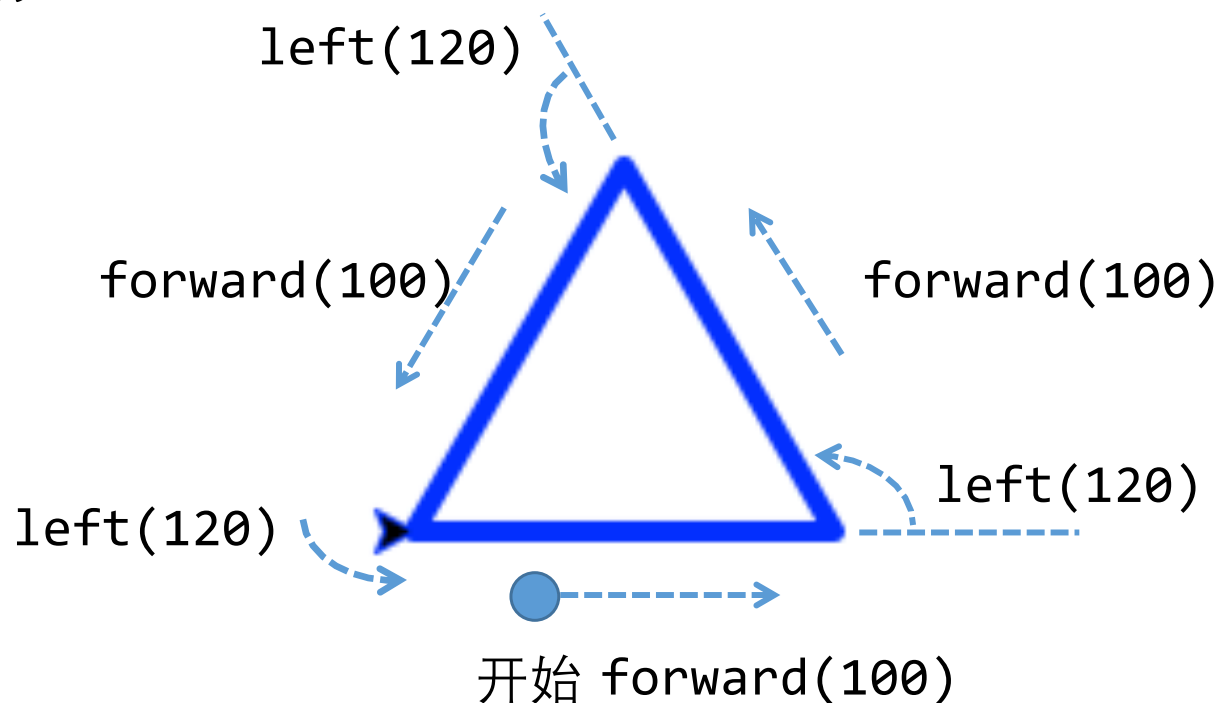
```
for n in [1, 3, 5, 35, -10]:  
    print(f"{n}^2=", n * n)
```

```
m = 1  
for f in [1.23, 34.5, 10.0, 245.7]:  
    m *= f  
print("product:", m)
```

```
for k in [12, 30, 8, 10, 9]:  
    print(f"{k:02d}>", "#" * k)
```

练习：循环语句的应用

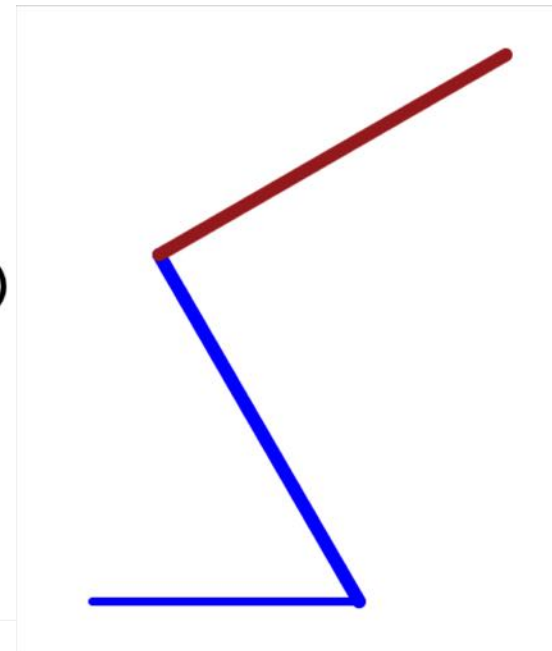
- 如何用循环语句画出等边三角形？
- 如何用循环语句画出一个4种颜色边的正方形？



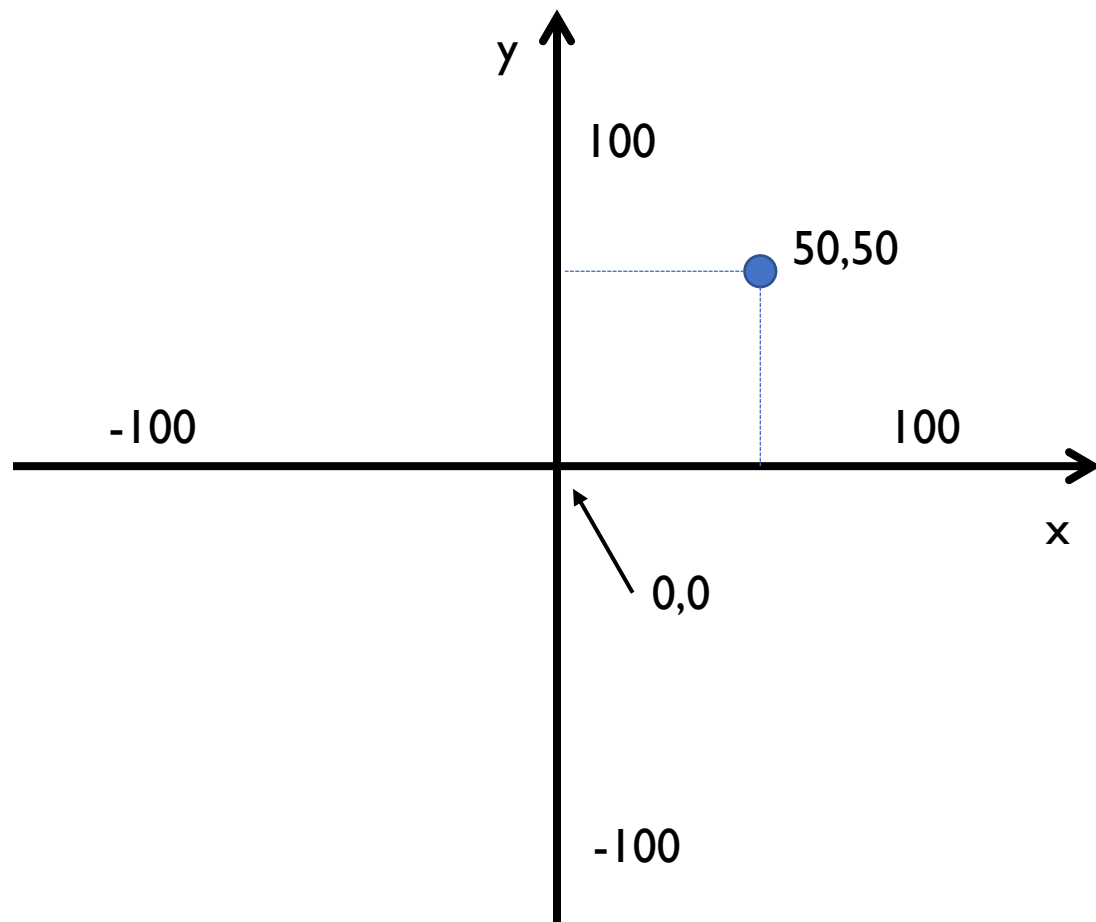
作图程序模版

- 首先，导入turtle模块
- 然后，生成一只海龟
 - 可以做一些初始化设定
- 程序主体：用作图语句绘图
- 最后结束作图
 - 可选隐藏海龟： `t.hideturtle()`

```
1 # 1. 导入海龟模块
2 import turtle
3
4 # 2. 生成一只海龟，做一些设定
5 t = turtle.Turtle()
6 t.color("blue")
7 t.pensize(3)
8
9 # 3. 用海龟作图
10 t.forward(100)
11 t.right(60)
12 t.pensize(5)
13 t.backward(150)
14 t.left(90)
15 t.color("brown")
16 t.forward(150)
17
18 # 4. 结束作图
19 t.hideturtle()
20 turtle.done()
```



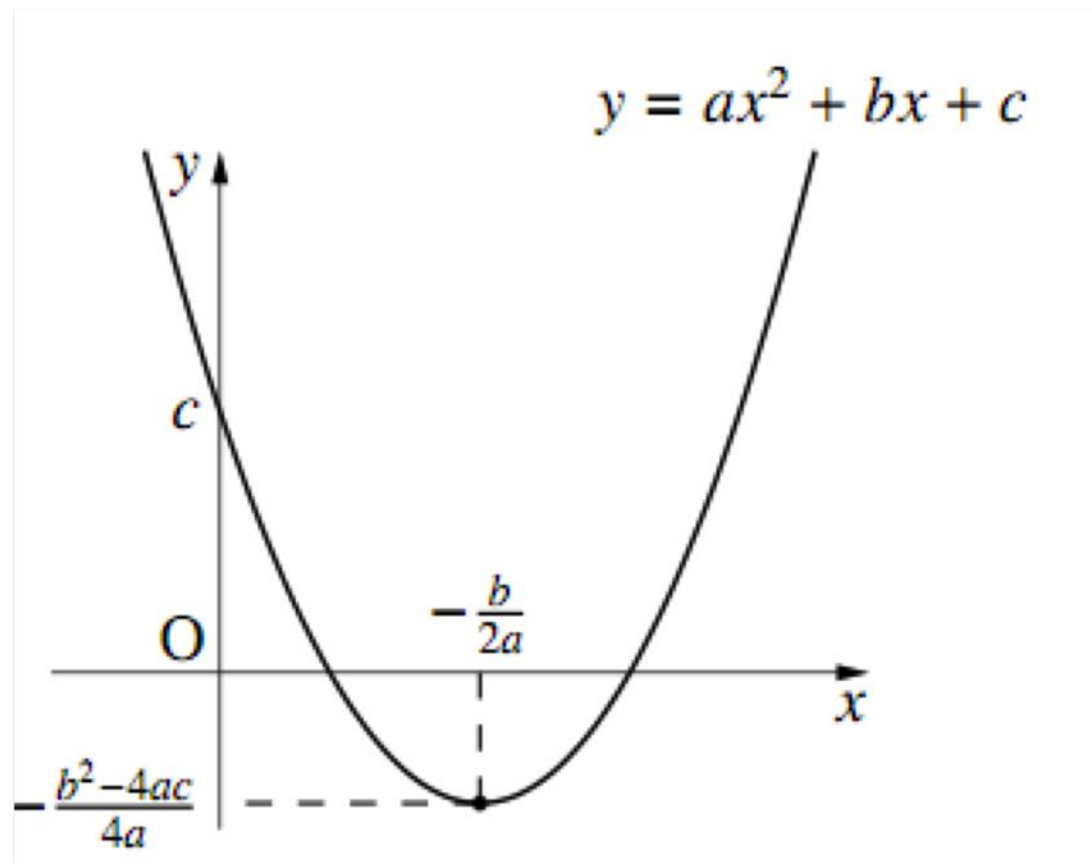
让海龟直接去到某个位置goto(x,y)



- 最开始海龟在 (0,0)
- 可以用goto让海龟去到任何地方
- 用position()获取海龟当前位置
- 如果不希望留下轨迹，则需要先在goto之前调用penup()

平面直角坐标系曲线绘制

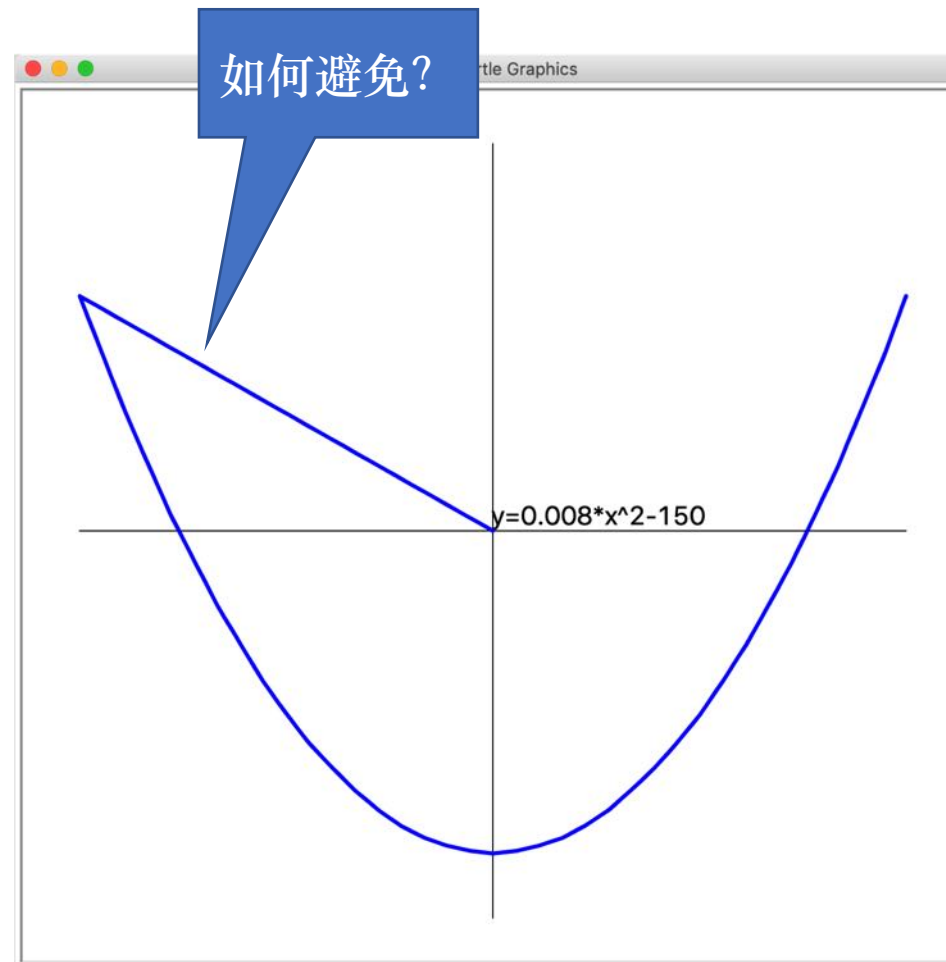
- 如何用循环语句绘制数学曲线？
 - $y=ax+b$ 、 $y=ax^2+bx+c$ 、 $y=\sin(x)$
- 一般步骤
 - 估计 x,y 的范围
 - 设定坐标系：左下角/右上角坐标
 - 画出坐标轴（可选：标注公式）
 - 迭代循环 x ，计算 y
 - goto(x,y)将点连接起来
 - 可以叠加多条曲线



示例：绘制数学曲线

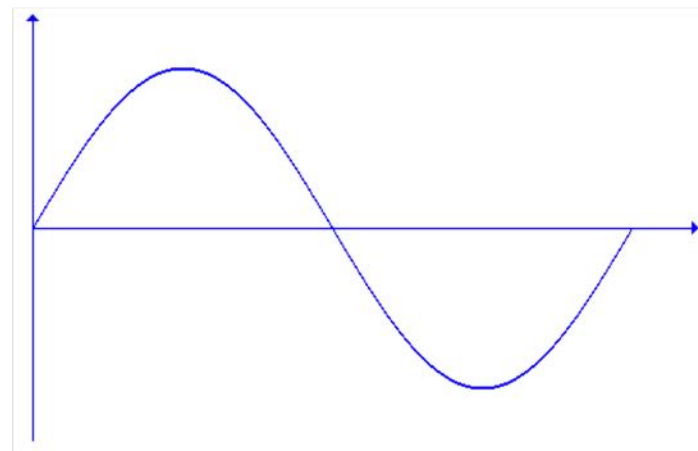
```
1 # 1. 导入海龟模块
2 import turtle
3
4 # 2. 生成一只海龟，做一些设定
5 t = turtle.Turtle()
6 # 设定坐标系：左下角x,y; 右上角x,y
7 turtle.setworldcoordinates(-200, -200, 200, 200)
8
9 # 3. 用海龟作图
10 t.penup()
11 t.goto(-180, 0)
12 t.pendown()
13 t.goto(180, 0)
14 t.penup()
15 t.goto(0, -180)
16 t.pendown()
17 t.goto(0, 180)
18 t.goto(0, 0)
19 t.write("y=0.008*x^2-150", font=("consolas", 20, "normal"))
20 t.pencolor("blue")
21 t.pensize(3)
22 for x in range(-180, 181, 10):
23     y = 0.008 * x * x - 150
24     t.goto(x, y)
25
26 # 4. 结束作图
27 t.hideturtle()
28 turtle.done()
```

如果x是小数
怎么办？



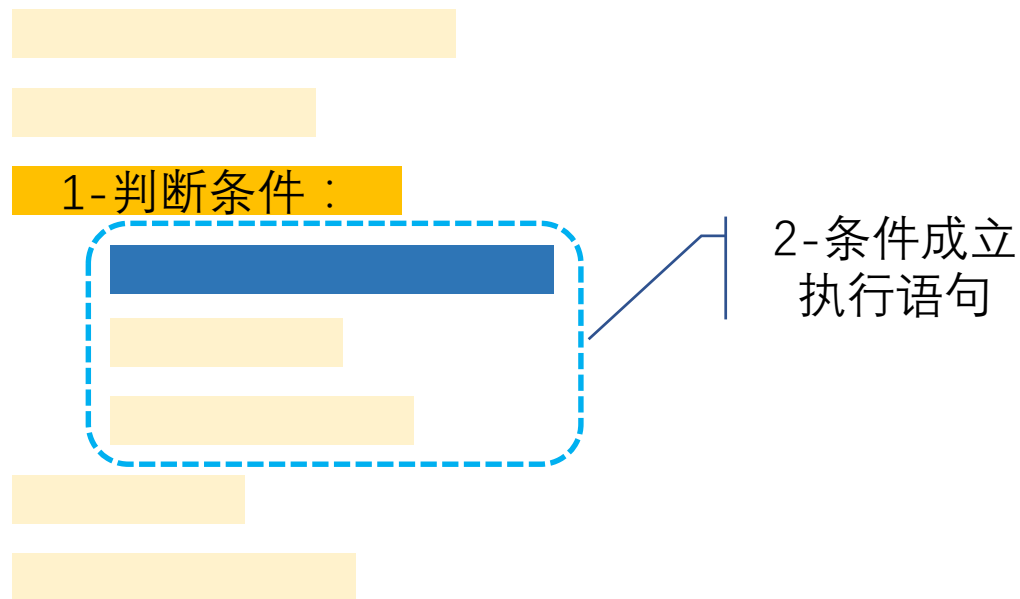
随堂作业：绘制三角函数曲线

- 写一个程序tri.py
- 提交希悦
- 叠加绘制下面3个函数
 - 绿色： $y=\sin(x)$
 - 红色： $y=\cos(x)$
 - 蓝色： $y=2\cos(2x)$
 - x 的范围是 $-2\pi \sim 2\pi$



条件分支结构：if语句

- 让计算机能够自动根据当前的状况来决定执行哪些语句
- 条件分支结构的2个要素
 - 判断条件
 - 一组语句
- if语句首先计算判断条件
 - 如果得到True，就执行这组语句
 - 否则，不执行



条件分支结构两个基本要素

```
# 判断偶数
n = int(input("n="))
print("Your number is", n)
if n % 2 == 0:
    print("It's a even number!")
```

if语句的附加要素： elif和else

- if语句可以附加两个子句
- else子句可以指定在判断条件不成立的时候，要执行的一组语句
- elif子句可以在判断条件不成立的时候，再继续判断另一个条件，相当于else: if

```
# 计算x1和x2之间的距离
x1 = int(input("x1="))
x2 = int(input("x2="))
if x1 > x2:
    d = x1 - x2
else:
    d = x2 - x1
print("distance=", d)
```

```
# 判断年龄
age = int(input("age="))
print("年龄:", age)
if 0 <= age <= 6:
    print("童年")
elif 7 <= age <= 17:
    print("少年")
elif 18 <= age <= 40:
    print("青年")
elif 41 <= age <= 65:
    print("中年")
else:
    print("老年")
```

随堂作业：判断三角形/找最大数

- 程序1 (judge3.py)
 - 输入a,b,c三个整数
 - 输出这三个长度的边是否可以构成三角形
- 程序2 (fmax.py)
 - 输入若干个数，用空格隔开
 - 输出这些数中的最大数

```
# 获取输入一行用空格隔开的整数
numbers = input("some numbers:").split()
numbers = list(map(int, numbers))
print(numbers)
```

```
# 简写
numbers = list(map(int, input().split()))
```


什么是OJ?

- Online Judge，在线测评
- 会有若干组输入输出来判断程序代码是否正确
- 每组输入输出
 - 输入：由input读入
 - 输出：print输出
- 测评机只判断输出的字符串跟标准答案是否相等
 - 输出分行、分隔符号、小数点位数、大小写等等都要**完全一致**

input得到输入的数据

程序代码处理数据

print输出给测评机评判对错

关于Online Judge中Python代码的技巧

- 提示：不要在input里加任何提示符的参数！
- 读入数据：一行就一个值
 - `astr = input()`
 - `n = int(input())`
 - `f = float(input())`
- 读入数据：一行多个整数值
 - `alist = list(map(int, input().split()))`

关于Online Judge中Python代码的技巧

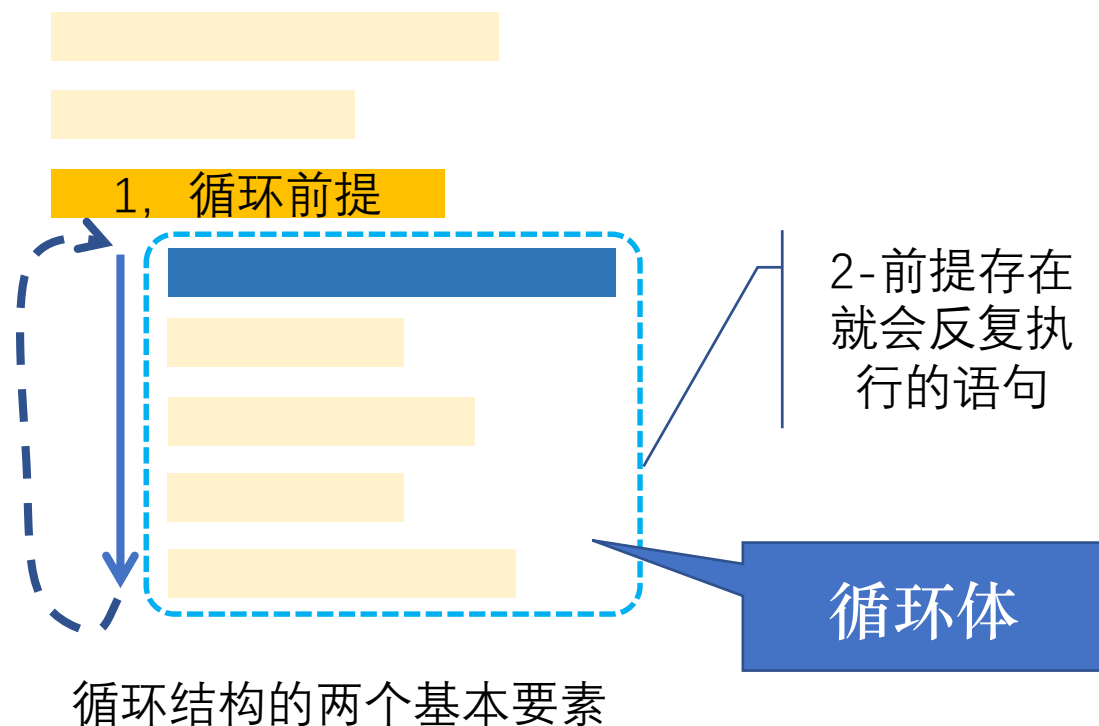
- 输出数据：一行就一个值
 - `print(n)`
 - `print("%.2f", f)` # 小数点后两位
- 输出数据：一行多个值
 - `print(m, n, i)` # 三个整数
 - `print(" ".join(map(str, alist)))`

【H4】完成OJ题：基本表达式

- 请到如下网址完成练习题：
 - <https://vijos.org/d/pkuchenbin/training/5c8e53d5f413620934d099a8>

复习：循环结构（loop）

- 我们需要让计算机反复做设定的任务
- 又能在**该停止的时候**自动停止重复
- 循环结构具有两个要素
 - 一个循环前提
 - 一组重复执行的语句（**循环体**）
- 只要循环前提成立，**循环体**就会被反复执行



条件循环：while语句

- 循环前提是一个判断条件
 - 逻辑类型表达式
- while语句每次都计算表达式
 - 如果结果为“真” True，就执行循环体，然后再计算条件
 - 如果结果为“假” False，就退出循环
- 条件循环一般用在事先不确定的情况
 - 但知道循环什么时候应该结束

```
# 当n等于多少的时候，n阶乘超过10亿？
n = np = 1
while np < 10**9: # 不超过10亿就继续循环
    n = n + 1
    np = n * np
print(n, "阶乘超过10亿，等于", np)

# =====

# 第一个能同时整除2/3/4/5/6的整数是哪个？
n = 1
while not (n % 2 == 0 and n % 3 == 0 \
           and n % 4 == 0 and n % 5 == 0 \
           and n % 6 == 0):
    n = n + 1
print("第一个能同时整除2/3/4/5/6的整数是", n)
```

条件循环和条件分支有什么不同？

```
# 摇几次骰子能摇到6?  
import random  
n = 1  
while random.randint(1, 6) != 6:  
    n = n + 1  
print("摇了", n, "次骰子得到6")
```

```
import random  
n = 1  
if random.randint(1, 6) != 6:  
    n = n + 1  
print("摇了", n, "次骰子得到6")|
```

【练习】验证 $3x+1$ 问题

- 有一个神奇的数学问题叫做“ $3x+1$ ”问题
- 从任意一个正整数开始，重复对其进行下面的操作：
 - 如果这个数是偶数，把它除以2；
 - 如果这个数是奇数，则把它扩大到原来的3倍后再加1。
- 你会发现，序列最终总会变成4, 2, 1, 4, 2, 1, ... 的循环
- 写一个程序x3l.py
- 请用户输入一个正整数n
- 然后打印出整数变化序列
- 直到变成1为止
- 统计n变到1的变化次数step

【随堂作业】 计算机模拟统计概率



- 写一个程序dice.py
- 请用户输入摇骰子次数n
- 让计算机模拟摇骰子n次
- 统计摇到6点的概率是多少?
- 试几个n, 看看这个概率跟n有没有关系?
- 修改dice.py这个程序为dice2.py
- 不需要用户输入
- 让n从100开始, 每次10倍, 直到10亿
- 打印出n和对应的概率
- 看看变化规律?

2层嵌套循环

break语句

- 有时候需要立刻中断循环
- break语句立刻中断退出循环
 - 如果有多个循环嵌套，仅退出直接包含它的那一层循环
- 可以用在for和while循环语句中
- 我们试着用for + break语句重写阶乘的例子：

```
# 当n等于多少的时候，n阶乘超过10亿？
np = 1
for n in range(1, 100):
    np = np * n
    if np > 10**9:
        break
print(n, "阶乘超过10亿，等于", np)
```

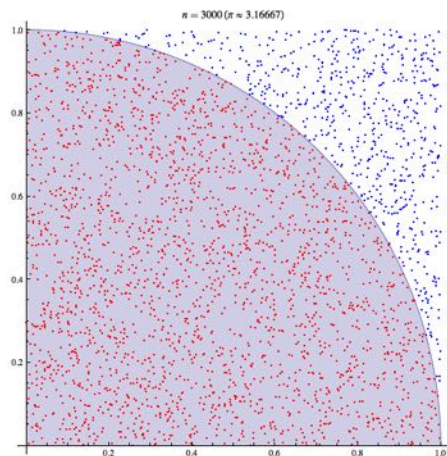
continue语句

- 有时候在执行循环体语句的时候，需要忽略余下的语句，直接跳到下一次循环
- continue语句立刻跳到下一次循环
 - 仅作用于直接包含它的循环语句
- 可以用在for和while语句
- 示例：过7
- 这个例子可以不用continue?

```
# =====  
  
for i in range(1, 100): # 从1到99的报数游戏  
    # 包含数字7，或者能被7整除  
    if ('7' in str(i)) or (i % 7 == 0):  
        print("过! ", end = ', ')  
        continue  
    print(i, end = ', ')
```

【H5】蒙特卡罗法求圆周率

- 蒙特卡罗是一类随机方法的统称。这类方法的特点是，可以在随机采样上计算得到近似结果，随着采样的增多，得到的结果是正确结果的概率逐渐加大



- 编写一个程序mcpi.py
 - `random.uniform(0, 1)`生成0~1之间的随机浮点数
 - 生成1万个落在正方形 (0,0,1,1) 里的点(x,y)
 - 用turtle画出正方形和1/4圆
 - 坐标系设置为(0,0,1,1)
 - `turtle.setworldcoordinates(0, 0, 1, 1)`
 - 把点标注在图形上
 - `t.dot(1, 'red')` # 在当前位置画点
 - 落在圆内用红色，圆外用蓝色
 - 计算出圆周率pi是多少？