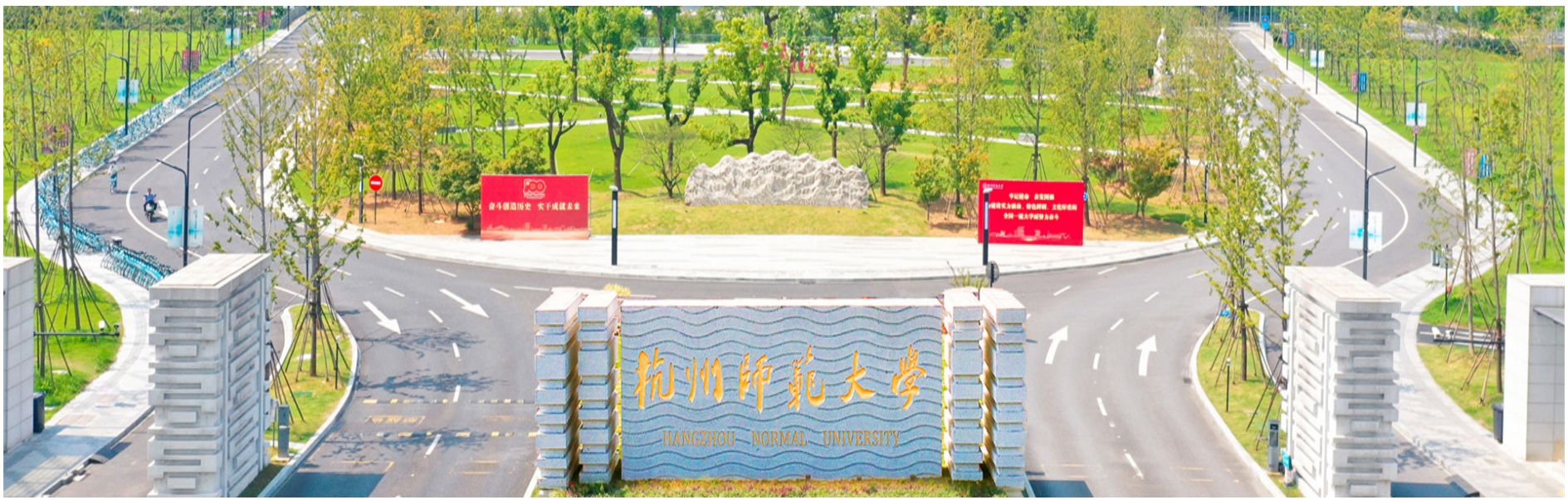
第四讲 - Python 流程控制

张建章

阿里巴巴商学院

杭州师范大学

2023-09



目录 I

1 三种基本控制结构

2 选择结构

3 循环

4 break,continue,pass, 嵌套

5 循环结构与 else 子句搭配

6 课堂实训

7 课后练习

- **顺序结构**：按照书写顺序依次解释执行；
- **选择结构**：按照判断条件选择其中一个分支执行；
- **循环结构**：满足指定条件时重复复执行某些操作。

```
# 顺序结构，先接收键盘输入，再打印输出
age = float(input("Please input your age: "))
print("Your age is {}".format(age))

# 选择结构，年龄小于18，禁止购买香烟
if age < 18:
  print("Sorry, we can not sell cigarettes to you!")
else:
  print("Please input your ID number.")

# 循环结构，如身份证号不是18位数字，则重新输入
while True:
  id_num = input("ID: ")
  if id_num.isdigit() and len(id_num) == 18:
    print('正在与公安系统联网核验......, 请稍后')
    break
```

单选结构

```
# 下面为伪代码，不可直接运行
if condition:
  statements1

other statements
```

当条件判断表达式 (condition) 返回结果为 `True` 时，执行语句块 1 (statements1)，否则 (`False`)，直接顺序往下运行与 `if` 对齐的后续代码，在上例中，如果 condition 计算结果返回 `False` ，则直接执行后续代码 (other statements)。

注意： ① 判断条件 (condition) 必须是一个能返回布尔值 (或等价于布尔值，如 0,1,None) 的表达式，如，比较运算和逻辑运算表达式；② 语句块 1 (statements1) 可以包含任意多行代码。

```python
# 互换两个变量的值，按照从小到大的顺序输出两个值
a = float(input("Please input the first integer: "))
b = float(input("Please input the second integer: "))
print('Before exchange:', a, b)
if a > b: # 条件
  a, b = b, a # 语句块
print('After exchange:', a, b) # 单选结构外的语句

# 寻找180+的男孩子
height = input("Please input your height(cm): ")
if float(height) < 180:
  print("Sorry, you are a good boy, but you know ...")

print('我是单选结构外的后续代码块')
```

双选结构

```
# 下面为伪代码，不可直接运行
if condition:
  statements1
else:
  statements2

other statements
```

当条件判断表达式 (condition) 返回结果为 `True` 时，执行语句块 1 (statements1)，否则 (`False`) 执行语句块 2 (statements)，最后再顺序执行后续的语句块 (other statements)。

注意: 双选结构中 statements1 和 statements2 必须只有一个被运行。

```
# 根据身份证号判断性别，第15-17位数字组成的三位数如果是奇数则为男性
last_four = input("Please input the last four number of your ID:
↪  ")

num = int(last_four[:-1])
if num % 2:
  print('Male')
else:
  print('Female')
```

上述条件 `num % 2` 返回 0 或者 1，等价于 `False` 或 `True`。

多选结构

```
# 下面为伪代码，不可直接运行
if condition1:
  statements1
elif condition2:
  statements2
... ...
elif conditionN:
  statementsN
...
else:
  statements0

other statements
```

满足条件 1 (condition1) 执行语句块 1 (statements1)，满足条件 2 (condition1) 执行语句块 1 (statements2)，……，所有条件都不满足，执行语句块 0 (statements0)，执行完选择结构中的某一个语句块后，立刻离开选择结构，执行后续代码 (other statements)。

```python
# 根据分数输出相应评语
score = float(input("Please input your final score: "))

if 90 <= score <= 100:
  print('Great!')
elif 80 <= score < 90:
  print('Good!')
elif 70 <= score < 80:
  print('You could be better!')
elif 60 <= score < 70:
  print('Dangerous!')
elif 0<= score < 60:
  print("It's a pity!")
else:
  print('The score is invalid!')
```

注意：使用多选结构时，要根据实际问题，全面考虑可能的条件判断，如，上例中，百分制下，合法的分数应该在 0 到 100 之间。

选择结构嵌套：条件满足时（`True`）所执行的语句块 (statements) 也是一个选择结构。

```
amount = float(input("输入驾驶员每100ml血液酒精的含量(mg): "))
if amount < 20:
  print("驾驶员不构成酒驾")
else:
  if amount < 80:
    print("驾驶员已构成酒驾")
  else:
    print("驾驶员已构成醉驾")
```

应用海伦公式和余弦定理的例子

问题: 给定三条边的长度，首先判断这三条边是否能够成一个三角形，如果能，计算三角形面积，并判断其构成的是哪种类型的三角形(锐角、直角、钝角)。

海伦公式: $S = \sqrt{p(p-a)(p-b)(p-c)}$，其中 $p = \frac{a+b+c}{2}$

余弦定理: $cosC = \frac{a^2+b^2-c^2}{2ab}$

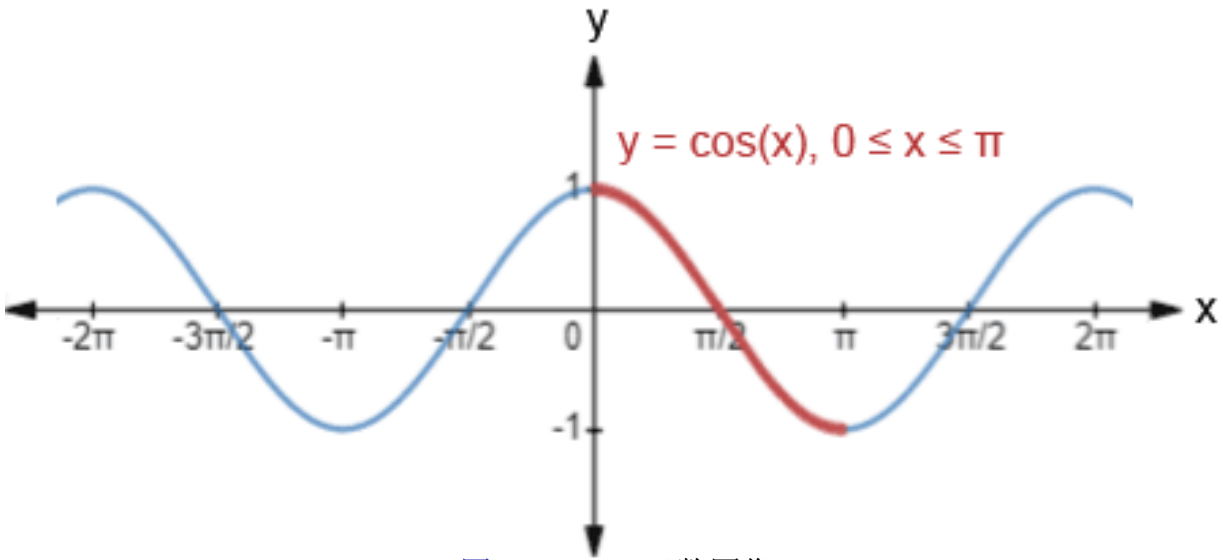


图 1: cosine 函数图像

应用海伦公式和余弦定理的例子 (续)

```
# 将三条边按照长度，从小到大排列
a = 3; b = 12; c = 10
a,b,c = sorted([a,b,c])
# 任意两边之和大于第三边（a<b<c），因此构成三角形
if (a + b) > c:
# 下面用海伦公式用三边求三角形面积
  p = (a + b + c)/2
  area = float((p*(p-a)*(p-b)*(p-c))**(1/2))
  print("The area of triangle is {:.2f}".format(area))
# 根据“同一个三角形内，长边对大角”，以及余弦定理，计算最大角的余弦值
  cos_C = (a**2 + b**2 - c**2)/(2*a*b)
  if cos_C == 0:
    print("Right triangle") # 直角
  elif cos_C < 0:
    print("Obtuse triangle") # 钝角
  else:
    print("Acute triangle") # 锐角
else:
  print("Invalid lengths")
```

课堂练习: 个人所得税计算

题目: 键盘接收年收入 (元为单位)，根据下图税率，使用选择结构计算应缴所得税额。

级数	全年应纳税所得额	税率 (%)
1	不超过 36000 元的	3
2	超过 36000 元至 144000 元的部分	10
3	超过 144000 元至 300000 元的部分	20
4	超过 300000 元至 420000 元的部分	25
5	超过 420000 元至 660000 元的部分	30
6	超过 660000 元至 960000 元的部分	35
7	超过 960000 元的部分	45

图 2: 个人所得税税率

选择结构总结

单选、双选结构都是多选结构的特殊形式，下面以多选结构为例，总结选择结构的要点：

- **顺序判断**：依次判断 condition1，condition2，...，是否满足，只要有一个 condition 满足，则不再进行后续判断，并且执行该 condition 对应的语句块 (statements)，如果都不满足执行 `else` 对应的语句块 (statements0)；
- **运行一次**：运行完 condition 对应的代码块 (statements) 后，立刻离开选择结构，继续执行后续代码块 (other statements)；
- **冒号、缩进、对齐**：一定记得写冒号，冒号下的代码要缩进，同一层级的代码要对齐。

while 循环

```
# 下面为伪代码，不可直接运行
while condition:
  statements

other statements
```

当条件判断表达式 (condition) 返回结果为 `True` 时，执行语句块 (statements)，再次计算 condition，如果结果返回 `True`，则继续执行 statements，循环往复...，否则 (`False`)，跳出循环，顺序往下运行与 `while` 对齐的后续代码 (other statements)。

注意：① 判断条件 (condition) 必须是一个能返回布尔值 (或等价于布尔值，如 0,1,None) 的表达式，如，关系运算和逻辑运算表达式；② 语句块 (statements) 可以包含任意多行代码；③ 选择结构只计算一次 condition，`while` 循环计算多次 condition，每次运行完语句块 (statements) 就立刻再次判断 condition 是否满足。

```python
# 笨办法求1+2+3+ ...+100的和
result = 0
number = 1
while number < 101:
  result += number
  number += 1 # number = number + 1
print(result)

# 输出小于10的奇数
a = 1
while a < 10:
  print(a)
  a += 2
print('我就是与while对齐的后续代码^_^')
```

注意: 上述两例中，`while` 循环中的 statements 中都对 condition 中的变量进行了修改，才使得 condition 可以返回 `False`，程序跳出 `while` 循环，从而执行后续代码 (other statements)。

for 循环

```
# 下面为伪代码，不可直接运行
for item in iterables:
  statements
```

通过遍历可迭代对象 (iterables) 中的元素 (item)，来控制语句块 (statements) 执行的次数。典型的可迭代对象有序列 (列表，元组，字符串)、集合、字典、range、zip、enumerate 等。

```
for char in '中华民族伟大复兴':
  print(char)
for char in list('中华民族伟大复兴'):
  print(char)
for char in tuple('中华民族伟大复兴'):
  print(char)
for char in set('中华民族伟大复兴'):
  print(char)
```

内置函数 range

`range` 和 `enumerate` 是两个经常与 `for` 循环搭配使用的内置函数，典型用法如下例:

```
# 打印出1-9范围内的数字
for i in range(1,10):
  print(i)
# 以3为步长打印出1-9范围内的数字
for i in range(1,10,3):
  print(i)
```

`range(start, end, step)` 返回一个 range 对象，使用内置函数 `list` 可将该 range 对象转化为列表，列表会包含从 `start` 到 `end-1` 之间的整数，步长为 `step`。此处的 `start`、`end` 和 `step` 与列表切片所使用的参数非常像，**都是有头没有尾、带步长就跳、正负步长分别对应从左到右和从右到左，对应错误则结果为空。**

内置函数 enumerate

```
for idx,num in enumerate(range(48,59)):
  print(idx,num)
```

内置函数 `enumerate` 接收一个可迭代对象为参数，返回一个 enumerate 对象，通过 `list` 函数可将该对象转化为列表，列表中包含的元素均为形如 `(index,element)` 的二元组，element 表示可迭代对象中的一个元素，index 为该元素在可迭代对象中的索引 (index 本质上是从 0 开始的计数器，看下面例子)。

```
for idx, num in enumerate(set([1,2,3])):
  print(idx,num)
```

集合无序，上例中 idx 不表示索引，表示 num 是第几个被取出来的。

注意: 在使用 `range` 和 `enumerate` 时，不是必须要在其外层套一个 `list` 函数，套 `list` 函数便于查看他们的内容。

课堂练习

题目 1：使用 `for` 循环遍历字典中的键值对。

```
# 题目1
actors = ['沈腾','马丽','艾伦']
actor_dict = dict([(101+idx,name) for idx,name in
↪  enumerate(actors)])
for idx, name in actor_dict.items():
  print(idx,name)
```

题目 2: 计算 1~1000 范围内 3 的倍数和 7 的倍数的数字之和。

题目 3: 水仙花数是一个 3 位数，它的每个位上的数字的 3 次幂之和等于它本身 (如: $1^3 + 5^3 + 3^3 = 153$)，求所有水仙花数。

课堂练习-答案

```python
# 题目2
sum = 0
for i in range(1, 1001):
  if i % 3 == 0:
    sum += i
  elif i % 7 == 0:
    sum += i
print(sum)

# 题目3
for i in range(100, 1000):
  sum = 0
  for j in str(i):
    sum += int(j)**3
  if sum == i:
    print(i)
```

`break`、`continue`、`pass`都是 Python 的保留关键字，用法如下：

- `break`：只能出现在循环结构 (for 循环, while 循环) 中的语句块 (statements) 中，程序一旦运行到`break`，立刻跳出循环，继续执行后续语句 (other statements)；
- `continue`：只能出现在循环结构 (for 循环, while 循环) 中的语句块 (statements) 中，程序一旦运行到`continue`，立刻结束本次循环，进入下次循环；
- `pass`：可以单独出现在任何一行，主要作用是保证代码格式的规范和完整性，程序一旦遇到`pass`，什么操作也不执行。

注意: ① 当`break`和`continue`出现在嵌套循环中时，只作用于它所在的那一层循环；② `break`、`continue`、`pass`都是单独出现在某一行，同一行没有其他代码。

```python
vowel = 'aeiou'

# 打印输出br
for char in 'brown':
  if char in vowel:
    break
  print(char)

# 打印输出brwn
for char in 'brown':
  if char in vowel:
    continue
  print(char)

# 打印输出brwn
for char in 'brown':
  if char in vowel:
    pass
  else:
    print(char)
```

题目: 对英文句子去做去元音化操作 (Disemvoweling)，即，将单词中非首尾元音字母去除，这是短信 (SMS) 语言的典型特征，阅读起来需要的认知努力很小。

```
vowel = 'aeiou'
sent = 'The quick brown fox jumps over the lazy dog'
word_list = []

for word in sent.strip().split():
  char_list = []
  for idx,char in enumerate(word):
    if char in vowel and idx not in [0, len(word) -1]:
      pass
    else:
      char_list.append(char)
  word_list.append(''.join(char_list))
print(' '.join(word_list))
# The qck brwn fx jmps ovr the lzy dg
```

```
# 下面为伪代码，不可直接运行
while condition:
  statements1
else:
  statements2
other statements
```

```
# 下面为伪代码，不可直接运行
for item in iterables:
  statements1
else:
  statements2
other statements
```

在循环未经 break 打断的情况下，整个循环完成时执行 else 子句下的语句块 (statements2)，再执行后续代码 (other statements)。

题目: 找出 [2, 100] 中的素数。素数 (质数)，指在大于 1 的自然数中，除了 1 和该数自身外，无法被其他自然数整除的数。

```
for i in range(2, 101):
  for j in range(2, i):
    if i % j == 0:
      break
  else:
    print(i, ' 是素数')
```

题目：寻找整数 `x` 的最大因子 (除 1 和自身外)。

```
x = int(input('Please input a integer bigger than 1: '))

for num in range(x-1,1,-1):
  if x % num == 0:
    print('The biggest factor is {}'.format(num))
    break
else:
  print('{} is a prime number'.format(x))
```

题目：求两个数的最大公约数。

```
a = int(input("Please input the first number: "))
b = int(input("Please input the second number: "))
for i in range(min(a,b), 1,-1):
  if a%i == 0 and b%i == 0:
    print("{}和{}的最大公约数是{}".format(a,b,i))
    break
else:
  print("{}和{}是互素数".format(a,b))
```

题目：最多三次密码输入机会。

```
PASSWORD="123456"
max_times = 3
flag = False
for i in range(3):
  pwd = input("Please input the password: ")
  if pwd == PASSWORD:
    flag = True
    break
  else:
    print("Come on baby, try again, {} times
    ↪  left.".format(max_times - i - 1))
if flag:
  print("Great, Baby!")
else:
  print("Sorry, Baby, See you tomorrow!")
```

题目: 1-100 之间不是 7 的倍数的数字之和。

```
sum = 0
for i in range(1, 101):
  if i%7 == 0:
    continue
  else:
    sum += i
print(sum)
```

题目: 思考下面代码的运行结果。

```
i = 0
sum = 0
while i < 5:
  i += 1
  if i == 3:
    continue
  sum += i
print(sum)
```

题目: 输出字符串中的每个字符，h 除外。

```
for letter in 'Python':
  if letter == 'h':
    pass
  else:
    print('Current letter is {}'.format(letter))
```

题目：输出 100 之内的所有素数，每行 10 个依次输出。

```
tmp_list = []
for i in range(2, 100):
  for j in range(2, i):
    if i % j == 0:
      break
  else:
    tmp_list.append(str(i))
  if len(tmp_list) == 10:
    print(', '.join(tmp_list))
    tmp_list.clear()
```

题目: 输出如下图所示的九九乘法表。

```
1x1=1
1x2=2   2x2=4
1x3=3   2x3=6   3x3=9
1x4=4   2x4=8   3x4=12  4x4=16
1x5=5   2x5=10  3x5=15  4x5=20  5x5=25
1x6=6   2x6=12  3x6=18  4x6=24  5x6=30  6x6=36
1x7=7   2x7=14  3x7=21  4x7=28  5x7=35  6x7=42  7x7=49
1x8=8   2x8=16  3x8=24  4x8=32  5x8=40  6x8=48  7x8=56  8x8=64
1x9=9   2x9=18  3x9=27  4x9=36  5x9=45  6x9=54  7x9=63  8x9=72  9x9=81
```

```python
for i in range(1, 10):
  for j in range(1, i+1):
    print('{}x{}={}\t'.format(j, i, i*j), end='')
  print()
```

题目：使用 `*` 作为填充物打印输出如下图所示的高度为 10 的等腰三角形。

```
*
***
*****
*******
*********
***********
*************
***************
*****************
*******************
```

```
row = 10
for i in range(row):
  for _ in range(row - i - 1):
    print(' ', end='')
  for _ in range(2 * i + 1):
    print('*', end='')
  print()
```

题目 1: 用蒙特卡罗法计算圆周率 π 的值。

蒙特卡罗方法是指用随机数通过求解概率而获得近似值的方法。

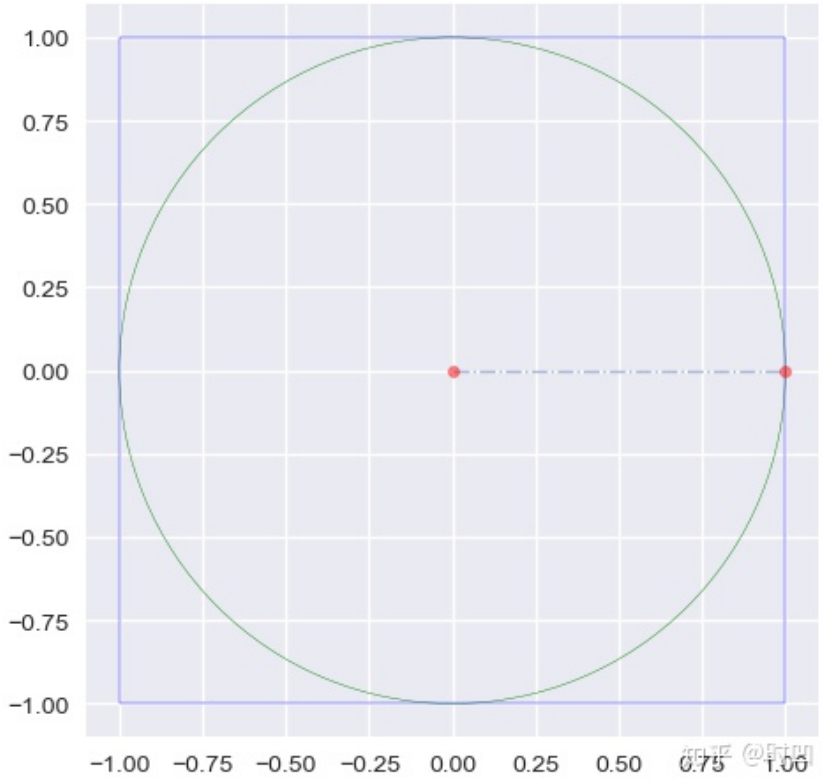


图 3: 以原点为圆心的单位圆

请先思考从上图中随机取一个点，这个点落在单位圆内 (包含边界) 的概率是多少。

假设圆的半径为 r，圆的面积为 $S_{circle} = \pi r^2$，其外接正方形的面积为 $S_{square} = (2r)^2 = 4r^2$

那么，对外接正方形里的任一一个点，它落在圆里的概率为：

$$P(in\ circle) = \frac{S_{circle}}{S_{square}} = \frac{\pi}{4}$$

通过上述公式，转换得：

$$\pi = 4 \times P(in\ circle) \approx 4 \times \frac{m}{n}$$

上式中使用频率近似概率，从外接正方形中随机采样 n 个点，其中有 m 个点落在了圆内。

落在圆内的点满足如下条件：

$$(x - x_0)^2 + (y - y_0)^2 <= r^2, (x_0, y_0)\text{为圆心坐标}$$

下面以图3中的圆为例，半径为 1，圆心为原点，使用蒙特卡罗法估算圆周率。

```
import math
import random
hit = 0 # 点落在圆内的次数
number_of_trials = 1000000 # 实验次数要足够大
for i in range(number_of_trials):
  # 从外接正方形中随机选择一个点
  x = random.uniform(-1.0,1.0) # 按照均匀分布取横坐标值
  y = random.uniform(-1.0,1.0) # 按照均匀分布取纵坐标值
  # 如果点落在圆内，增加一次计数
  if x**2 + y**2 <= 1:
    hit += 1
print('math.pi is {:>30.8f}'.format(math.pi))
print('The estimated one is
↪  {:>20.8f}'.format(4*(hit/number_of_trials)))
```

题目 2： 用微元法计算定积分 $\int_0^1 x^2\,dx$ 的值 $(\frac{1}{3})$。

微元法: 将积分值表示的面积看做无穷多个小矩形的面积之和 (即，分割、近似、求和)。

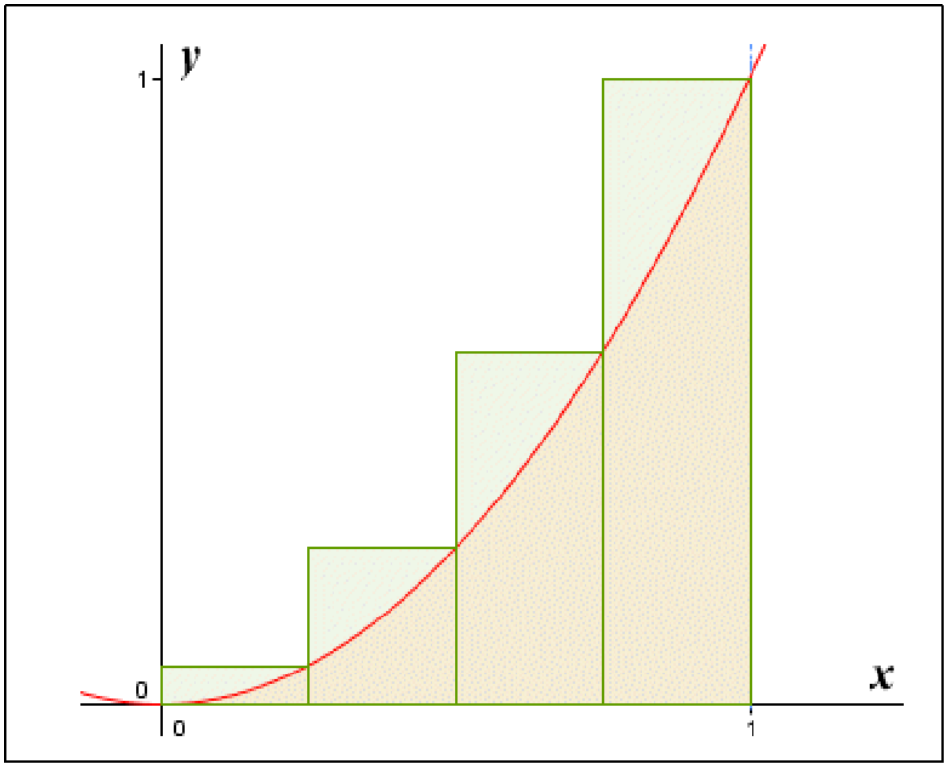


图 4: $f(x) = x^2$ 的函数曲线

下面使用循环结构，将上图中函数曲线下方不规则图形分割为足够多个小矩形，用这些小矩形的面积和近似计算积分值。

```python
import math

N = 1000000 # 分割为1000000个小矩形

sum_of_area = 0
i = 1
while i <= N:
  small_area =  (1/N) * (i/N)**2 # 宽×高，求小面积
  sum_of_area += small_area # 小面积累加
  i += 1
print(sum_of_area)
```

THE END