

python语言入门与实践

习题课2：用python计算

主讲人：吴陈炜 博士

复旦大学智能机器人研究院 工程博士

北京大学信息科学与技术学院 理学硕士

浙江大学信息与电子工程学系 工学学士

网易（杭州）网络有限公司 项目经理

美国项目管理协会 PMP（项目管理专业人士）

之江实验室 科研主管

2020 年 5 月 12 日

上节回顾

- 变量的概念和使用
- python中的数据类型
- 字符串数据类型
- 数值型数据数据类型

思考题

- 根据王者荣耀伤害结算公式，制作一个普攻伤害计算函数，并且计算李白在物攻834、物理穿透300|45%的状态下普攻物抗为1588的项羽时造成的伤害。并且打印伤害提示“李白对项羽造成了XX点伤害”

- 参考公式：

$$damage = attack * (1 - remission)$$

$$remission = \frac{real_defence}{602 + real_defence}$$

$$real_defence = (nominal_defence - penetration) * (1 - percentage)$$



把数学语言翻译成程序语言

```
1  attack = 834
2  defence = 1588
3  penetrate = 300
4  percentage = 0.45
5  r_defence = (defence-penetrate)*(1-percentage)
6  remission = r_defence/(602+r_defence)
7  damage = int(attack*(1-remission))
8  print(f"李白对项羽造成了{damage}点伤害")
9
```

```
李白对项羽造成了383点伤害
[Finished in 0.2s]
```

注意变量先定义后使用

使用未定义的变量会报错

```
Traceback (most recent call last):
  File "C:\Users\wuchenwei\Desktop\pythoncrashcourse\D3\honor.py", line 5, in <module>
    damage = int(attack*(1-remission))
NameError: name 'remission' is not defined
[Finished in 0.1s with exit code 1]
```



```
honor.py  x  testmath.py  x
1  attack = 834
2  defence = 1588
3  penetrate = 300
4  percentage = 0.45
5  damage = int(attack*(1-remission))
   File "C:\Users\wuchenwei\Desktop\pythoncrashcourse\
6  r_defence = (defence-penetrate)*(1-percentage)
7  remission = r_defence/(602+r_defence)
8  # damage = int(attack*(1-remission))
9  print(f"李白对项羽造成了{damage}点伤害")
10
```

python是一门优雅的语言

Python报错:



C++报错:



字符串的习题

动手试一试

在做下面的每个练习时，都编写一个独立的程序，并将其保存为名称类似于 `name_cases.py` 的文件。如果遇到了困难，请休息一会儿或参阅附录 C 提供的建议。

2-3 个性化消息：将用户的姓名存到一个变量中，并向该用户显示一条消息。显示的消息应非常简单，如 “Hello Eric, would you like to learn some Python today?”。

2-4 调整名字的大小写：将一个人名存储到一个变量中，再以小写、大写和首字母大写的方式显示这个人名。

2-5 名言：找一句你钦佩的名人说的名言，将这个名人的姓名和他的名言打印出来。输出应类似于下面这样（包括引号）：

Albert Einstein once said, “A person who never made a mistake never tried anything new.”

2-6 名言 2：重复练习 2-5，但将名人的姓名存储在变量 `famous_person` 中，再创建要显示的消息，并将其存储在变量 `message` 中，然后打印这条消息。

2-7 剔除人名中的空白：存储一个人名，并在其开头和末尾都包含一些空白字符。务必至少使用字符组合 `"\t"` 和 `"\n"` 各一次。

打印这个人名，以显示其开头和末尾的空白。然后，分别使用剔除函数 `lstrip()`、`rstrip()` 和 `strip()` 对人名进行处理，并将结果打印出来。

```
12 print(help(str.strip))
```

Help on method_descriptor:

```
strip(...)
    S.strip([chars]) -> str
```

Return a copy of the string S with leading and trailing whitespace removed.

If chars is given and not None, remove characters in chars instead.

```
None
[Finished in 0.2s]
```

今日知识要点

- 函数的基本概念
- python中常用函数的使用
- 编写简单的函数解决简单数学
- 使用sympy解决高等数学问题

函数的初步认识

函数的定义

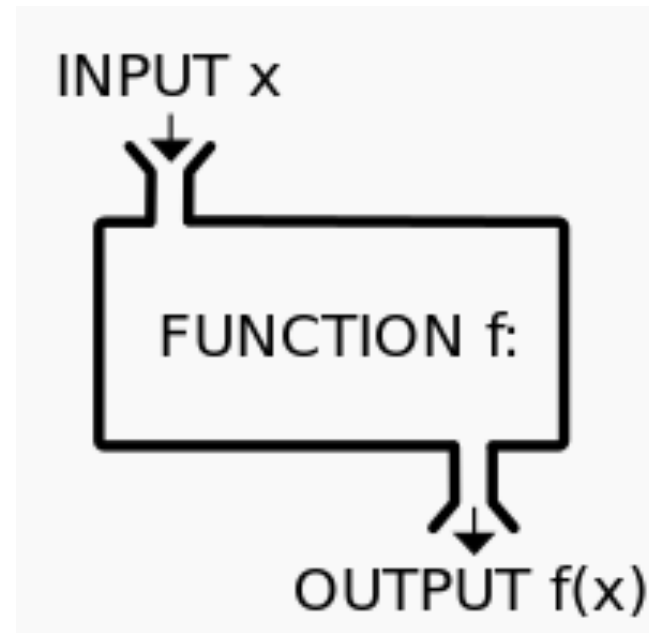
数学函数：

$$y = f(x)$$

↑ ↑ ↙

输出值 函数代号 输入值

 (函数名)



计算机函数：代码片段

- 一段具有特定功能的、可重复使用的代码。
- 用函数名来表示并通过函数名完成功能调用。

函数的定义

使用函数的意义：

- 代码的重复利用；
- 减少程序中的代码重复量，使代码更加容易被理解；
- 让代码更易于维护与更新。

函数的使用

函数

内置函数

- Python编程语言中已经被定好功能的函数。
- 可以**直接调用**来执行特定的任务。

用户自定义函数

函数的功能和使用方法都由用户决定。

使用一个函数

Python本身内置了很多函数

我们可以直接使用这些函数帮助

我们完成一些事情

输出函数: `print()`

获取用户输入的函数: `input()`

类型转换的函数: `int()` `float()`

`str()`.....

.....

```
a = abs(-1)
print(a)

b = int('12')
c = 13
print(b+c)
```

使用一个函数

除了内置的函数以外，Python还有内置 的标准函数库

当我们把这些额外的函数库导入到我们的程序里时，我们可以用这些函数库中提供的方法

访问这些函数时需要以**库名.函数名**的方式来应用

```
import math  
  
a = math.sqrt(2)  
print(a)
```

```
import random  
  
a = random.randint(0,10)  
print(a)
```

常用数学计算库: math

- 更多math模块的用法可以参考api文档:
<https://www.apiref.com/python-zh/library/math.html>
- 或者可以使用dir()和help()函数查看用法

```
import math as m
result = m.cos(m.radians(60))
print(result)
result = m.acos(0.5)
print(m.degrees(result))
```

```
0.5000000000000001
60.00000000000001
[Finished in 0.1s]
```

math.sin(x)	返回x（弧度）的三角正弦值
math.asin(x)	返回x的反三角正弦值
math.cos(x)	返回x（弧度）的三角余弦值
math.acos(x)	返回x的反三角余弦值
math.tan(x)	返回x（弧度）的三角正切值
math.atan(x)	返回x的反三角正切值

常用数学函数库：random

```
import random  
print(random.random())  
print(random.randint(0,100))
```

```
0.04168987061190976  
53  
[Finished in 0.1s]
```


函数的定义

定义一个函数

`def`是define的缩写，表示定义一个函数

格式：

`def` 函数名称():

函数内的语句

当我们需要用这个函数时，直接写"`函数名称+()`"就即可

注意：函数名称中间不要有空格，可以用下划线代替

```
def HelloUser():  
    print('Hello')  
  
HelloUser()
```

形参和实参 (Parameter&Argument)

- 形参 (Parameter) :

函数执行需要的信息数据, 用在函数的定义中

- 实参 (Argument) :

在函数调用时提供的具体信息

位置参数vs关键字参数

■ 位置参数 (Positional Argument) :

实参的顺序和形参顺序相同, 参数放置的顺序很重要

■ 关键字参数 (Keyword Argument) :

实参由变量名和值通过等号连接组成

默认值参数

■ 默认值参数 (Default Value Augument) :

定义函数时给形参的默认赋值，如果调用的时候没有实参赋值则给形参赋默认值，如果有实参，默认值会被忽略。

利用参数向函数传递信息

```
def HelloUser(username):  
    print('Hello!' + username)
```

} 定义函数

```
HelloUser('Lilei')
```

调用函数

```
Hello!Lilei
```

传递参数

- 位置参数

```
#显示某次考试成绩  
def TestGrade(name, grade):  
    print(name + '的成绩是' + grade + '。')  
  
TestGrade('李雷', '99')
```

李雷的成绩是99。

传递参数

- 关键字参数

```
#显示某次考试成绩  
def TestGrade(name, grade):  
    print(name + '的成绩是' + grade + '。')  
  
TestGrade(name = '李雷', grade = '99')
```

李雷的成绩是99。

传递参数

■ 默认值参数

```
1  def TestGrade(name, grade='100'):  
2      print(name + '的成绩是' + grade + '。')  
3  
4  TestGrade(name='李雷')
```

```
李雷的成绩是100。  
[Finished in 0.1s]
```

返回值

def 函数名称(参数1, 参数2,):

函数内的语句

.....

return 返回内容

定义返回值的关键字

```
def add(a, b):  
    sum_number = a + b  
    return sum_number  
  
value = add(4, 12)  
print(value)
```

返回值

def 函数名称(参数1, 参数2,):

函数内的语句

.....

return 返回内容

红色部分是必须有的

紫色部分是可选的

```
def say_hi():  
    print('Hi')  
    print("I'm Dr.Wu.")
```

```
Hi  
I'm Dr.Wu.  
[Finished in 0.1s]
```

变量的作用域

```
a = 1

def hello():
    a = 2
    print(a)

hello()
print(a)
```

```
a = 1

def hello():
    print(a)

hello()
print(a)
```

```
a = 1

def hello():
    print(a)
    a = 2

hello()
print(a)
```

变量的作用域

```
a = 1
```

```
def hello():  
    global a  
    print(a)  
    a = 2
```

```
print(a)  
hello()  
print(a)
```

引入 **global** 关键词来声明 `a` 是全局变量

创建并使用函数的好处

1. 减少程序中重复的代码
2. 使主程序更抽象，结构更清晰

```
initial()  
draw_head()  
draw_eyes()  
draw_body()
```

重做思考题

- 根据王者荣耀伤害结算公式，制作一个普攻伤害计算函数，并且计算李白在物攻834、物理穿透300|45%的状态下普攻物抗为1588的项羽时造成的伤害。并且打印伤害提示“李白对项羽造成了XX点伤害”

- 参考文献：

<https://zhuanlan.zhihu.com/p/68485973>



用函数提高代码的可读性

```
import random
def get_damage(attack,defence,penetrate,percentage):
    r_defence = (defence-penetrate)*(1-percentage)
    remission = r_defence/(602+r_defence)
    damage = int(attack*(1-remission))+random.randint(-3,3)
    return damage

attack = 834
defence = 1588
penetrate = 300
percentage= 0.45
damage = get_damage(attack,defence,penetrate,percentage)
print(f"李白对项羽造成了{damage}点伤害")
damage = get_damage(attack,defence,penetrate,percentage)
print(f"李白对项羽造成了{damage}点伤害")
damage = get_damage(attack,defence,penetrate,percentage)
print(f"李白对项羽造成了{damage}点伤害")
```

```
李白对项羽造成了383点伤害
李白对项羽造成了386点伤害
李白对项羽造成了380点伤害
[Finished in 0.1s]
```


思考题

- 编写python程序用求解方程：

$$x^2 - 3x - 4 = 0$$

编写函数实现数学计算公式

一元二次方程求根公式

$$x = \frac{-b \pm \sqrt{\Delta}}{2a} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
1 import math as m
2 def get_qewou_solution(a,b,c):
3     delta = b**2-4*a*c
4     solution1 = (-b+m.sqrt(delta))/(2*a)
5     solution2 = (-b-m.sqrt(delta))/(2*a)
6     return solution1,solution2
7
8 a = 1
9 b = -3
10 c = -4
11 x1 = get_qewou_solution(a,b,c)[0]
12 x2 = get_qewou_solution(a,b,c)[1]
13 print(f'x1 = {x1}')
14 print(f'x1 = {x2}')
```

```
x1 = 4.0
x1 = -1.0
[Finished in 0.1s]
```

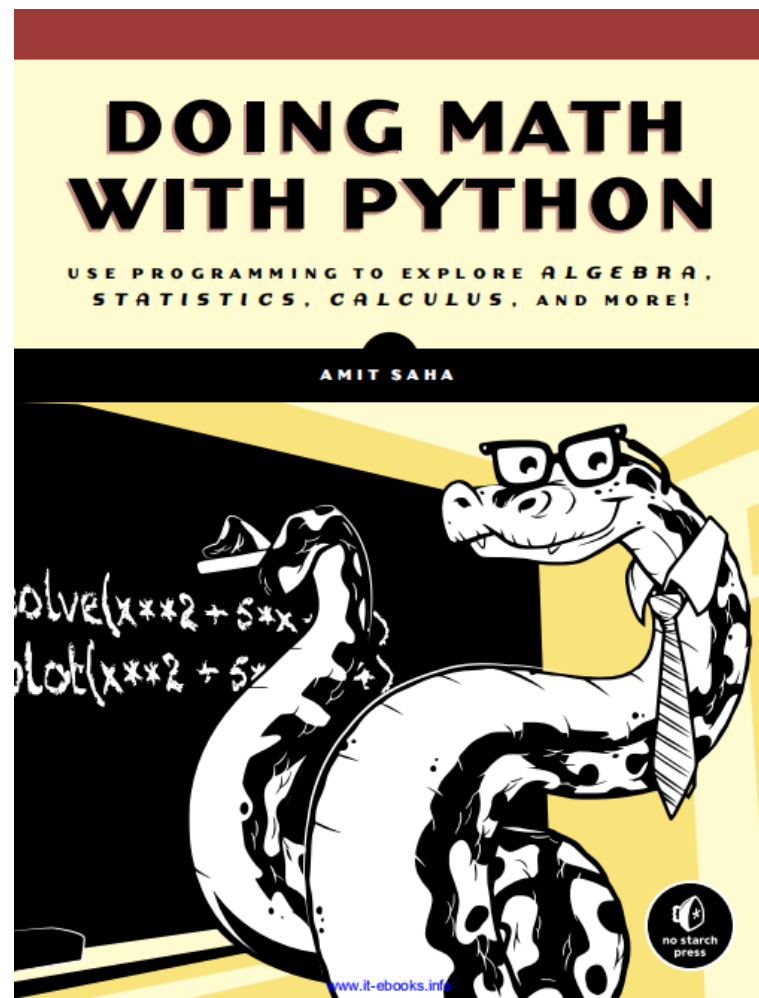
Bonus: 使用sympy解决数学问题

sympy的安装

- `pip install sympy`

```
命令提示符 - python
C:\Users\wuchenwei>pip install sympy
Requirement already satisfied: sympy in s:\users\wuchenwei\
appdata\local\programs\python\python36\lib\site-packages
Requirement already satisfied: mpmath>=0.19 in s:\users\wuc
henwei\appdata\local\programs\python\python36\lib\site-pack
ages (from sympy)
You are using pip version 9.0.3, however version 20.0.2 is
available.
You should consider upgrading via the 'python -m pip instal
l --upgrade pip' command.

C:\Users\wuchenwei>python
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MS
C v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more i
nformation.
>>> import sympy
>>> _
```



用sympy解方程

1. 导入模块和类
2. 使用Symbol类列方程
3. 使用solve方法解方程

例：求解方程： $x^2 - 3x - 5 = 0$

```
from sympy import Symbol, solve, pprint
expr = x**2 - 3*x - 5
solution = solve(expr, x)
pprint(solution)
```

$\left[-\frac{3}{2} - \frac{\sqrt{29}}{2}, -\frac{3}{2} + \frac{\sqrt{29}}{2} \right]$
[Finished in 1.5s]

例：求解方程： $ax^2 + bx + c = 0$

$\left[\frac{-b + \sqrt{b^2 - 4ac}}{2a}, \frac{-b - \sqrt{b^2 - 4ac}}{2a} \right]$
[Finished in 1.6s]

```
from sympy import Symbol, solve, pprint
x = Symbol('x')
a = Symbol('a')
b = Symbol('b')
c = Symbol('c')
expr = a*x**2 + b*x + c
solution = solve(expr, x)
pprint(solution)
```

用sympy解方程组

例：求解方程组：

$$2x + 3y = 6$$

$$3x + 2y = 12$$

```
from sympy import Symbol, solve, pprint
x = Symbol('x')
y = Symbol('y')
expr1 = 2*x + 3*y - 6
expr2 = 3*x + 2*y - 12
solution = solve((expr1, expr2), dict=True)
pprint(solution)
```



```
[{x: 24/5, y: -6/5}]
[Finished in 1.2s]
```

用sympy求微分（导数）

1. 导入模块和类
2. 列出表达式
3. 使用Derivative(f,x,order=1)建立对象
4. 使用doit()方法求解
5. 使用subs方法赋值

例：求下列函数的导数表达式
及其在 $t = 3$ 时的值：

$$S = 5t^2 + 3t + 8$$

```
from sympy import Symbol, Derivative
t = Symbol('t')
S = 5*t**2 + 3*t + 8
dSdt = Derivative(S, t).doit()
print(dSdt)
print(dSdt.subs({t: 3}))
```



```
10*t + 3
33
[Finished in 1.1s]
```

用sympy求积分

1. 导入模块和类
2. 列出表达式
3. 使用Integral(f,x,order=1)建立对象
4. 如果是定积分，第二个参数为元组(x,min,max)
5. 使用doit方法求解答案

```
from sympy import Symbol, Integral, pprint
x = Symbol('x')
k = Symbol('k')
f = k*x
solution = Integral(f,x).doit()
pprint(solution)
result = Integral(f,x,(x,0,2)).doit()
pprint(result)
```

例：求以下不定积分
和定积分：

$$\int kx dx$$

$$\int_0^2 kx dx.$$

```
      2
    k·x
    —
      2
    4·k
    —
      3
[Finished in 1.4s]
```


作业1

- 根据温度转换公式，编写温度转换函数，输入华氏温度，输出摄氏温度。

Fahrenheit To Celsius

$$C = \frac{5}{9}(F - 32)$$

作业2

- 使用sympy方法求解一下两道考研数学真题

(9) 已知函数 $f(x) = \frac{1}{1+x^2}$, 则 $f^{(3)}(0) = \underline{\hspace{2cm}}$ 。

16. (本题满分 10 分)

求不定积分 $\int \frac{3x+6}{(x-1)^2(x^2+x+1)} dx$.

作业3

- 分别通过自己编写函数和调用sympy函数来求解鸡兔同笼问题

今有雉兔同笼，上有三十五头，
下有九十四足，问雉兔各几何？

Tips: 输入n与m，表示有n个头、m只脚，算出笼子中各有多少只鸡和兔？