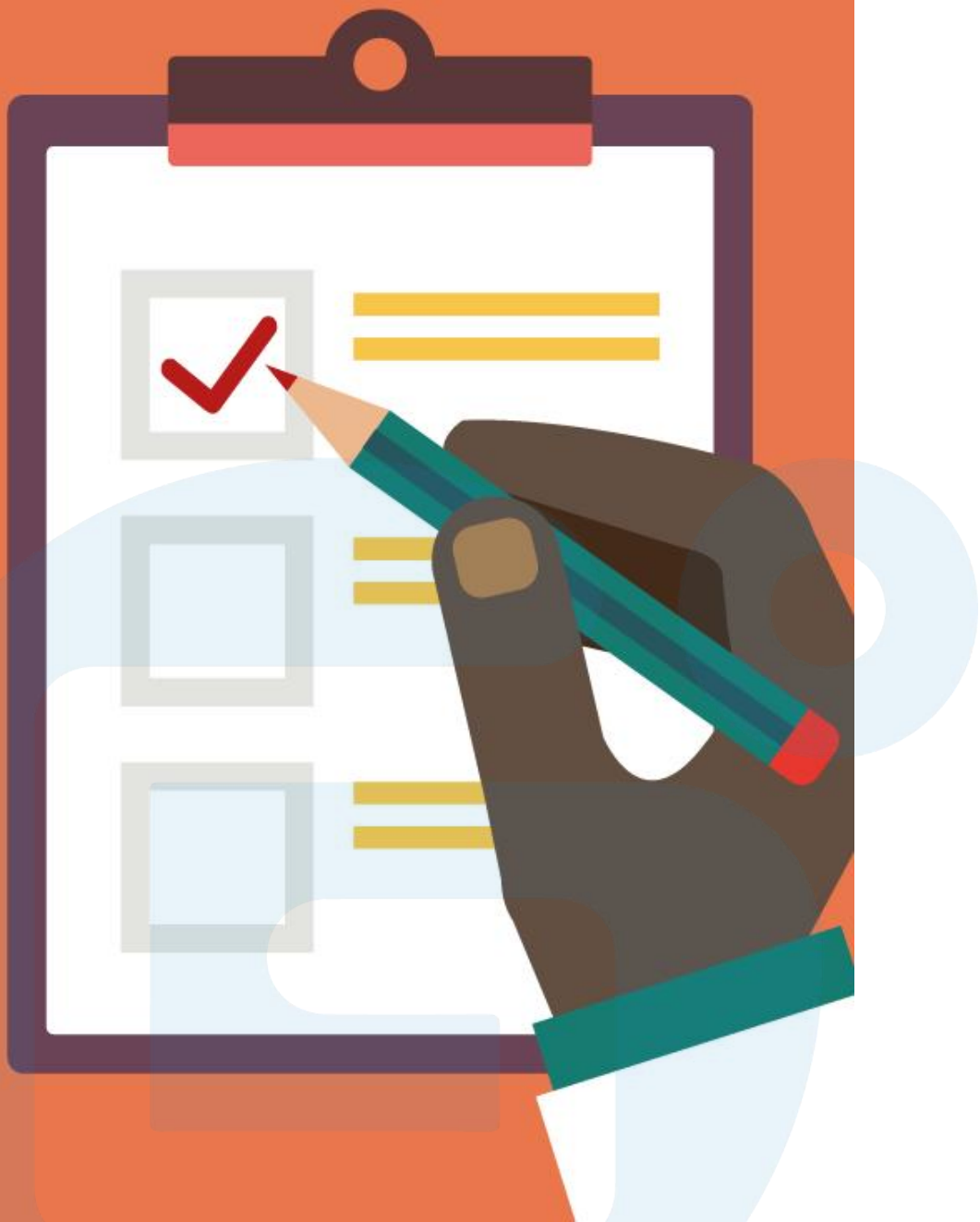


Talk is cheap, show me the code

# 第三课：语法与数据结构基础

Python初阶入门课程系列



---

# OUTLINE

➤ Python语法基础

➤ Python数据结构基础

➤ 复数

➤ 符号运算

---



# 一 Python语法基础



## Python语言的风格

可读性，简洁，清晰。称之为可运行的伪代码.....

### 著名的特点，利用“空格”组织代码

Python使用空白字符（tab和空格）来组织代码，而不是像其它语言，如R、C++、JAVA和Perl那样使用括号。

```
for x in array:
    if x < pivot:
        less.append(x)
    else:
        greater.append(x)
```

```
for x in array:
    if x < pivot:
        less.append(x)
    else:
        greater.append(x)
```

冒号标志着缩进代码块的开始，冒号之后的所有代码的缩进量必须相同，直到代码块结束。不管是否喜欢这种形式，使用空白符是Python程序员开发的一部分，可以让python的代码可读性大大优于其它语言。

在spyder里，使用tab按键可以直接缩进4个空格。通常遇到需要缩进的地方，编辑器也会自动帮助缩进。

采用其它IDE的同学，也建议使用4个空格作为默认的缩进方法。

Python语言无需分号作为语句结束的标志。

不过如果你想把几个语句写在同一行里，还是需要用分号分割一下。

总的来说，不太建议把几个语句写在同一行里，会降低代码的可读性。

除非语句真的非常简短。并且几个语句之间存在并列关系。

```
7  
8  
9  
10  
11  
12
```

```
#%%
```

```
a=1
```

```
b=2
```

```
c=3
```

```
#%%
```

```
a=3;b=1;c=2|
```

## Python代码的注释

注释有助于他人和自己理解代码的具体意义。

分为单行注释和多行注释

单行注释用“#”符号开头。写在语句前面后面都可以。

多行注释用三个“”或者”圈起来，一般用于注释过长的情况

**注意注释不是越多越好。**

- 对于一目了然的代码，不需要添加注释。
- 对于复杂的操作，应该在操作开始前写上相应的注释。
- 对于不是一目了然的代码，应该在代码之后添加注释。
- 绝对不要描述代码。一般阅读代码的人都了解Python的语法，只是不知道代码要干什么。

```
7  ###
8  a=1
9  b=2
10 c=3
11 ###
12 #这是注释
13 a=3;b=1;c=2 #这是注释
14 #这是注释
15
16 ...
17 这些是
18 多行
19 注释
20 ...
21
22 """三个双引号
23 也可以
24 作为多行注释"""
```

## 函数 (FunctionType)

函数是封装了一些独立的功能，可以直接调用，能将一些数据（参数）传递进去进行处理，然后返回一些数据（返回值），也可以没有返回值。可以直接在模块中进行定义使用。所有传递给函数的数据都是显式传递的。

## 方法 (MethodType)

方法和函数类似，同样封装了独立的功能，但是方法是只能依靠类或者对象来调用的，表示针对性的操作。

方法中的数据self和cls是隐式传递的，即方法的调用者；方法可以操作类内部的数据  
简单的说，函数在python中独立存在，可直接使用的，而方法是必须被别人调用才能实现的。

- 与类和实例无绑定关系的function都属于函数（function）；
  - 与类和实例有绑定关系的function都属于方法（method）。
-



## Python里常用的两个标点符号

### 点和圆括号.()

- 用圆括号调用函数，传递零个或几个参数，或者将返回值给一个变量：

```
result = f(x, y, z)  
g()
```

- 用点引用对象里附加的方法

```
obj.some_method(x, y, z)
```

---

## Python里变量的创建、赋值、参数传递

变量，类似于xyz之类的名字，是代码编写过程中的基本元素。

Python作为高级语言，变量无需申明即可直接调用，也无需强制手动规定变量类型。

赋值使用“=”符号实现，注意这里的=并不表示等于的意思，而是将右边的值传递给左边。

```
7  ###
8  a=1
9  b=2
10 c=3
11 x=a
12 y=b
13 z=c
14 print("x=",x,"y=",y,"z=",z)|
```

控制台 1/A

```
In [9]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wdir='C:/Users/Administrator/Desktop/test')
x= 1 y= 2 z= 3
```

## Python里变量的创建、赋值、参数传递

变量可以赋数值，也可以是布尔变量，字符串，字典等各种数据结构。如果是数值，可以是一个标量值，也可以是一个向量值，或者是矩阵，复数，实数等。

```
8 a=1
9 b=2
10 c=3
11 x=[1,2,3]
12 y="this is word"
13 z=True
14 |
15 print("x=",x,"y=",y,"z=",z)
```

控制台 1/A

```
In [12]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wdir='C:/Users/Administrator/Desktop/test')
x= [1, 2, 3] y= this is word z= True
```

## Python里库的引入

库的本质就是其它.py文件。如果在某个新的.py文件里使用其它.py文件，就需要提前引用。

```
import some_module
```

```
result=some_module.f(5)
```

```
pi=some_module.PI
```

利用.符号使用已有文件里的函数（方法）和变量

另外一种引用方式

```
from some_module import f, g, PI
```

```
result=g(5,PI)
```

## Python里库的引入

还有一种引用方式，使用as，可以给引入的文件起个新名字

```
import some_module as sm
```

```
from some_module import PI as pi, g as gf
```

```
r1=sm.f(pi)
```

```
r2=gf(6,pi)
```

## Python的二元运算符和比较符

运算	说明
$a + b$	a 加 b
$a - b$	a 减 b
$a * b$	a 乘以 b
$a / b$	a 除以 b
$a // b$	a 除以 b, 结果只取整数部分
$a ** b$	a 的 b 次幂
$a \& b$	a 或 b 都为 True, 则为 True; 对于整数, 取逐位 AND
$a   b$	a 或 b 有一个为 True, 则为 True; 对于整数, 取逐位 OR
$a \wedge b$	对于布尔, a 或 b 有一个为 True, 则为 True, 二者都为 True, 为 False; 对于整数, 取逐位 EXCLUSIVE-OR
$a == b$	a 等于 b, 则为 True
$a != b$	a 不等于 b, 则为 True
$a < b, a \leq b$	a 小于 (或小于等于) b, 则为 True
$a > b, a \geq b$	a 大于 (或大于等于) b, 则为 True
$a \text{ is } b$	a 和 b 引用同一个 Python 对象, 则为 True
$a \text{ is not } b$	a 和 b 引用不同的 Python 对象, 则为 True

```
7  #%%
8  a=1
9  b=2
10 c=3
11 x=a*b
12 y=c**b
13 z=a-b
14 print("x=",x,"y=",y,"z=",z)
```

控制台 1/A

```
In [15]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wdir=
x= 2 y= 9 z= -1
```

```
In [16]:
```



---

## 二 Python数据结构基础



## Python的标量

Python的标准库中有一些内建的类型，用以处理数值数据、字符串、布尔值，和日期时间。这些单值类型被称为标量类型。

类型	说明
None	Python 的空值（只存在一个 None 对象的实例）
str	字符串类型，存有 Unicode（UTF-8 编码）字符串
bytes	原生 ASCII 字节（或 Unicode 编码为字节）
float	双精度（64 位）浮点数（注意没有 double 类型）
bool	True 或 False 值
int	任意精度整数

日期和时间处理会另外讨论，因为它们是标准库的datetime模块提供的



# Python 数据结构基础

## Python的标量

主要数值类型是int和float。 int可以存储任意大的整数，没有溢出问题：

```
8 import sys
9 MAX_INT=sys.maxsize
10 print(MAX_INT)
11 |
```

控制台 1/A

```
In [19]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wdir='C:/Users/Administrator/Desktop/test')
9223372036854775807
```

```
8 a=9999999999
9 b=6
10 x=a**b
11
12 print("x=",x)
```

控制台 1/A

```
In [15]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wdir='C:/Users/Administrator/Desktop/test')
x= 2 y= 9 z= -1

In [16]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wdir='C:/Users/Administrator/Desktop/test')
x= 9999999994000000001499999999800000000149999999994000000001
```

## Python的标量（数值）

float可以存储浮点数，全部是**双精度(64位)**存储：

```
7  """
8  a=0.000000000005
9  b=1.23e-10
10 c=52.55
11 print(a,b,c)
```

控制台 1/A

```
In [21]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wdir='C:/Users/Administrator/Desktop/test')
5e-12 1.23e-10 52.55
```

int型的除法如果除不尽，则自动产生float型结果：如果想产生int型结果，可以用//

```
8  a=12
9  b=4
10 c=5
11 print(a/b)
12 print(a//c)
13 print(a/c)
```

控制台 1/A

```
In [24]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wdir='C:/Users/Administrator/Desktop/test')
3.0
2
2.4
```

## Python的字符串

可以用单引号或双引号来写字符串，  
换行字符串用三个引号：

```
7  ###
8  a='this is string'
9  b="also string"
10 c='''
11 this
12 is
13 string
14 too
15 '''
16 print(a)
17 print(b)
18 print(c)
```

```
In [26]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wc
this is string
also string
```

```
this
is
string
too
```

字符串可以直接赋值获得，也可以用`str`函数获得。变成字符串之后的数值尽管看起来还是数字，但已经不能用于运算了。

```
7  ###
8  a=38
9  b=str(a)
10 print(a)
11 print(b)
12
13 print(a*a)
14 print(b*b)
```

控制台 1/A

```
In [29]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wdir='C:/Users/Adm
38
38
1444
```

Traceback (most recent call last):

```
File "C:\Users\Administrator\Desktop\test\aaaa.py", line 14, in <module>
    print(b*b)
```

`TypeError: can't multiply sequence by non-int of type 'str'`

## Python的字符串

字符串里的反斜杠符号有特殊含义 \  
通常用于表示特殊字符，例如换行符  
\n 空格符

转义符	描述
(在行尾时)	续行符
\	反斜杠符号
\n	换行
\v	纵向制表符
\t	纵向制表符
\r	回车
\f	换页

如果字符串里确实含有反斜杠符，则需要  
两个反斜杠符才能正确显示。

```
7  #%%
8  a='this is \n'
9  b='this is \\n'
10
11 print(a)
12 print(b)
```

控制台 1/A

```
In [34]: runfile('C:/Users/Administrator/Desktop/1
this is
this is \n
```

## Python的字符串

如果字符串里含有很多反斜杠符，但实际上都不是转义符。一个一个键入反斜杠符很麻烦，可以在字符串前面加r表示这里的反斜杠符是字符串本身。

```
7  #%%
8  a='this is \n ,\it is \not \special\characters'
9  b='this is \n ,\it is \not \special\characters'
10
11  print(a)
12  print('#\n#\n#')
13  print(b)
```

控制台 1/A

```
In [44]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wdir='C:/Users/Admi
this is \n ,\it is \not \special\characters
#
#
#
this is
 ,\it is
ot \special\characters
```

字符串可以相加

```
6  #%%
7
8  a='hello'
9  b='world'
10
11  print(a)
12  print('#\n#\n#')
13  print(b)
14  print('#\n#\n#')
15  print(a+b)
```

控制台 1/A

```
In [3]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wdir='C:/Users
hello
#
#
#
world
#
#
#
helloworld
```

## Python的布尔变量

布尔变量（Boolean Variable）就是逻辑变量， True False ，在程序里用0和1表示，要注意尽管是数字形式，但是它不是数值。

如同数值变量可以进行加减乘除运算一样，布尔变量可以进行逻辑运算。

```
6
7
8 a=True
9 b=False
10 x=bool(0)
11 y=bool(1)
12
13 print(a,b,x,y)
14 print(a or b)
15 print(x and y)
16 print(not a)
17
```

控制台 1/A

```
In [12]: runfile('C:/Users/Administrator/Desktop/test/
True False False True
True
False
False
```

尽管不用0或1，或者不用布尔变量也可以执行布尔运算，但是容易造成代码逻辑混乱，因此不推荐滥用数值变量进行布尔运算。

```
8 a=100
9 b=200
10 x=bool(10)
11 y=bool(-20)
12
13 print(a,b,x,y)
14 print(a or b)
15 print(x and y)
16 print(not a)
17
```

控制台 1/A

```
In [16]: runfile('C:/Users/Administ
100 200 True True
100
True
False
```

## Python的变量类型转换

str, int, float, bool这些常用类型可以互相转换，另外有一个特殊的类型，None

```
7  #%%
8  a=3.1415926
9
10 b=str(a)
11 x=int(a)
12 y=float(a)
13 z=bool(a)
14 |
15 print(b,x,y,z)
16
17
```

控制台 1/A

```
In [19]: runfile('C:/Users/Administrat
3.1415926 3 3.1415926 True
```

```
7  #%%
8  a=None #表示空值
9  b=5
10
11 c=a is None
12 d=b is not None
13 print(c,d)
14
15
```

控制台 1/A

```
In [21]: runfile('C:/Users/Administrator/Desktop/test/aa
True True
```



## Python的日期变量

内置了一个datetime的包，用来表示时间

```
8 from datetime import datetime, date, time
9 dt=datetime(2011,10,29,20,30,21)
10
11 print(dt)
12 print(dt.day)
13 print(dt.minute)
```

控制台 1/A

```
In [33]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wdir='C:/Users/Administra
2011-10-29 20:30:21
29
30
```





# 三 复数



## 复数的概念

复数是由一个实数和一个虚数组合构成，表示为： $x+yj$

一个复数时一对有序浮点数  $(x,y)$ ，其中  $x$  是实数部分， $y$  是虚数部分。

### Python 语言中有关复数的概念：

- 1、虚数不能单独存在，它们总是和一个值为 0.0 的实数部分一起构成一个复数
- 2、复数由实数部分和虚数部分构成
- 3、表示虚数的语法：`real+imagej`
- 4、实数部分和虚数部分都是浮点数
- 5、虚数部分必须有后缀j或J

# 复数

注意用j而不是习惯的i表示复数的虚部，另外直接用数值加j表示虚部

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Thu Apr  8 20:05:47 2021
4
5 @author: Administrator
6 """
7 va=155.05+90j
8 print (va.real) # output 实数部分
9 print (va.imag) # output 虚数部分
```

控制台 1/A

```
In [4]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wdir='C:/Users/Administrator/Desktop/t
155.05
90.0
```

# 复数

复数对象有三个内建功能。**real**实部，**imag**虚部，**conjugate**共轭。  
另外可以用**complex**函数创建复数。

```
6  
7 va=155.05+90j  
8 print (va.real) # output 实数部分  
9 print (va.imag) # output 虚数部分  
10 print (va.conjugate())  
11  
12 a=10.5  
13 b=20  
14 c=a-b*1j  
15 print(c)  
16 d=complex(a,b)  
17 print(d)
```

控制台 1/A

```
In [11]: runfile('C:/Users/Administrator/Desktop/test  
155.05  
90.0  
(155.05-90j)  
(10.5-20j)  
(10.5+20j)
```

# 复数

对应math库，有处理复数的数学库，cmath

```
5 @author: Administrator
6 """
7 #%%
8 import math
9 a=math.pi/6
10 print(math.sin(a))
11 #%%
12 from math import pi,sin
13 b=pi/2
14 print(sin(b))
```

控制台 1/A

```
In [12]: runfile('C:/Users/Administrat
0.49999999999999994
1.0
```

```
7 #%%
8 import cmath
9 a=cmath.pi/6*1j
10 print(cmath.sin(a))
11 #%%
12 from cmath import pi,sin
13 b=pi/2*1j
14 print(sin(b))
```

控制台 1/A

```
In [16]: runfile('C:/Users/Administrat
0.54785347388880397j
2.3012989023072947j
```

复变正弦函数与余弦函数定义是通过欧拉公式转换成的

$$\sin z = \frac{1}{2i} (e^{iz} - e^{-iz}), \cos z = \frac{1}{2} (e^{iz} + e^{-iz})$$

```
7  #%%  
8  import cmath  
9  a=cmath.pi/6*1j  
10 print(cmath.sin(a))  
11 #%%  
12 from cmath import pi,sin  
13 b=pi/2*1j  
14 print(sin(b))
```

控制台 1/A

```
In [16]: runfile('C:/Users/Administrat  
0.5478534738880397j  
2.3012989023072947j
```

## 欧拉公式的验证

$$e^{ix} = \cos x + i \sin x$$

```
7  ###
8  from cmath import exp,sin,cos
9  a=5*1j
10 b=exp(5*1j)-cos(5)-1j*sin(5)
11 c=sin(5)
12 d=cos(5)
13 e=exp(5*1j)
14 print(b,'\n',c,'\n',d,'\n'|e)
```

控制台 1/A

```
In [22]: runfile('C:/Users/Administrator/Desktop/test/aaa
0j
(-0.9589242746631385+0j)
(0.28366218546322625+0j)
(0.28366218546322625-0.9589242746631385j)
```



---

# 四 基础符号运算



---





# 基础符号运算

符号运算与数值计算对应。通俗地说就是用计算机推导数学公式，如对表达式进行因式分解、化简、微分、积分、解代数方程、求解常微分方程等。

python利用SymPy库实现这一功能

数值解是带浮点数的近似值，而符号解(解析解)则是精确值。

```
5  @author: Administrator
6  """
7  ###
8  import math
9  math.pi
10 print(math.sin(math.pi))
11 ###
12 from sympy import sin, pi
13 print(sin(pi))
```

控制台 1/A

```
In [26]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wdir='C
1.2246467991473532e-16
0
```

# 基础符号运算

对竖直上抛物体的位移方程进行微分，然后再积分

$$h(s) = v_0 t - \frac{1}{2} g t^2$$

```
7  #%%
8  from sympy import *
9  t,v0,g=symbols('t v0 g')
10 y=v0*t-Rational(1,2)*g*t**2 #rational numbers
11 dydt=diff(y,t)
12 print(dydt)
13 dydt2=diff(y,t,t)
14 print(dydt2)
15 y2=integrate(dydt,t)
16 print(y2)
```

控制台 1/A

```
In [31]: runfile('C:/Users/Administrator/Desktop/test/aaaa.
-g*t + v0
-g
-g*t**2/2 + t*v0
```

```
7  #%%
8  from sympy import symbols,Rational,diff,integrate
9  t,v0,g=symbols('t v0 g')
10 y=v0*t-Rational(1,2)*g*t**2 #rational numbers
11 dydt=diff(y,t)
12 print(dydt)
13 dydt2=diff(y,t,t)
14 print(dydt2)
15 y2=integrate(dydt,t)
16 print(y2)
```

控制台 1/A

```
In [33]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wdir='C:
-g*t + v0
-g
-g*t**2/2 + t*v0
```

# 基础符号运算

## 新建符号

在使用符号之前，先要利用**symbols**函数定义符号

```
7  ###
8  from sympy import symbols
9  # 新建符号 x, y
10 x, y = symbols('x y')
11 # 利用 SymPy 的 abc 子模块新建符号 x, y
12 ###
13 from sympy.abc import x, y
14 # 利用 SymPy 的 abc 子模块导入所有拉丁、希腊字母:
15 ###
16 from sympy import symbols
17 x = symbols('x', positive = True)
18 #新建符号变量时可以指定其定义域
19 vars = symbols('x_1:5')
20 print(vars)
```

```
###
from sympy import *
x, y, z = symbols('x y z')
y = expand((x + 1)**2) # expand() 是展开函数
print(y)
```

$$x^2 + 2x + 1$$

# 基础符号运算

符号运算的基本操作  
替换subs

sympify字符串转换符号表达式

```
7  ###
8  from sympy import *
9  x = symbols('x')
10 expr = cos(x) + 1
11 print(expr.subs(x, 0))
12
```

控制台 1/A

```
In [52]: runfile('C:/Users/Administrator/Desktop/test/aaaa.py', wdir=
2
```

```
6  ###
7  from sympy import *
8  str_expr = 'x**2 + 2*x + 1'
9  expr = sympify(str_expr)
10 print(expr)
11
12
```

控制台 1/A

```
In [54]: runfile('C:/Users/Administrator/Des
x**2 + 2*x + 1
```

# 基础符号运算

## 符号运算的基本操作 化简simplify

```
7  #%%
8  from sympy import *
9  from sympy.abc import x,y
10 print(simplify(sin(x)**2 + cos(x)**2))
11 print(simplify(2*sin(y)*cos(y)))
12
```

控制台 1/A

```
In [58]: runfile('C:/Users/Administrator/Desktop/tes
1
sin(2*y)
```



# 下节预告



- 字符串型变量
  - **Time**型变量
  - 若干综合性实例
  - ...
-



感谢参与 下堂课见

