

序号	符号	名称	作用	说明和举例	详细介绍
1	,	逗号	1.分隔变量、列表、元组、集合、字典类型的各个元素	列表[3,4,5]、元组(3,4,5)、集合{3,4,5}、字典	第5章: Python数据类型
			2.分隔函数参数	def func(a,b):	第7章: Python函数
			3.分隔多继承的类	class C(A,B):	第8章: Python类和对象
2	.	点号	实例化对象调用类的属性或方法	class Dog(): dog1=Dog() dog1.name dog1.intro()	第8章: Python类和对象
3	=	赋值号	给变量赋值	a=5表示将5赋值给变量a a=b表示将变量b的值赋给变量a a,b,c=7,8,9表示将7、8、9分别赋值给变量a、变量b、变量c	第5章: Python数据类型
4	==	等号	等于	a==5表示变量a的值等于5 a==b表示变量a的值等于b	第6章: Python控制结构
5	!=	不等号	不等于	a!=b表示变量a的值不等于b	第6章: Python控制结构
6	>	大于号	大于		第6章: Python控制结构
7	>=	大于等于号	大于等于		第6章: Python控制结构
8	<	小于号	小于		第6章: Python控制结构
9	<=	小于等于号	小于等于		第6章: Python控制结构
10	+=	自增赋值号	符号左边的变量加上符号右边的数值或者变量,取得的结果再赋值给符号左边的变量	a=9; a+=1; print(a) 10	
11	-=	自减赋值号	符号左边的变量减去符号右边的数值或者变量,取得的结果再赋值给符号左边的变量	b=9; c=6; b-=c; print(b) 3	
12	:	冒号	1.提示条件语句体的开始和分支	if dog_age == 1: elif dog_age > 1:	第6章: Python控制结构
			2.提示while循环语句体的开始	while counter <= n:	第6章: Python控制结构
			3.提示for循环语句体的开始	for a in range (10):	第6章: Python控制结构
			4.提示函数体的开始	def func(a,b):	第7章: Python函数
			5.提示类的开始	class Dog():	第8章: Python类和对象
			6.分隔字典类型数据的key和value	{'Name': 'Jim', 'Age': 20, 'Gender':'Male'}	第8章: Python类和对象
13	;	分号	一般不使用。如果在一行中书写多条语句,就必须使用分号分隔每个语句	x=1; y=1 ; z=1	

序号	符号	名称	作用	说明和举例	详细介绍
14	\	转义符	配合特定字符使用，表达特殊含义	\n表示换行 \t表示Tab键 \'表示单引号 \"表示双引号 \\表示一个\ print('\n\'m \"OK!\") I'm "OK!"	第5章：Python数据类型
15	#	注释符	以#开头的是语句是注释语句，不参与程序运行	#本行的作用是增加一个变量	第4章：Python基础语法
16	'	单引号	1.字符串的开始和结尾需要用单引号包起来	a='1'; b='2'; a+b '12'	第5章：Python数据类型
			2.表示多行注释，首尾都必须用三个单引号包起来	"""本程序开发人员A 开发时间YYYY-MM-DD 版本号1.0.0"""	第4章：Python基础语法
17	"	双引号	1.字符串的开始和结尾需要用双引号包起来	a="1"; b="2"; a+b '12'	第5章：Python数据类型
			2.表示多行注释，首尾都必须用三个双引号包起来	"""本程序开发人员A 开发时间YYYY-MM-DD 版本号1.0.0"""	第4章：Python基础语法
18	+	加号	1.两个数字相加	a=1; b=2.5; a+b 3.5	第5章：Python数据类型
			2.两个字符串相连接	s1='hello'; s2='world'; s1+s2 'helloworld'	第5章：Python数据类型
19	-	减号	两个数字相减，或者表示负数	c = -5; d = -8; c-d 3	第5章：Python数据类型
20	*	乘号	1.两个数字相乘	e=7.5;f=8;e*f 60.0	第5章：Python数据类型
			2.相同字符串多次重复	s='hello'; t=3; s*t 'ellohellohello'	第5章：Python数据类型
21	**	乘方号	数字的n次乘方	a=2; b=5; a**b 32	第5章：Python数据类型
22	/	除号	两个数字相除	m=3; n=2; m/n 1.5 m=2; n=3; m/n 0.6666666666666666	第5章：Python数据类型

序号	符号	名称	作用	说明和举例	详细介绍
23	//	整除号	两个数字相除，取结果的整数部分	m=3; n=2; m//n 1 m=2; n=3; m//n 0	第5章：Python数据类型
24	%	取模号	1.两个数字相除，取余数	j=17; k=3; j%k 2	第5章：Python数据类型
			2.配合特定字符使用，表示打印格式显示的格式	%d表示整型数字，%f表示浮点型数字，%s表示字符串，%%表示一个% num=9.99 print("%d" %num) 9 print("%f" %num) 9.990000 print("%s" %num) 9.99	
25	()	小括号	1.表示函数参数	def func(a,b):	第7章：Python函数
			2.表示继承的类	class C(A):	第8章：Python类和对象
			3.表示元组数据类型	a=(1,2,3)	第5章：Python数据类型
26	[]	中括号	1.表示列表数据类型	b=[1,2,3,4,5]	第5章：Python数据类型
			2.表示字符串、列表、元组、字典数据类型的元素的索引	s="hello"; a=[1,2,3]; b=('a','b','c'); c={'Name': 'Jim', 'Age': 20} s[0] 'h' a[0] 1 b[0] 'a' c['Name'] 'Jim'	第5章：Python数据类型
27	{}	大括号	1.表示集合数据类型（注意集合类型元素不能被索引）	j=set('abcde') j {'d', 'e', 'b', 'c', 'a'}	第5章：Python数据类型
			2.表示字典数据类型	k={'Name': 'Jim', 'Age': 20, 'Gender': 'Male'}	第5章：Python数据类型

序号	符号	名称	作用	说明和举例	详细介绍
28	_	单下划线	单下划线开始的变量相当于私有变量，只有类和子类实例能访问	<pre>class Dog: _hair = '棕色' dog1=Dog() print(dog1._hair) 棕色</pre>	第8章：Python类和对象
29	__	双下划线	1.双下划线表示类中的私有变量或方法名，只有类自己能访问，连类实例也不能直接访问，但可以通过类定义的getter函数访问	<pre>class Dog: __hair = '棕色' def set_hair(self): self.__hair='棕色' def get_hair(self): return self.__hair dog1=Dog() print(dog1.get_hair()) 棕色</pre>	第8章：Python类和对象
			2.开头和结尾都有双下划线的表示系统定义名字，是Python里特殊方法专用的标识，必须按规定使用	相当于Java程序的main入口的写法： <pre>if __name__ == '__main__':</pre>	
30	@	装饰器符号	1.类方法是不需要实例化对象就可以直接调用的方法，需要在该方法的上一行加上装饰器@classmethod	<pre>class Animal(): @classmethod #类方法要加上装饰器 def test2(cls): #类方法的默认参数是cls cls.num=100 print(cls.num) Animal.test2() 100</pre>	第8章：Python类和对象
			2.静态方法主要是用来存放逻辑性的代码，逻辑上属于类，但是和类本身没有关系，不会涉及到类中的属性和方法的操作，需要在该方法的上一行加上装饰器@staticmethod	<pre>import time class TimeTest(): @staticmethod #静态方法要加上装饰器 def showCurrentTime(): return time.strftime("%H:%M:%S", time.localtime()) TimeTest.showCurrentTime() '19:31:10'</pre>	第8章：Python类和对象