



3.4 常量、变量和表达式



3.4.1 数据类型、常量和变量

■ 基本数据类型

Python中支持**6种标准的数据类型**：

数字,字符串，列表，元组，字典，集合



□ 数字 (Numbers) —— 整型(Integer)

■ 正整数

```
int_a=65536
```

■ 负整数

```
int_b=-200
```

■ 零

```
int_c=0
```

Python3中的整数可以**任意大**，而不用担心位数不够而导致溢出的情况

```
>>>int_num=1234567899876543215578987651136  
>>>print(int_num,type(int_num))  
1234567899876543215578987651136 <class 'int'>
```



□ 数字 (Numbers) ——浮点数(Float)

小数 3.14, -0.28

太阳直径: 1392000千米——> 1.392×10^9 米——> 1.392E9

浮点数

```
>>>float_sun=1.392E9  
>>>print(float_sun,type(float_sun))  
1392000000.0 <class 'float'>
```

浮点数

```
>>>print(1.392E9, ";", 1.392e9, ";", 0.1392E10)  
1392000000.0; 1392000000.0; 1392000000.0;
```

整数

```
>>>print(type(1392000000))  
<class 'int'>
```



□ 数字 (Numbers) —— 布尔值 (Boolean)

True (真) —— 1 False (假) —— 0

~~true~~ ~~TRUE~~

~~false~~ ~~FALSE~~

□ 数字 (Numbers) —— 复数 (Complex)

$a+bi$

$1+2i$

实部

虚部



□ 字符串 (String)

- 使用**成对的单引号**或者**双引号**括起来。

```
"Python", "666", '3.5'
```

- 使用三重引号指定一个多行字符串。

```
>>> print('''  
这是一个文档字符串  
文档字符串可以跨越多行  
文档字符串可以使用三重单引号  
''')  
这是一个文档字符串  
文档字符串可以跨越多行  
文档字符串可以使用三重单引号
```

```
>>> print("""  
这也是一个文档字符串  
使用三重双引号  
""")  
这也是一个文档字符串  
使用三重双引号
```



3.4.1 数据类型、常量和变量

多行注释：可以直接使用**多行字符串**，作为程序的注释

多行注释

```
"""
```

```
这个程序的功能是接收用户输入  
并判断它是否大于0
```

```
"""
```

```
num=0
```

```
if num > 0:
```

```
    print ("num是正数")
```

```
else:
```

```
    print ("num可能是0")
```

```
    print ("num也可能是负数")
```



■ **标识符 (Identifier)**：变量、函数、数组、文件、对象等的**名字**。

- ❑ 标识符的**第1个字符**必须是**字母或下划线**
- ❑ 其他字符可以由字母、下划线、或数字组成。
- ❑ 标识符长度任意。

test_data, test_data_1, _y, _, 1test, 1_test ❌

- ❑ 标识符是**大小写敏感**的。score ,Score
- ❑ 标识符不能与**Python关键字**重名。
- ❑ 在Python3中，标识符**支持非英语字符**，只要是Unicode字符集支持的字符都可以。建议尽量使用英文的标识符。

在给标识符命名时，应该尽量采用有意义的标识符。



Python关键字

```
>>>help("keywords")  
Here is a list of the Python keywords. Enter any keyword  
to get more help.
```

False	def	if	raise
None	del	import	return
True	elif	in	try
and	else	is	while
as	except	lambda	with
assert	finally	nonlocal	yield
break	for	not	
class	from	or	
continue	global	pass	



■ 常量 (constant)

- 数字、字符串、布尔值、空值等

2, -10086, 3.5, "Python", True、False, None

- Python中没有**命名常量**,不能给常量起一个名字



■ 变量 (variable)

- Python的变量**不需要声明**
- 其**数据类型**由**所赋的值**来决定
- 不同类型的**数字型**数据运算时，会自动的进行**类型转换**
 $\text{bool} < \text{int} < \text{float} < \text{complex}$
- 自动的类型转换，仅存在于**数字型数据**之间

```
>>> print(1+True, 1+False)
2 1
```

```
>>> a=10
>>> b=3.5
>>> print(a+b, type(a+b))
13.5 <class 'float'>
```



■ 数据类型转换函数

函 数	功 能	示 例
<code>int(x,base)</code>	转换为整型	<code>int("123")</code>
<code>float()</code>	转换为浮点型	<code>float("123")</code>
<code>bool()</code>	转换为布尔型	<code>bool(1)</code>
<code>str()</code>	转换为字符串	<code>str(123)</code>
<code>chr()</code>	将整数转换为对应的ASCII字符	
<code>ord()</code>	将一个字符转换为对应的ASCII码	
<code>complex()</code>	转换为复数型	<code>complex(1)</code>



3.4.1 数据类型、常量和变量

```
#判断变量num是否是正数
num=int(input("请输入一个整数: "))
print ("您输入的整数是: %d" %(num))
if num > 0:
    print ("num是正数")
else:
    print ("num可能是0")
    print ("num也可能是负数")
```

```
>>>a="11"
>>>b=int(a)+1
>>>print("a=",a ,type(a))
>>>print("b=",b ,type(b))
a= 11 <class 'str'>
b= 12 <class 'int'>
```

向0取整

```
>>>print(int(3.6))
3
>>>print(int(-3.6))
-3
>>>print(int())
0
```

```
>>>print(chr(65))
A

>>>print(ord('A'))
65
```



□ 使用help()函数查看函数的用法

```
>>>help(int)
Help on class int in module builtins:
class int(object)
|   int(x=0) -> integer
|   int(x, base=10) -> integer
|
|   Convert a number or string to an integer, or return 0 if no arguments
|   are given. If x is a number, return x.__int__(). For floating point
|   numbers, this truncates towards zero.
|
|   If x is not a number or if base is given, then x must be a string,
|   bytes, or bytearray instance representing an integer literal in the
|   given base. The literal can be preceded by '+' or '-' and be surrounded
|   by whitespace. The base defaults to 10. Valid bases are 0 and 2-36.
|   Base 0 means to interpret the base from the string as an integer literal.
|   >>> int('0b100', base=0)
|   4
```

