



大连理工大学

未来技术学院 / 人工智能学院

SCHOOL OF FUTURE TECHNOLOGY, SCHOOL OF ARTIFICIAL INTELLIGENCE, DUT



变量与运算符

- 作用：用来存储一些之后可能会变化的值
- 例：className='机器学习'

变量名必须是大小写英文字母、数字或下划线 _ 的组合，不能用数字开头，并且对大小写敏感

关键字不能用于命名变量（31个），如and、as、assert、break、class、continue、def、del等

- 通过赋值运算符 =

- 变量名和想要赋予变量的值连接起来，变量的赋值操作就完成了声明和定义的过程，在其他语言中需要制定类型；
- 同一变量可以反复赋值，而且可以是不同类型的变量，这也是Python语言称之为动态语言的原因

```
In [1]: example=1  
print(example, type(example))  
example='AI'  
print(example, type(example))
```

```
1 <class 'int'>  
AI <class 'str'>
```



命名与保留字



大连理工大学

未来技术学院 / 人工智能学院

SCHOOL OF FUTURE TECHNOLOGY, SCHOOL OF ARTIFICIAL INTELLIGENCE, DUT

- 常量：程序中值不发生改变的元素
- 变量：程序中值发生改变或者可以发生改变的元素

Python语言允许采用大写字母、小写字母、数字、下划线(_)和汉字等字符及其组合给变量命名，但名字的首字符不能是数字，中间不能出现空格，长度没有限制

注意：标识符对大小写敏感，python和Python是两个不同的名字



命名与保留字



大连理工大学

未来技术学院 / 人工智能学院

SCHOOL OF FUTURE TECHNOLOGY, SCHOOL OF ARTIFICIAL INTELLIGENCE, DUT

- ✓ 保留字，也称为关键字，指被编程语言内部定义并保留使用的标识符。
- ✓ 程序员编写程序不能定义与保留字相同的标识符。
- ✓ 每种程序设计语言都有一套保留字，保留字一般用来构成程序整体框架、表达关键值和具有结构性的复杂语义等。
- ✓ 掌握一门编程语言首先要熟记其所对应的保留字。



命名与保留字



大连理工大学

未来技术学院 / 人工智能学院

SCHOOL OF FUTURE TECHNOLOGY, SCHOOL OF ARTIFICIAL INTELLIGENCE, DUT

✓ Python 3.x保留字列表 (33个)

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

常量?



- 常量表示“不能变”的变量
- Python中是没有常量的关键字的。只是我们常常约定使用大写字母组合的变量名表示常量，也有不要对其进行赋值”的提醒作用

```
PI = 3.14159265
```

数据类型



大连理工大学

未来技术学院 / 人工智能学院

SCHOOL OF FUTURE TECHNOLOGY, SCHOOL OF ARTIFICIAL INTELLIGENCE, DUT

- Python语言的数据类型包括整型、浮点型、字符串、布尔型和空值。

标量和非标量

标量对象是不可分的，即原子。非标量对象比如字符串，具有内部结构的。

整型数据 (int)



- 整型的取值为整数，有正有负，如 2，-666，666 等。

```
integer=input('请输入一个整数：')  
print(integer, type(integer))  
integer=int(integer)  
print(integer, type(integer))
```

请输入一个整数： 23

23 <class 'str'>

23 <class 'int'>

浮点型数据 (float)



- 浮点型的取值为小数，当计算有精度要求时被使用，由于小数点可以在相应的二进制的不同位置浮动，故而称为浮点数

例如， 3.14， -6.66 等，但是如果是非常大或者非常小的浮点数，就需要使用科学计数法表示，用 e 代替 10 。

```
In [7]: 2.14e5
```

```
Out[7]: 214000.0
```

```
In [6]: complexData=2.4+5.6j  
print(complexData.real, complexData.imag)
```

```
2.4 5.6
```

布尔型 (bool)



- 布尔型只有 True 和 False 两种值。比较运算和条件表达式都会产生 True 或 False。

```
In [13]: boolData1=True  
print(boolData1)  
boolData2=2>3  
print(boolData2)  
boolData3=bool(-4)  
print(boolData3, bool(0))
```

True

False

True False

布尔型



- 布尔型 (bool)

布尔值可以进行 and 、 or 和 not 运算

and 运算

	True	False
True	True	False
False	False	False

or 运算

	True	False
True	True	True
False	True	False

空值 (NoneType)



- 空值是Python里一个特殊的值，用 None 表示，一般用 None 填充表格中的缺失

```
[14]: noneData=None  
print(noneData, type(noneData))  
  
None <class 'NoneType'>
```



字符串 (str)



- 字符串或串(str)是由数字、字母、下划线组成的一串字符，一般以两个单引号或两个双引号包裹起来的文本。

单引号和双引号等价

```
str1 = 'AI Programming'
```

```
str2 = "AI Programming"
```

```
In [28]: print("AI Programming")  
          print('AI Programming')  
          print("It's good")
```

AI Programming

AI Programming

It's good

输出中含有单引号的时候，可以用双引号来避免歧义

```
print("it's good")
```

等价于

```
print('it\'s good')
```



字符串 (str)



- 语句中单引号和双引号都有，那么可以使用三引号。

```
str3='''It's "C",I have "A" and "B"。'''
```

```
str4="""It's "C",I have "A" and "B"。"""
```

In [31]:

```
str5="""It's "C"  
I have "A" and "B"。"""  
print(str5)
```

```
It's "C"  
I have "A" and "B"。
```

In [35]:

```
print('AI \nProgramming')
```

```
AI  
Programming
```

字符串



- 转义字符：字符串里常常存在一些如换行、制表符等有特殊含义的字符，这些字符称之为转义字符。

例：

\n 表示换行

\t 表示制表符，

r“ ” 表示“ ” 内部的字符串

默认不转义

```
In [49]: print('AI Programming')  
          print('AI \t Programming')  
          print(r'AI \t Programming')
```

```
AI Programming  
AI      Programming  
AI \t Programming
```




其他类型数据



- 日期数据类型

```
In [36]: import time
```

```
In [43]: print(time.strftime("%Y-%m-%d %H:%M:%S", time.localtime()))
```

```
2019-10-18 18:52:17
```



算术运算符



大连理工大学

未来技术学院 / 人工智能学院

SCHOOL OF FUTURE TECHNOLOGY, SCHOOL OF ARTIFICIAL INTELLIGENCE, DUT

运算符	说明	实例	结果
+	加	$12.45 + 15$	27.45
-	减	$4.56 - 0.26$	4.3
*	乘	$5 * 3.6$	18.0
/	除	$7 / 2$	3.5
%	取余，即返回除法的 余数	$7 \% 2$	1
//	整除，返回商的整数 部分	$7 // 2$	3
**	幂，即返回 x 的 y 次方	$2 ** 4$	16，即 2^4



算术运算符



大连理工大学

未来技术学院 / 人工智能学院

SCHOOL OF FUTURE TECHNOLOGY, SCHOOL OF ARTIFICIAL INTELLIGENCE, DUT

整数与浮点数运算关系

$3+2=5$	$3.0+2=5.0$
$3/2=1.5$	$3//2=1$
$3.0/2=1.5$	$3.0//2=1.0$
$3.0/2.0=1.5$	$3.0//2.0=1.0$
$3\%2=1$	$3\%2.0=1.0$



比较运算符



大连理工大学

未来技术学院 / 人工智能学院

SCHOOL OF FUTURE TECHNOLOGY, SCHOOL OF ARTIFICIAL INTELLIGENCE, DUT

a=10,b=20

运算符	描述	实例
==	等于 - 比较对象是否相等	(a == b) 返回 False
!=	不等于 - 比较两个对象是否不相等	(a != b) 返回 True
>	大于 - 返回x是否大于y	(a > b) 返回 False
<	小于 - 返回x是否小于y	(a < b) 返回 True
>=	大于等于 - 返回x是否大于等于y	(a >= b) 返回 False
<=	小于等于 - 返回x是否小于等于y	(a <= b) 返回 True



赋值运算符



大连理工大学

未来技术学院 / 人工智能学院

SCHOOL OF FUTURE TECHNOLOGY, SCHOOL OF ARTIFICIAL INTELLIGENCE, DUT

运算符	描述	实例
=	简单的赋值运算符	$c = a + b$ 将 $a + b$ 的运算结果赋值为 c
+=	加法赋值运算符	$c += a$ 等效于 $c = c + a$
-=	减法赋值运算符	$c -= a$ 等效于 $c = c - a$
*=	乘法赋值运算符	$c *= a$ 等效于 $c = c * a$
/=	除法赋值运算符	$c /= a$ 等效于 $c = c / a$
%=	取模赋值运算符	$c \% = a$ 等效于 $c = c \% a$
**=	幂赋值运算符	$c ** = a$ 等效于 $c = c ** a$
//=	取整除赋值运算符	$c //= a$ 等效于 $c = c // a$



逻辑运算符



大连理工大学

未来技术学院 / 人工智能学院

SCHOOL OF FUTURE TECHNOLOGY, SCHOOL OF ARTIFICIAL INTELLIGENCE, DUT

a=10,b=20

运算符	逻辑表达式	描述	实例
and	x and y	布尔"与" - 如果 x 为 False，x and y 返回 False，否则它返回 y 的计算值。	(a and b) 返回 20。
or	x or y	布尔"或" - 如果 x 是 True，它返回 x 的值，否则它返回 y 的计算值。	(a or b) 返回 10。
not	not x	布尔"非" - 如果 x 为 True，返回 False。如果 x 为 False，它返回 True。	not(a and b) 返回 False

位运算符 (**)



运算符	描述	实例
&	按位与运算符：参与运算的两个值，如果两个相应位都为1，则该位的结果为1，否则为0	(a & b) 输出结果 12 ， 二进制解释：0000 1100
	按位或运算符：只要对应的二个二进制位有一个为1时，结果位就为1。	(a b) 输出结果 61 ， 二进制解释：0011 1101
^	按位异或运算符：当两对应的二进制位相异时，结果为1	(a ^ b) 输出结果 49 ， 二进制解释：0011 0001
~	按位取反运算符：对数据的每个二进制位取反,即把1变为0,把0变为1。 $\sim x$ 类似于 $-x-1$	(~a) 输出结果 -61 ， 二进制解释：1100 0011， 在一个有符号二进制数的补码形式。
<<	左移动运算符：运算数的各二进制位全部左移若干位，由"<<"右边的数指定移动的位数，高位丢弃，低位补0。	a << 2 输出结果 240 ， 二进制解释：1111 0000
>>	右移动运算符：把">>"左边的运算数的各二进制位全部右移若干位，">>"右边的数指定移动的位数	a >> 2 输出结果 15 ， 二进制解释：0000 1111

注释



- Python语言会通过注释符号识别出注释的部分，将它们当做纯文本，并在执行代码时跳过这些纯文本
- 在Python语言中，使用#进行行注释

```
# 这是在代码之前的一个注释!
```

```
shot_id = 1 # 这是在代码所在行的一个注释!
```

```
# 这是在代码之后的一个注释!
```


注释



- 多行注释使用连续单个单引号或者双引号

```
' ' '
```

```
这是第一行注释!
```

```
这是第二行注释!
```

```
' ' '
```

```
" " "
```

```
这是第一行注释!
```

```
这是第二行注释!
```

```
" " "
```

总结



- 变量、注释、print函数
- 数据类型、算术运算符、类型转换