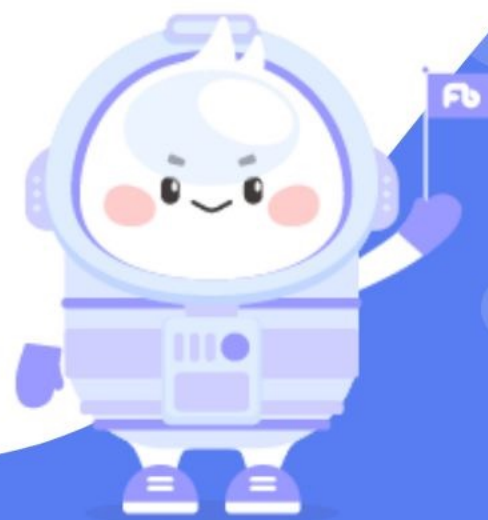


《信息技术》

Python程序设计 1/4

► 讲师：孙珍珍

更多干货关注  粉笔教师教育  粉笔教师

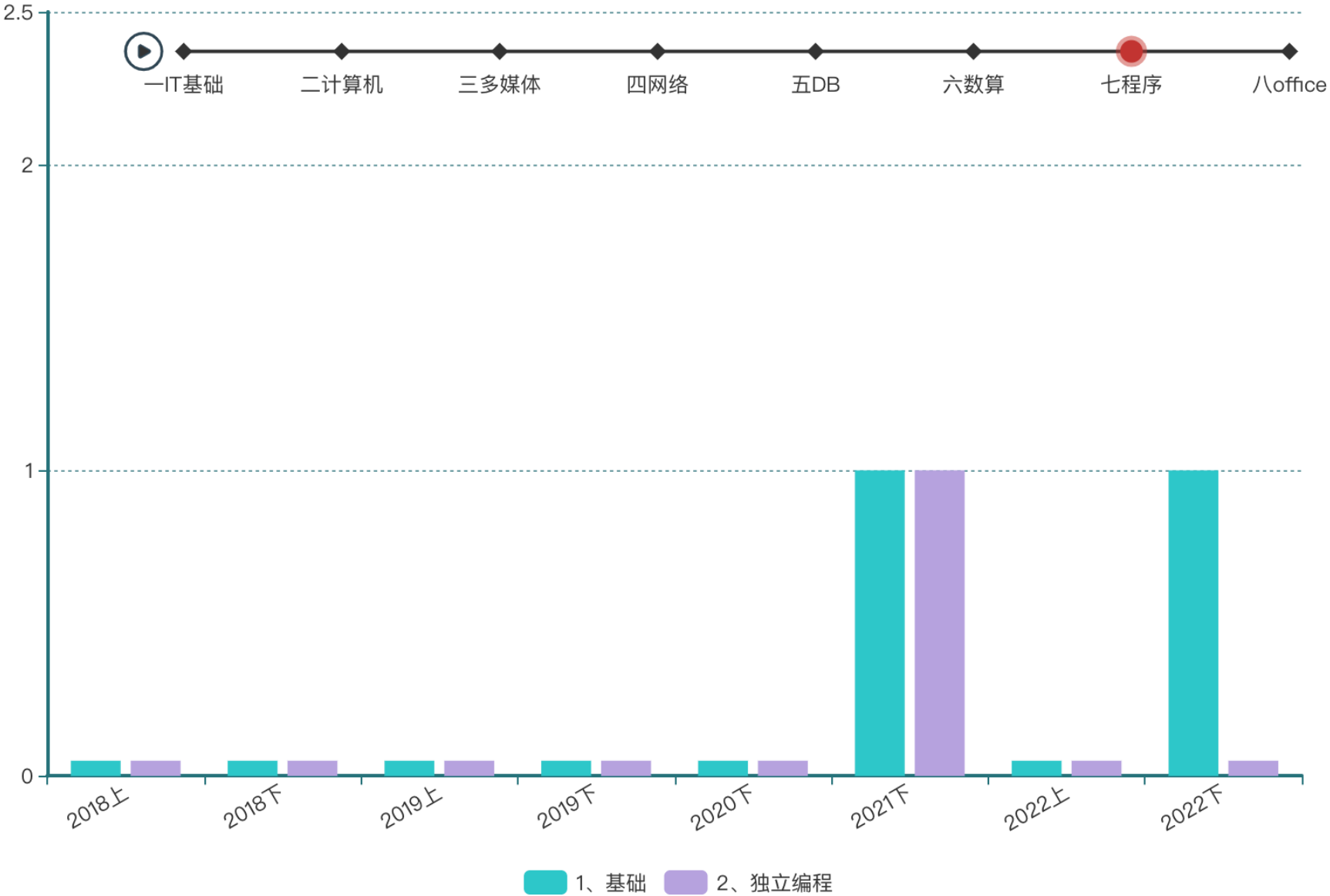


✿ 复习一下

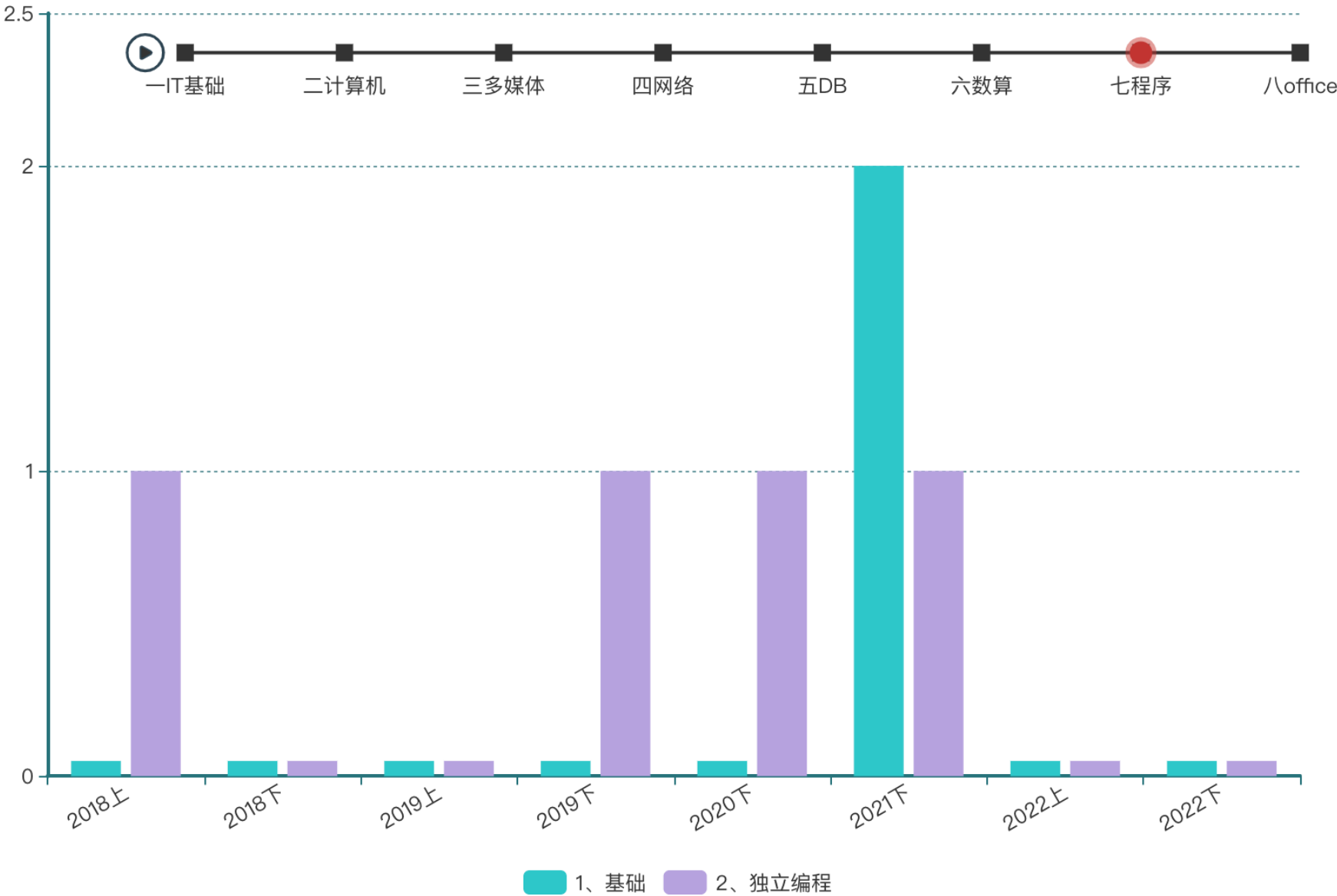


第七章	Python 程序设计	338
第一节	Python 程序设计基础	338
第二节	选择结构程序设计	356
第三节	循环结构程序设计	361
第四节	序 列	365
第五节	函 数	377

【初中】-近8次考试专业知识-节占比



【高中】-近八次考试专业知识-节占比





第一节 Python程序设计基础

(一) 认识Python 程序

【例1】编写程序，计算半径为2 的圆的面积。

```
>>> r = 2
>>> s = 3.14 * r * r
>>> print("面积为: ", s)
面积为:  12.56
>>> |
```

命令行交互式环境

```
1  r = 2
2  s = 3.14 * r * r
3  print("面积为: ", s)
4  |
```

PyCharm编辑器

【例2】编写程序，输入三角形三边（假设输入的是合法的三角形的三边），计算三角形的面积。

```
1  import math                                # 导入math模块
2  a = int(input("请输入三角形的第一条边: ")) # 输入第一条边并将其转换为整型
3  b = int(input("请输入三角形的第二条边: ")) # 输入第二条边并将其转换为整型
4  c = int(input("请输入三角形的第三条边: ")) # 输入第三条边并将其转换为整型
5  s = 1 / 2 * (a + b + c)                    # 计算s
6  area = math.sqrt(s * (s - a) * (s - b) * (s - c)) # 调用sqrt函数计算面积
7  print("此三角形面积为: ", area)           # 输出三角形面积
```

提示：若三角形三条边长为 a 、 b 、 c

则三角形的面积为 $\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$ ，
其中 $s = (a + b + c)/2$ 。

```
请输入三角形的第一条边: 3
请输入三角形的第二条边: 4
请输入三角形的第三条边: 5
此三角形面积为: 6.0
```



```
1 r = 2          #半径r
2 s = 3.14 * r * r    #圆面积s
3 print('面积为: ', s)  #打印面积值
```

```
1 import math          # 导入math模块
2 a = int(input("请输入三角形的第一条边: "))    # 输入第一条边并将其转换为整型
3 b = int(input("请输入三角形的第二条边: "))    # 输入第二条边并将其转换为整型
4 c = int(input("请输入三角形的第三条边: "))    # 输入第三条边并将其转换为整型
5 s = 1 / 2 * (a + b + c)    # 计算s
6 area = math.sqrt(s * (s - a) * (s - b) * (s - c))    # 调用sqrt函数计算面积
7 print("此三角形面积为: ", area)    # 输出三角形面积
```

◆ 变量、常量、数据类型、运算符和表达式、函数、字符串、注释

(一) 变量的命名

- 变量名只能由字母、数字和下划线组成
- 字母区分大小写
- 第一个字符必须是字母或下划线，不能是数字
- 不能使用Python 的关键字（保留字）

下列选项中，不能作为Python程序变量名的是（ ）。

A.abc

B.abc123

C.123abc

D.abc_123



1. 整型 (int)

```
>>> b = 11
>>> print(b)
11
```

十进制

- 无前缀标识
- 值为0~9的组合

```
>>> b = 0b11
>>> print(b)
3
```

二进制【补】

- 前缀标识0b
- 值为0~1的组合

```
>>> b = 0o11
>>> print(b)
9
```

八进制

- 前缀标识0o
- 值为0~7的组合

```
>>> b = 0x11
>>> print(b)
17
```

十六进制

- 前缀标识0x
- 值为0~9、A~F的组合

2. 浮点型 (float)

- **十进制形式**：由数值和小数点组成； 前可省、后可省、前后不可同时省
- **指数形式**：由数值和字母e组成； 前有数、后有数、其后必须为整数

```
>>> c1=1.3
>>> c1
1.3
>>> c2=.3
>>> c2
0.3
>>> c3=1.
>>> c3
1.0
>>> c4=.
File "<stdin>", line 1
  c4=.
    ^
SyntaxError: invalid syntax
```

```
>>> c1=1e3
>>> c1
1000.0
>>> c2=1e-3
>>> c2
0.001
>>> c3=1.3e1
>>> c3
13.0
>>> c4=3e1.3
File "<stdin>", line 1
  c4=3e1.3
    ^
SyntaxError: invalid syntax
```

3. 布尔类型 (bool)

- 只有两个值：True、False

4. 字符串 (string)

- 用一对引号括起来 (单双均可)
- 里面有单引号，外面用双引号
- 里面有双引号，外面用单引号
- 认识转义字符，以 \ 开头

转义字符	说 明
\'	单引号'
\"	双引号"
\n	回车换行

```
print("1. hello~")
print('2. hi~~')
print("3. I'm OK~")
print('4. I love "Python"')
print('5. I\'m \'OK\'')
print("6. I'm learning\nPython.")
```

```
1. hello~
2. hi~~
3. I'm OK~
4. I love "Python"
5. I'm "OK"
6. I'm learning
Python.
```

5.数据类型转换

	int()	float()	str()
结果	整型	浮点型	字符串型
参数类型	浮点数、整数字符串	整数、数值型字符串	整数、浮点型

```
>>> int(3.4)
3
>>> int(3.6)
3
>>> int(-3.6)
-3
>>> int('3')
3
>>> int('3.4')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() with base 10
```

```
>>> float(3)
3.0
>>> float('3')
3.0
>>> float('3.5')
3.5
>>> float('a')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: could not convert string to float: 'a'
```

```
>>> str(3)
'3'
>>> str(3.5)
'3.5'
```

1. 单个变量赋值

➤ 变量名 = 值

```
>>> a = 123
>>> b = 4.56
>>> c = 'hello'
>>> type(a)
<class 'int'>
>>> type(b)
<class 'float'>
>>> type(c)
<class 'str'>
```

2. 多个变量赋值

① 通用：多值多变量，用逗号

② 特殊：等值多变量，用等号

```
>>> a = b = c = 1
>>> print(a, b, c)
1 1 1
>>> a1, b1, c1 = 2, 2, 2
>>> print(a1, b1, c1)
2 2 2
>>> a2, b2, c2 = 3, 5.6, 'hi'
>>> print(a2, b2, c2)
3 5.6 hi
```


在`x,y,z = 1,'hello','python'`表达式中，变量`x`的值是（ ）。

- A. '1'
- B. 'hello'
- C. 'python'
- D. 1

二、变量

命名 — 组成：____、____、____ — 大小写 — 首字符不能是____

数据类型

- 整型
 - 类型：—— 十进制 —— 二进制 —— 八进制 —— 十六进制
 - 前缀：—— ———— ———— ————
 - 组合：—— 0~9 —— 0~1 —— 0~7 —— 0~9 A~F

- 浮点型
 - 直接形式 —— 组成：数字和____ —— 规则：_____
 - 指数形式 —— 组成：数字和____ —— 规则：_____
e 等价于____的幂次

- 字符串型 —— 用一对____括起来 —— 转移字符：以____开头

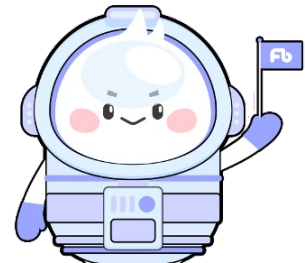
- 强制转换
 - int() —— 转为____ —— 参数类型：_____
 - ____ —— 转为浮点型 —— 参数类型：_____
 - str() —— 转为____ —— 参数类型：_____

- 布尔型 —— 两个值：____（真） 和 ____（假）

赋值

- 单个变量 —— 变量名=值
- 多个变量 —— 通用：____，如a1,a2=3,3 —— 特殊：等号，如a1=a2=3

判断合法性
会使用



二、变量

命名 — 组成：字母、数字、下划线 — 区分大小写 — 首字符不能是数字

★ 数据类型

整型

- 类型： — 十进制 — 二进制 — 八进制 — 十六进制
- 前缀： — 无 — 0b — 0o — 0x
- 组合： — 0~9 — 0~1 — 0~7 — 0~9 A~F

浮点型

- 直接形式 — 组成：数字和小数点 — 规则：前可省，后可省，前后不可同时省
- 指数形式 — 组成：数字和字母e —
 - 规则：前有数，后有数，其后必为整数
 - e 等价于【10】的幂次

字符串型 — 用一对【引号】括起来 — 转移字符：以【\】开头

强制转换

- int() — 转为整型 — 参数类型：浮点数、整数字符串
- float() — 转为浮点型 — 参数类型：整数、数值型字符串
- str() — 转为字符串型 — 参数类型：整数、浮点型

布尔型 — 两个值：True（真） 和 False（假）

赋值

单个变量 — 变量名=值

多个变量 — 通用：逗号，如a1,a2=3,3 — 特殊：等号，如a1=a2=3



(一) 算术运算符

运算符	说明
+	加法
-	减法
*	乘法
/	除法
//	整除 (求商)
%	模 (求余)
**	幂 (求次方)

```
1 a, b = 5, 2
2 print('a + b = ', a + b)
3 print('a - b = ', a - b)
4 print('a * b = ', a * b)
5 print('a / b = ', a / b)
6 print('a // b = ', a // b)
7 print('a % b = ', a % b)
8 print('a ** b = ', a ** b)
```

```
a + b = 7
a - b = 3
a * b = 10
a / b = 2.5
a // b = 2
a % b = 1
a ** b = 25
```

1.除 / : 精准除(除以) ; 结果为浮点数

2.优先级 : 幂 > 负号 > 乘、除、商、余 > 加、减

1. 数学式 $1+2^3\times 2\div 4-2$ 对应的Python表达式是 ()。

A. $1+2^3*2/4-2$

B. $1+2**3*2/4-2$

2. 表达式 $1+2**3*2/4-2$ 的执行结果是 ()。

A. 3

B. 3.0



1. 简单赋值

- 格式：变量名 = 表达式
- 优先级：算术 > 赋值

2. 复合赋值

$a += b$	#相当于 $a = a + b$
$a -= b$	#相当于 $a = a - b$
$a *= b$	#相当于 $a = a * b$
$a /= b$	#相当于 $a = a / b$
$a //= b$	#相当于 $a = a // b$
$a \% = b$	#相当于 $a = a \% b$
$a ** = b$	#相当于 $a = a ** b$

```
a = 4           # 常规赋值

b = a + 1       # 先计算再赋值

a /= 2          # 复合赋值, 且a类型发生改变

b *= b - 2      # 复合赋值, 注意括号
```

运算符	功能
>	大于
>=	大于等于
<	小于
<=	小于等于
==	等于
!=	不等于

```
a, b, c = 1, 1, 2
print("1、", a > b)
print("2、", a == b)
print("3、", c > a > b)
print("4、", c > a >= b)
print("5、", a == c > b)
print("6、", a <= b + c)
a = b > c
print("7、", a)
```

```
1、 False
2、 True
3、 False
4、 True
5、 False
6、 True
7、 False
```

1.结果：True（真）、False（假）

2.执行：

- 若全部为关系运算符，则依次看每段的结果，全真为真，有假为假
- 若有其它运算符，则按优先级顺序执行。【算术 > 关系（相同）> 赋值】

1. 逻辑与：and

```
1 a, b = 3, 0
2 print(a and b)
3 print(a > b and a < b)
4 print(a > b and a + b + 1)
5 print("-----")
6 print(b and a)
7 print(a < b and a < b)
8 print(a < b and a >= b)
```

```
0
False
4
-----
0
False
False
```

1. 总规律：数值0为逻辑假（False），非零值为逻辑真（True）

2. and，与，遇到假或最后时，结束运算，结果是其表达式的值

2. 逻辑或：or

```
1 a, b = 3, 0
2 print(a or b)
3 print(a > b or a < b)
4 print(a > b or a >= b)
5 print("-----")
6 print(b or a)
7 print(a < b or a < b)
8 print(a < b or a + b - 1)
```

```
3
True
True
-----
3
False
2
```

1. 总规律：数值0为逻辑假（False），非零值为逻辑真（True）

2. or，或，遇到真或最后时，结束运算，结果是其表达式的值

设 $a=1$, $b=2$, $c=3$, $d=4$, 表达式 $a>b$ or $c\leq d$ and $2*a>c$ 的值是 ()。

优先级：and > or

- A. False
- B. True
- C. -1
- D. 1

3. 逻辑非：not

```
1 a, b = 3, 0
2 # 返回相反的逻辑值
3 print(not a)
4 print(not b)
5 print(not b + 1)
```

```
False
True
False
```

1. 总规律：数值0为逻辑假（False），非零值为逻辑真（True）

2. not，非，结果是取相反的逻辑值

运算符	举 例	说 明
in	x in y	在y中找到x的值返回True，否则返回False
not in	x not in y	在y中未找到x的值返回True，否则返回False

```
1 a, b, c = '1', '5', 'CD'
2 s = '123CDE'
3 print(a in s)
4 print(b in s)
5 print(c not in s)
```

```
True
False
False
```

运算符	举 例	说 明
is	x is y	如果x和y引用的是同一个对象则返回True，否则返回False
is not	x is not y	如果x和y引用的不是同一个对象则返回True，否则返回False

```
a = 10
b = 10
print(a == b)
print(id(a))
print(id(b))
print(a is b)
```

```
True
140286021999184
140286021999184
True
```

```
a = "hello"
b = "hi"
print(a == b)
print(id(a))
print(id(b))
print(a is b)
```

```
False
140356354400624
140356354403696
False
```

```
c = ['1', '2', '3']
d = ['1', '2', '3']
print(c == d)
print(id(c))
print(id(d))
print(c is d)
```

```
True
140673998480256
140673998950656
False
```

◆ 指数值的补码进行运算，但输出的是原码

运算符	名称	规则
&	与	全1为1，有0为0
	或	全0为0，有1为1
^	异或	不同为1
~	取反	取相反值
<<	左移	右侧空余位补0
>>	右移	右侧移除位删掉

```
1 a = 10
2 b = 23
3 print("a&b=", a & b)
4 print("a|b=", a | b)
5 print("a^b=", a ^ b)
6 print("~a=", ~a)
7 print("a<<2=", a << 2)
8 print("a>>2=", a >> 2)
```

```
a&b= 2
a|b= 31
a^b= 29
~a= -11
a<<2= 40
a>>2= 2
```

00001010

00010111

00001010

00010111

00001010

00010111

◆ 指数值的补码进行运算，但输出的是原码

运算符	名称	规则
&	与	全1为1，有0为0
	或	全0为0，有1为1
^	异或	不同为1
~	取反	取相反值
<<	左移	右侧空余位补0
>>	右移	右侧移除位删掉

00001010

```
1 a = 10
2 b = 23
3 print("a&b=", a & b)
4 print("a|b=", a | b)
5 print("a^b=", a ^ b)
6 print("~a=", ~a)
7 print("a<<2=", a << 2)
8 print("a>>2=", a >> 2)
```

```
a&b= 2
a|b= 31
a^b= 29
~a= -11
a<<2= 40
a>>2= 2
```

◆ 指数值的补码进行运算，但输出的是原码

运算符	名称	规则
&	与	全1为1，有0为0
	或	全0为0，有1为1
^	异或	不同为1
~	取反	取相反值
<<	左移	右侧空余位补0
>>	右移	右侧移除位删掉

00001010

```
1 a = 10
2 b = 23
3 print("a&b=", a & b)
4 print("a|b=", a | b)
5 print("a^b=", a ^ b)
6 print("~a=", ~a)
7 print("a<<2=", a << 2)
8 print("a>>2=", a >> 2)
```

00001010

```
a&b= 2
a|b= 31
a^b= 29
~a= -11
a<<2= 40
a>>2= 2
```


-- 讲义纠错 P350 --

(八) 优先级

优先级顺序	运算符	说明
1	**	幂（乘方运算）
2	~ + -	取反、正号运算和负号运算
3	* / % //	乘法、除法、模和整除运算
4	+ -	加法、减法运算
5	>> <<	右移、左移位运算符
6	&	按位与位运算符
7	^	按位异或和按位或位运算符
8	<= < > >=	小于、小于等于、大于、大于等于关系运算符
9	= %= /= //=- += *= **=	赋值运算符
10	is is not	身份运算符
11	in not in	成员运算符
12	not and or	逻辑运算符

优先级顺序	运算符	说 明
1	**	算术运算符：指数（次幂）运算
2	~ + -	单目运算符：取反、正号和负号
3	* / // %	算术运算符：乘、除、商、余
4	+ -	算术运算符：加、减
5	>> <<	位运算符：右移，左移位运算符
6	&	位运算符：按位与
7	^	位运算符：按位异或
8		位运算符：按位或
9	in not in is is not	成员运算符、身份运算符
	<= < > >= == !=	关系运算符
10	not	逻辑运算符：非
11	and	逻辑运算符：与
12	or	逻辑运算符：或
13	= %= /= //= -= += *= **=	赋值运算符

运算符

- 算术**
 - 分类：—— 幂 —— 负 —— 乘、除、商、余 —— 加、减
 - 符号：—— — —— — —— —、—、—、— —— +、—
 - 说明：—— /：结果是——
- 位运算**
 - 分类：—— 按位取反 —— 左移、右移 —— 按位与 —— 按位异或 —— 按位或
 - 符号：—— — —— —、— —— — —— — —— —
- 身份** —— 含义：判断——是否相等 —— 符号：——
- 成员** —— 含义：判断是否——某个序列中 —— 符号：——
- 关系**
 - 分类：—— 大于、大于等于、小于、小于等于、等于、不等于
 - 符号：—— > 、——、 < 、——、——、——
 - 说明：—— 计算每段表达式，——为真，——为假
- 逻辑**
 - 分类：—— 取反 —— 逻辑与 —— 逻辑或
 - 符号：—— — —— — —— —
 - 说明：
 - and：遇到——或最后时，结束运算，结果是——的值
 - or：遇到——或最后时，结束运算，结果是——的值
- 赋值** —— 符号：= —— 变量在——侧 —— 复合赋值时，右侧表达式为一个——

数值0为假
非零值为真



运算符

- 算术**
 - 分类：—— 幂 —— 负 —— 乘、除、商、余 —— 加、减
 - 符号：—— ** —— - —— *、/、//、% —— +、-
 - 说明：—— /: 结果是【小数】
- 位运算**
 - 分类：—— 按位取反 —— 左移、右移 —— 按位与 —— 按位异或 —— 按位或
 - 符号：—— ~ —— <<、>> —— & —— ^ —— |
- 身份** —— 含义：判断【内存地址】是否相等 —— 符号：is
- 成员** —— 含义：判断是否【在】某个序列中 —— 符号：in
- 关系**
 - 分类：—— 大于、大于等于、小于、小于等于、等于、不等于
 - 符号：—— > 、 >= 、 < 、 <= 、 == 、 !=
 - 说明：—— 【依次】计算每段表达式，全真为真，有假为假
- 逻辑**
 - 分类：—— 取反 —— 逻辑与 —— 逻辑或
 - 符号：—— not —— and —— or
 - 说明：
 - and: 遇到【假】或最后时，结束运算，结果是【其表达式】的值
 - or: 遇到【真】或最后时，结束运算，结果是【其表达式】的值
- 赋值** —— 符号：= —— 变量在【左】侧 —— 复合赋值时，右侧表达式为一个【整体】

数值0为假
非零值为真





有疑问没？等你吖

下节内容

第一节	Python 程序设计基础	338
	P350 ~ P360	
第二节	选择结构程序设计	356



岸上等你

THE TEST

光芒万丈
不负理想

粉笔
教师



机读卡

姓名:

考号:

