快速索引：

[Java](http://tool.oschina.net/commons?type=6#java_)[C++](http://tool.oschina.net/commons?type=6#cpp_)[C语言](http://tool.oschina.net/commons?type=6#c_)[PHP](http://tool.oschina.net/commons?type=6#php_)[Python](http://tool.oschina.net/commons?type=6#python_)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Java 运算符优先级 | | | | |
| **运算符** | | | **优先级** | |
| postfix | | | *expr*++ *expr*-- | |
| unary | | | ++*expr* --*expr* +*expr* -*expr* ~ ! | |
| multiplicative | | | \* / % | |
| additive | | | + - | |
| shift | | | << >> >>> | |
| relational | | | < > <= >= instanceof | |
| equality | | | == != | |
| bitwise AND | | | & | |
| bitwise exclusive OR | | | ^ | |
| bitwise inclusive OR | | | | | |
| logical AND | | | && | |
| logical OR | | | || | |
| ternary | | | ? : | |
| assignment | | | = += -= \*= /= %= &= ^= |= <<= >>= >>>= | |
| C++运算符优先级 | | | | |
| **Precedence** | **Operator** | **Description** | | **Associativity** |
| **1** | :: | Scope resolution | | Left-to-right |
| **2** | ++   -- | Suffix/postfix increment and decrement | |
| () | Function call | |
| [] | Array subscripting | |
| . | Element selection by reference | |
| −> | Element selection through pointer | |
| **3** | ++   -- | Prefix increment and decrement | | Right-to-left |
| +   − | Unary plus and minus | |
| !   ~ | Logical NOT and bitwise NOT | |
| (*type*) | Type cast | |
| \* | Indirection (dereference) | |
| & | Address-of | |
| sizeof | Size-of | |
| new, new[] | Dynamic memory allocation | |
| delete, delete[] | Dynamic memory deallocation | |
| **4** | .\*   ->\* | Pointer to member | | Left-to-right |
| **5** | \*   /   % | Multiplication, division, and remainder | |
| **6** | +   − | Addition and subtraction | |
| **7** | <<   >> | Bitwise left shift and right shift | |
| **8** | <   <= | For relational operators < and ≤ respectively | |
| >   >= | For relational operators > and ≥ respectively | |
| **9** | ==   != | For relational = and ≠ respectively | |
| **10** | & | Bitwise AND | |
| **11** | ^ | Bitwise XOR (exclusive or) | |
| **12** | | | Bitwise OR (inclusive or) | |
| **13** | && | Logical AND | |
| **14** | || | Logical OR | |
| **15** | ?: | Ternary conditional | | Right-to-Left |
| **16** | = | Direct assignment (provided by default for C++ classes) | |
| +=   −= | Assignment by sum and difference | |
| \*=   /=   %= | Assignment by product, quotient, and remainder | |
| <<=   >>= | Assignment by bitwise left shift and right shift | |
| &=   ^=   |= | Assignment by bitwise AND, XOR, and OR | |
| **17** | throw | Throw operator (for exceptions) | |
| **18** | , | Comma | | Left-to-right |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C语言运算符优先级 | | | | |
| **Precedence** | **Operator** | **Description** | | **Associativity** |
| **1** | ++ -- | Suffix/postfix increment and decrement | | Left-to-right |
| () | Function call | |
| [] | Array subscripting | |
| . | Structure and union member access | |
| −> | Structure and union member access through pointer | |
| (*type*){*list*} | Compound literal(C99) | |
| **2** | ++ -- | Prefix increment and decrement | | Right-to-left |
| + − | Unary plus and minus | |
| ! ~ | Logical NOT and bitwise NOT | |
| (*type*) | Type cast | |
| \* | Indirection (dereference) | |
| & | Address-of | |
| sizeof | Size-of | |
| \_Alignof | Alignment requirement(C11) | |
| **3** | \* / % | Multiplication, division, and remainder | | Left-to-right |
| **4** | + − | Addition and subtraction | |
| **5** | << >> | Bitwise left shift and right shift | |
| **6** | < <= | For relational operators < and ≤ respectively | |
| > >= | For relational operators > and ≥ respectively | |
| **7** | == != | For relational = and ≠ respectively | |
| **8** | & | Bitwise AND | |
| **9** | ^ | Bitwise XOR (exclusive or) | |
| **10** | | | Bitwise OR (inclusive or) | |
| **11** | && | Logical AND | |
| **12** | || | Logical OR | |
| **13** | ?: | Ternary conditional | | Right-to-Left |
| **14** | = | Simple assignment | |
| += −= | Assignment by sum and difference | |
| \*= /= %= | Assignment by product, quotient, and remainder | |
| <<= >>= | Assignment by bitwise left shift and right shift | |
| &= ^= |= | Assignment by bitwise AND, XOR, and OR | |
| **15** | , | Comma | | Left-to-right |
| PHP运算符优先级 | | | | |
| **结合方向** | **运算符** | | **附加信息** | |
| 非结合 | clone new | | clone 和 new | |
| 左 | [ | | array() | |
| 非结合 | ++ -- | | 递增／递减运算符 | |
| 非结合 | ~ - (int) (float) (string) (array) (object) (bool) @ | | 类型 | |
| 非结合 | instanceof | | 类型 | |
| 右结合 | ! | | 逻辑操作符 | |
| 左 | \* / % | | 算术运算符 | |
| 左 | + - . | | 算术运算符 和 字符串运算符 | |
| 左 | << >> | | 位运算符 | |
| 非结合 | < <= > >= <> | | 比较运算符 | |
| 非结合 | == != === !== | | 比较运算符 | |
| 左 | & | | 位运算符 和 引用 | |
| 左 | ^ | | 位运算符 | |
| 左 | | | | 位运算符 | |
| 左 | && | | 逻辑运算符 | |
| 左 | || | | 逻辑运算符 | |
| 左 | ? : | | 三元运算符 | |
| 右 | = += -= \*= /= .= %= &= |= ^= <<= >>= | | 赋值运算符 | |
| 左 | and | | 逻辑运算符 | |
| 左 | xor | | 逻辑运算符 | |
| 左 | or | | 逻辑运算符 | |
| 左 | , | | 多处用到 | |

这个表给出Python的运算符优先级（从低到高）.

从最低的优先级（最松散地结合）到最高的优先级（最紧密地结合）。

这意味着在一个表达式中，Python会首先计算表中较下面的运算符，然后在计算列在表上部的运算符。

|  |  |
| --- | --- |
| Python 运算符优先级 | |
| **运算符** | **描述** |
| lambda | Lambda表达式 |
| or | 布尔“或” |
| and | 布尔“与” |
| not x | 布尔“非” |
| in，not in | 成员测试 |
| is，is not | 同一性测试 |
| <，<=，>，>=，!=，== | 比较 |
| | | 按位或 |
| ^ | 按位异或 |
| & | 按位与 |
| <<，>> | 移位 |
| +，- | 加法与减法 |
| \*，/，% | 乘法、除法与取余 |
| +x，-x | 正负号 |
| ~x | 按位翻转 |
| \*\* | 指数 |
| x.attribute | 属性参考 |
| x[index] | 下标 |
| x[index:index] | 寻址段 |
| f(arguments...) | 函数调用 |
| (experession,...) | 绑定或元组显示 |
| [expression,...] | 列表显示 |
| {key:datum,...} | 字典显示 |
| 'expression,...' | 字符串转换 |