优先级运算类型关联性运算符19圆括号na( … )18成员访问从左到右… . …需计算的成员访问从左到右… [ … ]new (带参数列表)nanew … (

| 优先级 | 运算类型 | 关联性 | 运算符 |
| --- | --- | --- | --- |
| 19 | 圆括号 | n/a | ( … ) |
| 18 | 成员访问 | 从左到右 | … . … |
| | 需计算的成员访问 | 从左到右 | … [ … ] |
| | new (带参数列表) | n/a | new … ( … ) |
| 17 | 函数调用 | 从左到右 | … ( … ) |
| | new (无参数列表) | 从右到左 | new … |
| 16 | 后置递增(运算符在后) | n/a | … ++ |
| | 后置递减(运算符在后) | n/a | … -- |
| 15 | 逻辑非 | 从右到左 | ! … |
| | 按位非 | 从右到左 | ~ … |
| | 一元加法 | 从右到左 | + … |
| | 一元减法 | 从右到左 | - … |
| | 前置递增 | 从右到左 | ++ … |
| | 前置递减 | 从右到左 | -- … |
| | typeof | 从右到左 | typeof … |
| | void | 从右到左 | void … |
| | delete | 从右到左 | delete … |
| 14 | 乘法 | 从左到右 | … * … |
| | 除法 | 从左到右 | … / … |
| | 取模 | 从左到右 | … % … |
| 13 | 加法 | 从左到右 | … + … |
| | 减法 | 从左到右 | … - … |
| 12 | 按位左移 | 从左到右 | … << … |
| | 按位右移 | 从左到右 | … >> … |
| | 无符号右移 | 从左到右 | … >>> … |
| 11 | 小于 | 从左到右 | … < … |
| | 小于等于 | 从左到右 | … <= … |
| | 大于 | 从左到右 | … > … |
| | 大于等于 | 从左到右 | … >= … |
| | in | 从左到右 | … in … |
| | instanceof | 从左到右 | … instanceof … |
| 10 | 等号 | 从左到右 | … == … |
| | 非等号 | 从左到右 | … != … |
| | 全等号 | 从左到右 | … === … |
| | 非全等号 | 从左到右 | … !== … |
| 9 | 按位与 | 从左到右 | … & … |
| 8 | 按位异或 | 从左到右 | … ^ … |

| | | | |
|---|---|---|---|
| 7 | 按位或 | 从左到右 | … \| … |
| 6 | 逻辑与 | 从左到右 | … && … |
| 5 | 逻辑或 | 从左到右 | … \|\| … |
| 4 | 条件运算符 | 从右到左 | … ? … : … |
| 3 | 赋值 | 从右到左 | … = … |
| | | | … += … |
| | | | … -= … |
| | | | … *= … |
| | | | … /= … |
| | | | … %= … |
| | | | … <<= … |
| | | | … >>= … |
| | | | … >>>= … |
| | | | … &= … |
| | | | … ^= … |
| | | | … \|= … |
| 2 | yield | 从右到左 | yield … |
| | yield* | 从右到左 | yield* … |
| 1 | 展开运算符 | n/a | … … |
| 0 | 逗号 | 从左到右 | … , … |

几个简单的例子：

```
new Foo.getName();==============>  new (Foo.getName)();
```

点（.）的优先级高于new操作

```
new Foo().getName();==============>  (new Foo()).getName();
```

括号高于new

```
new new Foo().getName();==============>  new ((new Foo()).getName)();
```

这里确实是(new Foo()).getName()，但是跟括号优先级高于成员访问没关系，实际上这里成员访问的优先级是最高的，因此先执行了 .getName，但是在进行左侧取值的时候， new Foo() 可以理解为两种运算：new 带参数（即 new Foo()）和函数调用（即 先 Foo() 取值之后再 new），而 new 带参数的优先级是高于函数调用的，因此先执行了 new Foo()，或得 Foo 类的实例对象，再进行了成员访问 .getName。