

其他运算符

1. void 运算符

`void` 运算符不返回任何值，或者说返回 `undefined`;

```
void 0; // undefined <=> void(0)
```

主要用于超链接阻止跳转。

```
<a href="https://www.baidu.com" onclick="f(); return false;">点击跳转</a>
<script>
  function f() {
    console.log("a");
  }
</script>
```

上面代码中，点击链接后，会先执行 `onclick` 的代码，由于 `onclick` 返回 `false`，所以浏览器不会跳转到 `https://www.baidu.com`。

`void` 运算符可以取代上面的写法。

```
<a href="javascript: void(f())">点击跳转</a>
```

用户点击链接提交表单，但是不跳转页面。

```
<a href="javascript: void(document.form.submit())">提交</a>
```

2. 逗号运算符

逗号运算符用于对两个表达式求值，并返回后一个表达式的值。

```
"a", "b"; // "b"

let x = 0;
let y = (x++, 10);
x;        // 1
y;        // 10
```

逗号运算符的一个用途是，在返回一个值之前，进行一些辅助操作。

```
let value = (console.log("Hi!"), true); // Hi!  
value; // true
```

3. 圆括号的作用

圆括号 `()` 不是运算符，是一种语法结构。圆括号有两种用途，一种是使用圆括号会改变运算优先级，一种是放在函数名后面调用函数。

圆括号之中只能放置表达式，如果放置语句就会报错。

```
let a = "b";  
(a); // "b"  
(let a = "b"); // Uncaught SyntaxError: Unexpected token "let"
```

4. 运算符左结合或右结合

大多数运算符都是左结合，只有少数是右结合，比如：

- 指数运算符`**`
- 赋值运算符`=`
- 三元条件运算符`? :`

```
3 ** (2 ** 3); // <=> 3 ** 2 ** 3  
x = y = z; // <=> x = ( y = z )  
n = a ? b : c ? d : e; // <=> n = a ? b : (c ? d : e)
```