



链滴

前端 - 可选链和空值合并运算符功能, 你们觉得鸡肋么?

作者: [Rabbitzcc](#)

原文链接: <https://ld246.com/article/1598001338557>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

两种大家比较期待的两种语法新特性已经陆陆续续有浏览器支持了：

- 可选链操作符 (<https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Operators/%E5%8F%AF%E9%80%89%E9%93%BE>)
- 空值合并运算符 (https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Operators/Nullish_coalescing_operator)

空值合并运算符

空值合并操作符 (??) 是一个逻辑操作符，当左侧的操作数为 `null` 或者 `undefined` 时，返回其右侧操作数，否则返回左侧操作数。

```
const foo = null ?? 'default string';
console.log(foo);
// expected output: "default string"
```

```
const baz = 0 ?? 42;
console.log(baz);
// expected output: 0
```

可选链

可选链操作符 (?.) 允许读取位于连接对象链深处的属性的值，而不必明确验证链中的每个引用是否有。?. 操作符的功能类似于 . 链式操作符，不同之处在于，在引用为空 (nullish) (`null` 或者 `undefined`) 的情况下不会引起错误，该表达式短路返回值是 `undefined`。与函数调用一起使用时，如果给定的数不存在，则返回 `undefined`。

```
const adventurer = {
  name: 'Alice',
  cat: {
    name: 'Dinah'
  }
};
```

```
const dogName = adventurer.dog?.name;
console.log(dogName);
// expected output: undefined
```

```
console.log(adventurer.someNonExistentMethod?.());
// expected output: undefined
```

我到时候觉得【合并空值计算】挺有用的，这样能够减少写法

比如将 `var !== undefined || var !== null ? var : s0` 写成 `var ?? s0`

但我觉得可选链有点鸡肋，很容易在代码中写错，可读性实在是太差了，甚至一些报错也不能被捕获。