



链滴

# `null` 和 `undefined` 有什么区别?

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1548478149660>

来源网站: [链滴](#)

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

## 回答

在 JavaScript 中有两个值表示无——`undefined` 和 `null`。他们之间根本的区别在于 `null` 是显式的而 `undefined` 是隐式的。当一个属性不存在或一个变量没有被赋值时，他们的值为 `undefined`。把 `null` 赋给一个变量时，就显式的表明这个变量“无值”。实际上，当明确知道没有对象或不应该有值时使用 `null`，否则就使用 `undefined`。

## 加分回答

- `typeof` 结果不同

```
typeof undefined // "undefined"  
typeof null // "object", 尽管返回 `object`, 但他依旧是一个原始值, 这是 JavaScript 在实现上的  
个 bug。  
typeof NaN // "number"
```

- 等于操作符 `==` 和 `===` 有什么区别

```
undefined == null // true  
undefined === null // false
```

- JavaScript 的设计者 Brendan Eich 认为表示“空值”的不应该是一个对象且 `null` 在计算时会自转换为 0，很不容易发现错误。因此 `undefined` 就诞生了。

- `null` 常用于以下场景：
  - 作为函数的参数，表示该函数的参数不是对象。
  - 作为对象原型链的终点。
- `undefined` 常用于以下场景：
  - 变量被声明了，但没有赋值时，就等于 `undefined`。
  - 调用函数时，应该提供的参数没有提供，该参数等于 `undefined`。
  - 对象没有赋值的属性，该属性的值为 `undefined`。
  - 函数没有返回值时，默认返回 `undefined`。

## 返回总目录

每天 30 秒