



链滴

JavaScript 中如何对比两个对象?

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1547218337208>

来源网站: [链滴](#)

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

2019-01-11

回答

使用 `==` 或 `===` 对两个不同却具有相同属性及属性值的对象进行比较，他们的结果却不会相等。这是因为等号比较的是他们的引用（内存地址），而不是基本类型。

为了测试两个对象在结构上是否相等，需要一个辅助函数。他将遍历每个对象的所有属性，然后测试们是否具有相同的值，嵌套对象也需如此。当然，也可以使用参数来控制是否对原型链进行比较。

注意：此代码只对普通对象、数组、函数、日期和基本类型的数据结构进行对比。

```
function isDeepEqual(obj1, obj2, testPrototypes = false) {
  if (obj1 === obj2) {
    return true
  }

  if (typeof obj1 === "function" && typeof obj2 === "function") {
    return obj1.toString() === obj2.toString()
  }

  if (obj1 instanceof Date && obj2 instanceof Date) {
    return obj1.getTime() === obj2.getTime()
  }

  if (
    Object.prototype.toString.call(obj1) !==
    Object.prototype.toString.call(obj2) ||
    typeof obj1 !== "object"
  ) {
    return false
  }

  const prototypesAreEqual = testPrototypes
    ? isDeepEqual(
        Object.getPrototypeOf(obj1),
        Object.getPrototypeOf(obj2),
        true
      )
    : true

  const obj1Props = Object.getOwnPropertyNames(obj1)
  const obj2Props = Object.getOwnPropertyNames(obj2)

  return (
    obj1Props.length === obj2Props.length &&
    prototypesAreEqual &&
    obj1Props.every(prop => isDeepEqual(obj1[prop], obj2[prop]))
  )
}
```

加分回答

- 像数字和字符串这样的基本类型只需对比他们的值
- 当一个对象赋值给另一个新对象时，使用等号进行对比，他们就会相等。因为他们的引用（内存地址）是同一个。如：

```
const obj1 = {
  name: "Benjamin",
  sex: "male"
};
const obj2 = {
  name: "Benjamin",
  sex: "male"
};
const obj3 = obj1;
console.log(obj1 === obj3); // true
```

- 想要完整的实现对比，可参见 Lodash 中的 [_.isEqual](#) 和 [_.isEqualWith](#) 的实现。

返回总目录

每天 30 秒