

# JavaScript typeof, null, 和 undefined

JavaScript typeof, null, undefined, valueOf()。

## typeof 操作符

你可以使用 typeof 操作符来检测变量的数据类型。

### 实例

```
typeof "John"           // 返回 string
typeof 3.14             // 返回 number
typeof false            // 返回 boolean
typeof [1,2,3,4]        // 返回 object
typeof {name:'John', age:34} // 返回 object
```

[尝试一下 »](#)

在JavaScript中，数组是一种特殊的对象类型。因此 typeof [1,2,3,4] 返回 object。

## null

在 JavaScript 中 null 表示 "什么都没有"。

null是一个只有一个值的特殊类型。表示一个空对象引用。



用 typeof 检测 null 返回是object。

你可以设置为 null 来清空对象：

### 实例

```
var person = null;           // 值为 null(空)，但类型为对象
```

[尝试一下 »](#)

你可以设置为 undefined 来清空对象：

### 实例

```
var person = undefined;      // 值为 undefined，类型为 undefined
```

[尝试一下 »](#)

## undefined

在 JavaScript 中, **undefined** 是一个没有设置值的变量。

**typeof** 一个没有值的变量会返回 **undefined**。

### 实例

```
var person; // 值为 undefined(空), 类型是undefined
```

[尝试一下 »](#)

任何变量都可以通过设置值为 **undefined** 来清空。类型为 **undefined**。

### 实例

```
person = undefined; // 值为 undefined, 类型是undefined
```

[尝试一下 »](#)

## undefined 和 null 的区别

### 实例

null 和 undefined 的值相等, 但类型不等:

```
typeof undefined // undefined
typeof null      // object
null === undefined // false
null == undefined // true
```

[尝试一下 »](#)[← JavaScript 正则表达式](#)[JavaScript 类型转换 →](#)**1 篇笔记**[写笔记](#)

### 1. 定义

- (1) undefined: 是所有没有赋值变量的默认值, 自动赋值。
- (2) null: 主动释放一个变量引用的对象, 表示一个变量不再指向任何对象地址。

### 2. 何时使用null?

当使用完一个比较大的对象时, 需要对其进行释放内存时, 设置为 null。

### 3. null 与 undefined 的异同点是什么呢?

**共同点:** 都是原始类型, 保存在栈中变量本地。

**不同点:**

( 1 ) undefined——表示变量声明过但并未赋过值。

它是所有未赋值变量默认值，例如：

```
var a;    // a 自动被赋值为 undefined
```

( 2 ) null——表示一个变量将来可能指向一个对象。

一般用于主动释放指向对象的引用，例如：

```
var emps = ['ss','nn'];  
emps = null;    // 释放指向数组的引用
```

#### 4、延伸——垃圾回收站

它是专门释放对象内存的一个程序。

- ( 1 ) 在底层，后台伴随当前程序同时运行；引擎会定时自动调用垃圾回收期；
- ( 2 ) 总有一个对象不再被任何变量引用时，才释放。

木子李 4个月前 [11-12]