

数据类型概述

简介

JavaScript 语言的每一个值，都属于某一种数据类型。JavaScript 的数据类型，共有六种。（ES6 又新增了第七种 Symbol 类型的值，本教程不涉及。）

- 数值 (number)：整数和小数（比如`1`和`3.14`）
- 字符串 (string)：文本（比如`Hello World`）。
- 布尔值 (boolean)：表示真伪的两个特殊值，即`true`（真）和`false`（假）
- `undefined`：表示“未定义”或不存在，即由于目前没有定义，所以此处暂时没有任何值
- `null`：表示空值，即此处的值为空。
- 对象 (object)：各种值组成的集合。

通常，数值、字符串、布尔值这三种类型，合称为原始类型 (primitive type) 的值，即它们是最基本的数据类型，不能再细分了。对象则称为合成类型 (complex type) 的值，因为一个对象往往是多个原始类型的值的合成，可以看作是一个存放各种值的容器。至于`undefined`和`null`，一般将它们看成两个特殊值。

对象是最复杂的数据类型，又可以分成三个子类型。

- 狭义的对象 (object)
- 数组 (array)
- 函数 (function)

狭义的对象和数组是两种不同的数据组合方式，除非特别声明，本教程的“对象”都特指狭义的对象。函数其实是处理数据的方法，JavaScript 把它当成一种数据类型，可以赋值给变量，这为编程带来了很大的灵活性，也为 JavaScript 的“函数式编程”奠定了基础。

typeof 运算符

JavaScript 有三种方法，可以确定一个值到底是什么类型。

- `typeof` 运算符
- `instanceof` 运算符
- `Object.prototype.toString` 方法

`instanceof` 运算符和 `Object.prototype.toString` 方法，将在后文介绍。这里介绍 `typeof` 运算符。

`typeof` 运算符可以返回一个值的数据类型。

数值、字符串、布尔值分别返回`number`、`string`、`boolean`。

```
```javascript
typeof 123 // "number"
typeof '123' // "string"
typeof false // "boolean"
```
```

函数返回`function`。

```
```javascript
function f() {}
typeof f
// "function"
```
```

`undefined`返回`undefined`。

```
```javascript
typeof undefined
// "undefined"
```
```

利用这一点，`typeof`可以用来检查一个没有声明的变量，而不报错。

```
```javascript
v
// ReferenceError: v is not defined

typeof v
// "undefined"
```
```

上面代码中，变量`v`没有用`var`命令声明，直接使用就会报错。但是，放在`typeof`后面，就不报错了，而是返回`undefined`。

实际编程中，这个特点通常用在判断语句。

```
```javascript
// 错误的写法
if (v) {
 // ...
}
// ReferenceError: v is not defined

// 正确的写法
if (typeof v === "undefined") {
 // ...
}
```
```

对象返回`object`。

```
```javascript
typeof window // "object"
typeof {} // "object"
typeof [] // "object"
```
```

上面代码中，空数组（`[]`）的类型也是`object`，这表示在 JavaScript 内部，数组本质上只是一种特殊的对象。这里顺便提一下，`instanceof` 运算符可以区分数组 and 对象。`instanceof` 运算符的详细解释，请见《面向对象编程》一章。

```
```javascript
var o = {};
var a = [];

o instanceof Array // false
a instanceof Array // true
```
```

`null` 返回`object`。

```
```javascript
typeof null // "object"
```
```

`null` 的类型是`object`，这是由于历史原因造成的。1995年的 JavaScript 语言第一版，只设计了五种数据类型（对象、整数、浮点数、字符串和布尔值），没考虑`null`，只把它当作`object`的一种特殊值。后来`null`独立出来，作为一种单独的数据类型，为了兼容以前的代码，`typeof null` 返回`object`就没法改变了。

参考链接

- Axel Rauschmayer, [Improving the JavaScript typeof operator](<http://www.2ality.com/2011/11/improving-typeof.html>)