

变量的数据类型

为什么需要数据类型

在计算机中，不同的数据所需占用的存储空间不同，为了充分利用存储空间，于是定义了不同的数据类型。而且，不同的数据类型，寓意也不同。

我们都知道，无论这个变量是字符串类型，还是数字类型，我们都可以直接用 `var` 去定义它。比如：

```
var a = 'hello word';

var b = 123;
```

为什么可以这样做呢？这是因为：JavaScript 是一种「弱类型语言」，或者说是一种「动态语言」，这意味着不需要提前声明变量的类型，在程序运行过程中，类型会自动被确定。

JS 的变量数据类型，是在程序运行的过程中，根据等号右边的值来确定的。而且，变量的数据类型是可以变化的。比如说：

```
var name = 'geektime';

name = 123; // 强制将变量 name 修改为 数字类型
```

JS中一共有六种数据类型

- **基本数据类型（值类型）**：String 字符串、Number 数值、Boolean 布尔值、Null 空值、Undefined 未定义。
- **引用数据类型（引用类型）**：Object 对象。

注意：内置对象 Function、Array、Date、RegExp、Error等都是属于 Object 类型。也就是说，除了那五种基本数据类型之外，其他的，都称之为 Object 类型。

面试问：引用数据类型有几种？

面试答：只有一种，即 Object 类型。

数据类型之间最大的区别：

- 基本数据类型：参数赋值的时候，传数值。
- 引用数据类型：参数赋值的时候，传地址（修改的同一片内存空间）。

一个经典的例子

基本数据类型举例：

```
var a = 23;
var b = a;

a++; //数值加1，地址不变

console.log(a); // 打印结果：24
console.log(b); // 打印结果：23
```

上面的代码中：a 和 b 都是基本数据类型，让 b 等于 a，然后**改变 a 的值之后，发现 b 的值并没有被改变**。

但是在引用数据类型中，就不同了，我们来看一看。

引用数据类型举例：

```
var obj1 = new Object();
obj1.name = 'geektime';

// 让 obj2 等于 obj1
var obj2 = obj1;

// 修改 obj1 的 name 属性
obj1.name = '网络安全';

console.log(obj1.name); // 打印结果：网络安全
console.log(obj2.name); // 打印结果：网络安全
```

上面的代码中：obj1 和 obj2 都是引用数据类型，让 obj2 等于 obj1，然后**修改 obj1.name 的值之后，发现 obj2.name 的值也发生了改变**。

从上面的例子中，可以反映出，基本数据类型和引用数据类型是有区别的。

那到底有什么区别呢？我们进一步往下看。

栈内存和堆内存

我们首先记住一句话：JS中，所有的**变量**都是保存在**栈内存**中的。

然后来看看下面的区别。

基本数据类型：

基本数据类型的值，直接保存在栈内存中。值与值之间是独立存在，修改一个变量不会影响其他的变量。

引用数据类型：

对象是保存到**堆内存**中的。

每创建一个新的对象，就会在堆内存中开辟出一个新的空间；而**变量保存了对象的内存地址**（对象的引用），保存在栈内存当中。如果两个变量保存了同一个对象的引用，当一个变量修改属性时，另一个变量也会受到影响。