

字面量：数字和字符串

“字面量”即**常量**，是固定值，不可改变。看见什么，它就是什么。

字面量有3种：

- 数字
- 字符串
- 布尔字面量

(1) 数值的字面量非常简单，写上去就行，不需要任何的符号。例如：

```
alert(886); //886是数字，所以不需要加引号。
```

(2) 字符串的字面量也很简单，但一定要加上引号，可以是单词、句子等。例如：

```
console.log('886');  
  
console.log('极客时间');
```

注意：100是数字，'100'是字符串。

(3) 布尔字面量举例：

```
if (true) {  
    console.log('如果为真，就走这里的代码');  
}
```

总结

常量都可以直接使用，但是我们一般不会直接使用常量。

如果直接使用常量的话，比较麻烦。比如说，多个地方要用到同一个常量，还不如事先定义一个变量，用来保存常量。

变量更方便我们使用，所以在开发中都是通过变量去保存一个字面量，而不会直接使用字面量。

变量的概念

变量：是用于存放数据的容器。我们通过「变量名」获取数据，甚至可以修改数据。

变量还可以用来保存字面量。

本质：变量是程序在内存中申请的一块用来存放数据的空间。

变量的声明和赋值

变量的声明/定义

在ES6语法之前，统一使用 `var` 关键字来声明一个变量。比如：

```
var name; // 声明一个名为 name 的变量
```

PS：在JavaScript中，永远都是用var来定义变量（在 ES6 之前），这和C、Java等语言不同。

变量的声明/定义（ES6）

在ES6语法及之后，可以使用 `const`、`let` 关键字来声明一个变量

```
const name; // 声明一个名为 name 的常量
```

```
let age; // 声明一个名为 age 的变量
```

如果你想定义一个常量，就用 `const`；如果你想定义一个变量，就用 `let`。

变量的赋值

变量的赋值：

```
name = '极客时间'
```

变量的声明和赋值，写在一起，举例如下：

```
var a = 100; // ES5语法
```

```
const b = 'hello'; // ES6 语法
```

```
let b = 'world'; // ES6 语法
```

如下图所示：



`var`是英语“variant”变量的缩写。后面要加一个空格，空格后面的东西就是“变量名”：

- 定义变量：`var`就是一个**关键字**，用来定义变量。所谓关键字，就是有特殊功能的单词。
- 变量的赋值：等号表示**赋值**，将等号右边的值，赋给左边的变量。
- 变量名：我们可以给变量任意的取名字。

变量的初始化

有经验的程序员，会把声明和赋值写在一起：

```
var a = 100;    //声明，并且赋值100
console.log(a); //输出100
```

声明一个变量并赋值，我们称之为**变量的初始化**。

修改变量的值

一个变量被重新复赋值后，它原有的值就会被覆盖，变量值将以最后一次赋的值为准。

举例：

```
var a = 100;
a = 110;

console.log(a); // 打印结果：110。因为 110 覆盖了 100
```

同时声明多个变量

同时声明多个变量时，只需要写一个 var，多个变量名之间用英文逗号隔开。

举例：

```
var name = '极客时间', age = 27, number = 100;
```

变量声明的几种特殊情况

变量建议先声明，再使用。否则可能会报错。具体如下。

写法1、先声明，再赋值：（正常）

```
var a;
a = 100;
console.log(a); // 打印结果：100
```

写法2、不声明，只赋值：（正常）

```
a = 100;
console.log(a); // 打印结果：100
```

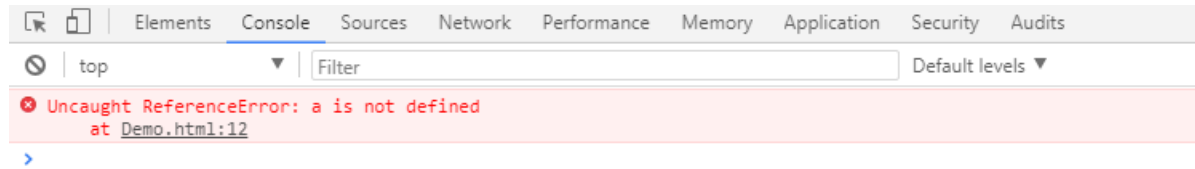
写法3、只声明，不赋值：（注意，打印 undefined）

```
var a;
console.log(a); // 打印结果：undefined
```

写法4、不声明，不赋值，直接使用：（会报错）

```
console.log(a); // 会报错
```

控制台会报错：



补充：写法1和写法2虽然都正常，但这两种写法是有区别的，写法2是程序进行了容错处理。大多数情况下，我们都是采用的写法1。

变量的命名规范

大写字母是可以使用的，并且大小写敏感。也就是说A和a是两个变量。

```
var A = 250;    //变量1
var a = 888;    //变量2
```

我们来整理一下**变量的命名规范**：

- 只能由字母(A-Z、a-z)、数字(0-9)、下划线(_)、美元符(\$)组成
- 不能以数字开头。也就是说，必须以字母(A-Z、a-z)、下划线(_)或者美元符(\$)开头。变量名中不允许出现空格。
- 不用使用JS语言中保留的「关键字」和「保留字」作为变量名。
- 建议用驼峰命名规则。比如getElementById、matherAndFather、aaaOrBbbAndCcc
- 变量名会区分大小写(javascript是区分大小写的语言)。
- 变量名长度不能超过255个字符。
- 汉语可以作为变量名。但是不建议使用，因为low。

标识符、关键字、保留字

标识符

标识符：在JS中所有的可以由我们**自主命名**的都可以称之为标识符。

例如：变量名、函数名、属性名、参数名都是属于标识符。通俗来讲，标识符就是我们写代码时为它们起的名字。

标识符的命名规则和变量的命名规则是一样的。看上面一段就可以了。

同样，标识符不能使用语言中保留的**关键字**及**保留字**。如下。

关键字

关键字：是指JS本身已经使用了的单词，具备固定的语义，我们不能再用它们充当变量、函数名等标识符。

JS中的关键字如下：

```
break、continue、case、default、  
  
if、else、switch、for、in、do、while、  
  
try、catch、finally、throw、  
  
var、void、function、return、new、  
  
this、typeof、instanceof、delete、with、  
  
true、false、null、undefined
```

保留字

保留字：实际上就是预留的“关键字”。意思是现在虽然还不是关键字，但是未来可能会成为关键字，同样不能使用它们当充当变量名、函数名等标识符。

JS 中的保留字如下：

```
abstract、boolean、byte、char、class、const、  
  
debugger、double、enum、export、extends、final、float、goto  
  
implements、import、int、interface、long、native、package、  
  
private、protected、public、short、static、super、synchronized、throws、  
  
transient、volatile
```

变量值的交换

举例：交换两个变量的值

代码实现：

```
var a1 = 100;  
var a2 = 200;  
  
var temp;  
  
temp = a1;  
a1 = a2;  
a2 = temp;  
  
console.log(a1);  
console.log(a2);
```