

JavaScript的注释和变量

注释

怎么用？

单行注释

多行注释

变量

就像代数那样

JavaScript 变量

JavaScript 数据类型

声明（创建）JavaScript 变量

一条语句，多个变量

Value = undefined

重新声明 JavaScript 变量

JavaScript 算数

注释

JavaScript的注释和C++啊、Java、CSharp的注释都一样，其实也没必要每个语言区分的那么明确，意思是就行，能达到目的才是最好的。

怎么用？

注释到底怎么用呢

单行注释

```
1 | //我是注释
```

单行注释只能注释一行，用//符号表示，在//后面的任何语句都不会被执行，那么如果做好几行的注释怎么办，总不能每一行都用//开头吧？

多行注释

```
1  /* 我是多行注释  */
2  /*
3  我是多行注释
4  可以注释多行
5
6
7  */
```

这很容易明白，就是用`/*`符号开头，以`*/`符号结尾的注释，`/**/`之间的语句不会执行，可以写好几行。

注释就说完了

变量

变量是用于存储信息的"容器"。

实例

```
1  var x=5;
2  var y=6;
3  var z=x+y;
```

就像代数那样

$x=5$

$y=6$

$z=x+y$

在代数中，我们使用字母（比如 x ）来保存值（比如 5）。

通过上面的表达式 $z=x+y$ ，我们能够计算出 z 的值为 11。

在 JavaScript 中，这些字母被称为变量。



您可以把变量看做存储数据的容器。

JavaScript 变量

与代数一样，JavaScript 变量可用于存放值（比如 $x=5$ ）和表达式（比如 $z=x+y$ ）。

变量可以使用短名称（比如 x 和 y ），也可以使用描述性更好的名称（比如 `age`, `sum`, `totalvolume`）。

- 变量必须以字母开头
- 变量也能以 \$ 和 _ 符号开头（不过我们不推荐这么做）
- 变量名称对大小写敏感（y 和 Y 是不同的变量）



JavaScript 语句和 JavaScript 变量都对大小写敏感。

JavaScript 数据类型

JavaScript 变量还能保存其他数据类型，比如文本值 (name="Bill Gates")。

在 JavaScript 中，类似 "Bill Gates" 这样一条文本被称为字符串。

JavaScript 变量有很多种类型，但是现在，我们只关注数字和字符串。

当您向变量分配文本值时，应该用双引号或单引号包围这个值。

当您向变量赋的值是数值时，不要使用引号。如果您用引号包围数值，该值会被作为文本来处理。

实例

```
1 var pi=3.14;  
2 var person="John Doe";  
3 var answer='Yes I am!';
```

声明（创建） JavaScript 变量

在 JavaScript 中创建变量通常称为"声明"变量。

我们使用 var 关键词来声明变量：

```
1 var carname;
```

变量声明之后，该变量是空的（它没有值）。

如需向变量赋值，请使用等号：

```
1 carname="Volvo";
```

不过，您也可以在声明变量时对其赋值：

```
1 var carname="Volvo";
```

在下面的例子中，我们创建了名为 carname 的变量，并向其赋值 "Volvo"，然后把它放入 id="demo" 的 HTML 段落中：

```
1 | var carname="Volvo";  
2 | document.getElementById("demo").innerHTML=carname;
```



一个好的编程习惯是，在代码开始处，统一对需要的变量进行声明。

一条语句，多个变量

您可以在一条语句中声明很多变量。该语句以 **var** 开头，并使用逗号分隔变量即可：

```
1 | var lastname="Doe", age=30, job="carpenter";
```

声明也可横跨多行：

```
1 | var lastname="Doe",  
2 |   age=30,  
3 |   job="carpenter";
```

一条语句中声明的多个不可以赋同一个值：

```
1 | var x,y,z=1;
```

x,y 为 undefined，z 为 1。

Value = undefined

在计算机程序中，经常会声明无值的变量。未使用值来声明的变量，其值实际上是 **undefined**。

在执行过以下语句后，变量 **carname** 的值将是 **undefined**：

```
1 | var carname;
```

重新声明 JavaScript 变量

如果重新声明 JavaScript 变量，该变量的值不会丢失：

在以下两条语句执行后，变量 `carname` 的值依然是 "Volvo"：

```
1 | var carname="Volvo";  
2 | var carname;
```

JavaScript 算数

您可以通过 JavaScript 变量来做算数，使用的是 `=` 和 `+` 这类运算符：

```
y=5; x=y+2;
```