

第 14 章 内置对象

学习要点:

- 1.Global 对象
- 2.Math 对象

主讲教师: 李炎恢

合作网站: <http://www.ibeifeng.com>

讲师博客: <http://hi.baidu.com/李炎恢>

ECMA-262 对内置对象的定义是: “由 ECMAScript 实现提供的、不依赖宿主环境的对象, 这些对象在 ECMAScript 程序执行之前就已经存在了。” 意思就是说, 开发人员不必显式地实例化内置对象; 因为它们已经实例化了。ECMA-262 只定义了两个内置对象: Global 和 Math。

一. Global 对象

Global(全局)对象是 ECMAScript 中一个特别的对象, 因为这个对象是不存在的。在 ECMAScript 中不属于任何其他对象的属性和方法, 都属于它的属性和方法。所以, 事实上, 并不存在全局变量和全局函数; 所有在全局作用域定义的变量和函数, 都是 Global 对象的属性和方法。

PS: 因为 ECMAScript 没有定义怎么调用 Global 对象, 所以, Global.属性或者 Global.方法()都是无效的。(Web 浏览器将 ~~Global 作为 window 对象的一部分加以实现~~)

将 window 作为 Global 并实现 Global 的一部分功能

Global 对象有一些内置的属性和方法:

1.URI 编码方法

URI 编码可以对链接进行编码, 以便发送给浏览器。它们采用特殊的 UTF-8 编码替换所有无效字符, 从而让浏览器能够接受和理解。

encodeURIComponent()不会对本身属于 URI 的特殊字符进行编码, 例如冒号、正斜杠、问号和#号; 而 encodeURIComponent()则会对它发现的任何非标准字符进行编码

```
var box = 'Lee 李';  
alert(encodeURIComponent(box));           //只编码了中文
```

```
var box = 'Lee 李';  
alert(encodeURIComponent(box));           //特殊字符和中文编码了
```

PS: 因为 encodeURIComponent()编码比 encodeURIComponent()编码来的更加彻底, 一般来说 encodeURIComponent()使用频率要高一些。

使用了 URI 编码过后, 还可以进行解码, 通过 `decodeURI()` 和 `decodeURIComponent()` 来进行解码

```
var box = 'Lee 李';  
alert(decodeURI(encodeURIComponent(box)));
```

//还原

decodeURIComponent 可以解码 encodeURIComponent 编码字符

```
var box = '//Lee 李';  
alert(decodeURIComponent(encodeURIComponent(box)));    //还原
```

PS: URI 方法如上所述的四种, 用于代替已经被 ECMA-262 第 3 版废弃的 `escape()` 和 `unescape()` 方法。URI 方法能够编码所有的 Unicode 字符, 而原来的只能正确地编码 ASCII 字符。所以建议不要再使用 `escape()` 和 `unescape()` 方法。

2. `eval()` 方法 类似PHP

`eval()` 方法主要担当一个字符串解析器的作用, 他只接受一个参数, 而这个参数就是要执行的 JavaScript 代码的字符串。

```
eval('var box = 100');    //解析了字符串代码  
alert(box);  
eval('alert(100)');    //同上
```

typeof 表达式返回
的都是字符串

```
eval('function box() {return 123}');    //函数也可以  
alert(box());
```

`eval()` 方法的功能非常强大, 但也非常危险。因此使用的时候必须极为谨慎。特别是在用户输入数据的情况下, 非常有可能导致程序的安全性, 比如代码注入等等。

3. Global 对象属性

Global 对象包含了一些属性: undefined、NaN、Object、Array、Function 等等。

```
alert(Array);    //返回构造函数
```

4. window 对象

之前已经说明, Global 没有办法直接访问, 而 Web 浏览器可以使用 window 对象来实现一全局访问。

```
alert(window.Array);    //同上
```

python中是math

二. Math 对象

ECMAScript 还为保存数学公式和信息提供了一个对象, 即 Math 对象。与我们在 JavaScript 直接编写计算功能相比, Math 对象提供的计算功能执行起来要快得多。

1. Math 对象的属性

Math 对象包含的属性大都是数学计算中可能会用到的一些特殊值。

属 性	说 明
Math.E	自然对数的底数, 即常量 e 的值
Math.LN10	10 的自然对数
Math.LN2	2 的自然对数

Math.LOG2E	以 2 为底 e 的对数
Math.LOG10E	以 10 为底 e 的对数
Math.PI	π 的值
Math.SQRT1_2	1/2 的平方根
Math.SQRT2	2 的平方根

```

alert(Math.E);                //
alert(Math.LN10);
alert(Math.LN2);
alert(Math.LOG2E);
alert(Math.LOG10E);
alert(Math.PI);
alert(Math.SQRT1_2);
alert(Math.SQRT2);           //

```

2.min()和 max()方法

Math.min()用于确定一组数值中的最小值。Math.max()用于确定一组数值中的最大值。

```

alert(Math.min(2,4,3,6,3,8,0,1,3));    //最小值
alert(Math.max(4,7,8,3,1,9,6,0,3,2));  //最大值

```

3.舍入方法

Math.ceil()执行向上舍入，即它总是将数值向上舍入为最接近的整数；
 Math.floor()执行向下舍入，即它总是将数值向下舍入为最接近的整数；
 Math.round()执行标准舍入，即它总是将数值四舍五入为最接近的整数；

```

alert(Math.ceil(25.9));           //26
alert(Math.ceil(25.5));           //26
alert(Math.ceil(25.1));           //26

alert(Math.floor(25.9));           //25
alert(Math.floor(25.5));           //25
alert(Math.floor(25.1));           //25

alert(Math.round(25.9));           //26
alert(Math.round(25.5));           //26
alert(Math.round(25.1));           //25

```

4.random()方法

Math.random()方法返回介于 0 到 1 之间一个随机数，不包括 0 和 1。如果想大于这个范围的话，可以套用一下公式：

值 = Math.floor(Math.random() * 总数 + 第一个值)

```

alert(Math.floor(Math.random() * 10 + 1));    //随机产生 1-10 之间的任意数

```

```
for (var i = 0; i < 10; i++) {
    document.write(Math.floor(Math.random() * 10 + 5));    //5-14 之间的任意数
    document.write('<br />');
}
```

为了更加方便的传递想要范围，可以写成函数：

```
function selectFrom(lower, upper) {
    var sum = upper - lower + 1;                //总数-第一个数+1
    return Math.floor(Math.random() * sum + lower);
}
```

```
for (var i = 0; i < 10; i++) {
    document.write(selectFrom(5, 10));          //直接传递范围即可
    document.write('<br />');
}
```

5. 其他方法

方 法	说 明
Math.abs(num)	返回 num 的绝对值
Math.exp(num)	返回 Math.E 的 num 次幂
Math.log(num)	返回 num 的自然对数
Math.pow(num, power)	返回 num 的 power 次幂
Math.sqrt(num)	返回 num 的平方根
Math.acos(x)	返回 x 的反余弦值
Math.asin(x)	返回 x 的正弦值
Math.atan(x)	返回 x 的反正切值
Math.atan2(y, x)	返回 y/x 的反正切值
Math.cos(x)	返回 x 的余弦值
Math.sin(x)	返回 x 的正弦值
Math.tan(x)	返回 x 的正切值

感谢收看本次教程！

本课程是由北风网(ibeifeng.com)

瓢城 **Web** 俱乐部(yc60.com)联合提供：

本次主讲老师：李炎恢

我的博客：hi.baidu.com/李炎恢/

我的邮件：yc60.com@gmail.com