

Math对象

ECMAScript 还为保存数学公式和信息提供了一个公共位置，即 `Math` 对象。与我们在 JavaScript 直接编写的计算功能相比，`Math` 对象提供的计算功能执行起来要快得多。`Math` 对象中还提供了辅助完成这些计算的属性和方法。

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。下表列出了这些属性。

属 性	说 明
<code>Math.E</code>	自然对数的底数，即常量e的值
<code>Math.LN10</code>	10的自然对数
<code>Math.LN2</code>	2的自然对数
<code>Math.LOG2E</code>	以2为底e的对数
<code>Math.LOG10E</code>	以10为底e的对数
<code>Math.PI</code>	π的值
<code>Math.SQRT1_2</code>	1/2的平方根（即2的平方根的倒数）
<code>Math.SQRT2</code>	2的平方根

虽然讨论这些值的含义和用途超出了本书范围，但你确实可以随时使用它们。

2. min()和max()方法

min() 和 max()

其中，`min()` 和 `max()` 方法用于确定一组数值中的最小值和最大值。这两个方法都可以接收任意多个数值参数，如下面的例子所示。

图灵社区会员 StinkBC(StinkBC@gmail.com) 专享 尊重版权



```
var max = Math.max(3, 54, 32, 16);
alert(max);    //54

var min = Math.min(3, 54, 32, 16);
alert(min);    //3
```

要找到数组中的最大或最小值，可以像下面这样使用 `apply()` 方法。

```
var values = [1, 2, 3, 4, 5, 6, 7, 8];
var max = Math.max.apply(Math, values);
```

这个技巧的关键是把 `Math` 对象作为 `apply()` 的第一个参数，从而正确地设置 `this` 值。然后，可以将任何数组作为第二个参数。

3. 舍入方法

`Math.ceil()`、`Math.floor()` 和 `Math.round()`

下面来介绍将小数值舍入为整数的几个方法：`Math.ceil()`、`Math.floor()` 和 `Math.round()`。这三个方法分别遵循下列舍入规则：

- ❑ `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（这也是我们在数学课上学到的舍入规则）。

下面是使用这些方法的示例：



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

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

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

4. `random()` 方法

`Math.random()` 方法返回大于等于 0 小于 1 的一个随机数。对于某些站点来说，这个方法非常实用，因为可以利用它来随机显示一些名人名言和新闻事件。套用下面的公式，就可以利用 `Math.random()` 从某个整数范围内随机选择一个值。

值 = `Math.floor(Math.random() * 可能值的总数 + 第一个可能的值)`

公式中用到了 `Math.floor()` 方法，这是因为 `Math.random()` 总返回一个小数值。而用这个小数值乘以一个整数，然后再加上一个整数，最终结果仍然还是一个小数。举例来说，如果你想选择一个 1 到 10 之间的数值，可以像下面这样编写代码：

```
var num = Math.floor(Math.random() * 10 + 1);
```

总共有 10 个可能的值（1 到 10），而第一个可能的值是 1。而如果想要选择一个介于 2 到 10 之间的值，就应该将上面的代码改成这样：

```
var num = Math.floor(Math.random() * 9 + 2);
```

从 2 数到 10 要数 9 个数，因此可能值的总数就是 9，而第一个可能的值就是 2。多数情况下，其实都可以通过一个函数来计算可能值的总数和第一个可能的值，例如：

```
function selectFrom(lowerValue, upperValue) {
    var choices = upperValue - lowerValue + 1;
    return Math.floor(Math.random() * choices + lowerValue);
}

var num = selectFrom(2, 10);
alert(num); // 介于 2 和 10 之间（包括 2 和 10）的一个数值
```

函数 `selectFrom()` 接受两个参数：应该返回的最小值和最大值。而用最大值减最小值再加 1 得到了可能值的总数，然后它又把这些数值套用到了前面的公式中。这样，通过调用 `selectFrom(2,10)` 就可以得到一个介于 2 和 10 之间（包括 2 和 10）的数值了。利用这个函数，可以方便地从数组中随机取出一项，例如：

```
var colors = ["red", "green", "blue", "yellow", "black", "purple", "brown"];
var color = colors[selectFrom(0, colors.length-1)];
alert(color); // 可能是数组中包含的任何一个字符串
```

5. 其他方法

`Math` 对象中还包含其他一些与完成各种简单或复杂计算有关的方法，但详细讨论其中每一个方法的细节及适用情形超出了本书的范围。下面我们就给出一个表格，其中列出了这些没有介绍到的 `Math` 对象的方法。

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