

# 跟志佳学Web前端 冲击月薪30000+

Baidu 百度

Tencent 腾讯

阿里巴巴

珠峰培训  
前端专家级课程



你不会React还算前端吗？！

React Native App / React架构课程



[首页](#) » [博客](#) » [Airen的博客](#)

## 12个JavaScript技巧

作者：大漠 日期：2016-05-08 点击：8687

JavaScript JavaScript小技巧

妙味课堂 超值年费会员VIP



汇集国内最全  
WEB前端开发视频

在这篇文章中将给大家分享12个有关于JavaScript的小技巧。这些小技巧可能在你的实际工作中或许能帮助你解决一些问题。

### 使用!!操作符转换布尔值

有时候我们需要对一个变量查检其是否存在或者检查值是否有一个有效值，如果存在就返回 `true` 值。为了做这样的验证，我们可以使用 `!!` 操作符来实现是非常的方便与简单。对于变量可以使



学前端开发  
选珠峰培训



JS HTML5 Angular Node  
React Es6 Webpack

大漠前端网络班

1980跟大漠老师学前端  
仅限本月

关注我们

用 `!!variable` 做检测，只要变量的值为: `0`、`null`、`" "`、`undefined` 或者 `NaN` 都将返回的是 `false`，反之返回的是 `true`。比如下面的示例：

```
function Account(cash) {
    this.cash = cash;
    this.hasMoney = !!cash;
}

var account = new Account(100.50);
console.log(account.cash); // 100.50
console.log(account.hasMoney); // true

var emptyAccount = new Account(0);
console.log(emptyAccount.cash); // 0
console.log(emptyAccount.hasMoney); // false
```

在这个示例中，只要 `account.cash` 的值大于 `0`，那么 `account.hasMoney` 返回的值就是 `true`。

## 使用+将字符串转换成数字

这个技巧非常有用，其非常简单，可以交字符串数据转换成数字，不过其只适合用于字符串数据，否则将返回 `NaN`，比如下面的示例：

```
function toNumber(strNumber) {
    return +strNumber;
}

console.log(toNumber("1234")); // 1234
console.log(toNumber("ACB")); // NaN
```

这个也适用于 `Date`，在本例中，它将返回的是时间戳数字：



## 合作网站

- JavaScript学习案例
- 墨鱼前端培训
- HTML5梦工场
- Sass入门指南
- CSS解决方案
- W3ctech
- 前端圈
- Drupal中国

## 友情链接

- 广州微信开发
- 在线图片压缩
- java源代码学习
- 墨鱼前端开发培训
- 猪八戒网
- HTML5梦工场
- PHP教程

```
console.log(+new Date()) // 1461288164385
```

## 并条件符

如果你有一段这样的代码：

```
if (conected) {  
    login();  
}
```

你也可以将变量简写，并且使用 `&&` 和函数连接在一起，比如上面的示例，可以简写成这样：

```
conected && login();
```

如果一些属性或函数存在于一个对象中，你也可以这样做检测，如下面的代码所示：

```
user && user.login();
```

## 使用 || 运算符

在ES6中有默认参数这一特性。为了在老版本的浏览器中模拟这一特性，可以使用 `||` 操作符，并且将默认值当做第二个参数传入。如果第一个参数返回的值为 `false`，那么第二个值将会认为是一个默认值。如下面这个示例：

```
function User(name, age) {  
    this.name = name || "Oliver Queen";  
    this.age = age || 27;  
}
```

- 程序员客栈
- imweb 前端社区
- 前端笔记

```
var user1 = new User();  
console.log(user1.name); // Oliver Queen  
console.log(user1.age); // 27  
  
var user2 = new User("Barry Allen", 25);  
console.log(user2.name); // Barry Allen  
console.log(user2.age); // 25
```

## 在循环中缓存array.length

这个技巧很简单，这个在处理一个很大的数组循环时，对性能影响将是非常大的。基本上，大家都会写一个这样的同步迭代的数组：

```
for(var i = 0; i < array.length; i++) {  
    console.log(array[i]);  
}
```

如果是一个小型数组，这样做很好，如果你要处理的是一个大的数组，这段代码在每次迭代都将会重新计算数组的大小，这将会导致一些延误。为了避免这种现象出现，可以将 `array.length` 做一个缓存：

```
var length = array.length;  
for(var i = 0; i < length; i++) {  
    console.log(array[i]);  
}
```

你也可以写成这样：

```
for(var i = 0, length = array.length; i < length; i++) {  
    console.log(array[i]);  
}
```

## 检测对象中属性

---

当你需要检测一些属性是否存在，避免运行未定义的函数或属性时，这个小技巧就显得很有用。如果你打算定些一些跨兼容的浏览器代码，你也可能会用到这个小技巧。例如，你想使用 `document.querySelector()` 来选择一个 `id`，并且让它能兼容IE6浏览器，但是在IE6浏览器中这个函数是不存在的，那么使用这个操作符来检测这个函数是否存在就显得非常的有用，如下面的示例：

```
if ('querySelector' in document) {  
    document.querySelector("#id");  
} else {  
    document.getElementById("id");  
}
```

在这个示例中，如果 `document` 不存在 `querySelector` 函数，那么就会调用 `document.getElementById("id")`。

## 获取数组中最后一个元素

---

`Array.prototype.slice(begin,end)` 用来获取 `begin` 和 `end` 之间的数组元素。如果你不设置 `end` 参数，将会将数组的默认长度值当作 `end` 值。但有些同学可能不知道这个函数还可以接受负值作为参数。如果你设置一个负值作为 `begin` 的值，那么你可以获取数组的最后一个元素。如：

```
var array = [1,2,3,4,5,6];  
console.log(array.slice(-1)); // [6]  
console.log(array.slice(-2)); // [5, 6]  
console.log(array.slice(-3)); // [4, 5, 6]
```

## 数组截断

---

这个小技巧主要用来锁定数组的大小，如果用于删除数组中的一些元素来说，是非常有用的。例如，你的数组有 10 个元素，但你只想只要前五个元素，那么你可以通过 `array.length=5` 来截断数组。如下面这个示例：

```
var array = [1, 2, 3, 4, 5, 6];
console.log(array.length); // 6
array.length = 3;
console.log(array.length); // 3
console.log(array); // [1, 2, 3]
```

## 替换所有

---

`String.replace()` 函数允许你使用字符串或正则表达式来替换字符串，本身这个函数只替换第一次出现的字符串，不过你可以使用正则表达多中的 `/g` 来模拟 `replaceAll()` 函数功能：

```
var string = "john john";
console.log(string.replace(/hn/, "ana")); // "joana john"
console.log(string.replace(/hn/g, "ana")); // "joana joana"
```

## 合并数组

---

如果你要合并两个数组，一般情况之下你都会使用 `Array.concat()` 函数：

```
var array1 = [1, 2, 3];
var array2 = [4, 5, 6];
console.log(array1.concat(array2)); // [1, 2, 3, 4, 5, 6];
```

然后这个函数并不适合用来合并两个大型的数组，因为其将消耗大量的内存来存储新创建的数组。在这种情况下，可以使用 `Array.prototype.push.apply(arr1, arr2)` 来替代创建一个新数组。这种方法不是用来创建一个新的数组，其只是将第一个第二个数组合并在一起，同时减少内存的使用：

```
var array1 = [1, 2, 3];
var array2 = [4, 5, 6];
console.log(array1.push.apply(array1, array2)); // [1, 2, 3, 4, 5, 6];
```

## 将NodeList转换成数组

如果你运行 `document.querySelectorAll("p")` 函数时，它可能返回DOM元素的数组，也就是 `NodeList` 对象。但这个对象不具有数组的函数功能，比如 `sort()`、`reduce()`、`map()`、`filter()` 等。为了让这些原生的数组函数功能也能用于其上面，需要将节点列表转换成数组。可以使用  `[].slice.call(elements)` 来实现：

```
var elements = document.querySelectorAll("p"); // NodeList
var arrayElements = [].slice.call(elements); // Now the NodeList is an array
var arrayElements = Array.from(elements); // This is another way of converting NodeList to Array
```

## 数组元素的洗牌

对于数组元素的洗牌，不需要使用任何外部的库，比如Lodash，只要这样做：

```
var list = [1, 2, 3];
console.log(list.sort(function() { Math.random() - 0.5 })); // [2, 1, 3]
```

## 总结

现在你学会了些有用的JavaScript小技巧。希望这些小技巧能在工作中帮助你解决一些麻烦，或者说这篇文章对你有所帮助。如果你有一些优秀的JavaScript小技巧，欢迎在评论中与我们一起分享。

本文根据@Caio Ribeiro Pereira的《12 Extremely Useful Hacks for JavaScript》所译，整个译文带有我们自己的理解与思想，如果译得不好或有不对之处还请同行朋友指点。如需转载此译文，需注明英文出处：<https://blog.jscrambler.com/12-extremely-useful-hacks-for-javascript/>。



## 大漠

常用昵称“大漠”，W3CPlus创始人，目前就职于手淘。对HTML5、CSS3和Sass等前端脚本语言有非常深入的认识和丰富的实践经验，尤其专注对CSS3的研究，是国内最早研究和使用的CSS3技术的一批人。CSS3、Sass和Drupal中国布道者。2014年出版《[图解CSS3：核心技术与案例实战](#)》。

如需转载，烦请注明出处：<http://www.w3cplus.com/javascript/12-extremely-useful-hacks-for-javascript.html>

上一篇: [使用CSS制作Heart动画](#) | 下一篇: [JavaScript算法练习:关于字符串中每个单词的首字母大写化问题](#)

[CSS3](#)[Mobile](#)[Sass](#)[JavaScript](#)[视频教程](#)[译文](#)[SVG](#)[工具](#)[标签云](#)



## 关于我们



W3cplus是一个致力于推广国内前端行业的技术博客。它以探索为己任，不断活跃在行业技术最前沿，努力提供高质量前端技术博文；其文章范围广泛，主要以CSS3、HTML5、Sass、Mobile和各类DEMO为主。

W3cplus具有一支强大的团队，提供长期的前端项目外包，Drupal建站，Drupal主题制作服务，以及提供企业广告展示与招聘发布，有需要的请联系：

QQ:81059347，E-mail:w3cplus@hotmail.com

## 关于站长



常用昵称“大漠”，W3CPlus创始人，目前就职于淘宝。中国Drupal社区核心成员之一。对HTML5、CSS3和Sass等前端脚本语言有非常深入的认识和丰富的实践经验，尤其专注对CSS3、Sass和Mobile的研究，是国内最早研究和使用的CSS3和Sass技术的一批人。CSS3、Sass和Drupal中国布道者。2014年出版《图解CSS3：核心技术与案例实战》。



## 我的作品



本书是国内著名的Web前端专家历时2载的心血之作，根据最新的CSS3撰写，融入了作者在CSS领域多年的使用经验，旨在将本书打造成为CSS3领域最权威和实用的专业著作，供没有经验的读者系统学习，供有经验的读者参考备查。

本书理论知识系统全面，详细讲解了选择器、伸缩布局盒模型、渐变、过渡、动画等主题下涵盖的所有CSS3新特性。

湘ICP备13003850号-12，版权所有 衡阳瑞思信息技术有限公司 © 2011-2016 W3CPLUS，感谢Drupal开源技术。感谢七牛云存储提供静态资源空间。



CSS3

Mobile

Sass

JavaScript

视频教程

译文

SVG

工具

标签云