

# 看，for..in和for..of在那里吵架！

这篇文章发布于 2018年08月6日，星期一，23:57，归类于 [JS API](#)。阅读 14087 次, 今日 24 次 [60 条评论](#)

by zhangxinxu from <https://www.zhangxinxu.com/wordpress/?p=7908>

本文可全文转载，但需得到原作者书面许可，同时保留原作者和出处，摘要引流则随意。

本故事纯属虚构，如有雷同，恭喜你，人品不错。本文6000多字，阅读约15分钟。

今天可是个好日子！

你问我为什么？

你这都不知道，ChinaJoy啊！可是身为瘦宅的我的大日子啊，不说了，我赶时间，我可是偷偷瞒着我老婆出来的，只能出来2小时。

哇塞，好热闹啊！琳琅满目的展位，熙熙攘攘的人群，喔哦哦，一排排的showgirl都长得都好好看，虽然和我老婆比还差点。



突然，旁边一个戴眼镜的小胖子一声大叫：“看，for..in和for..of在那里吵架！”

我顺着望去，只见2点钟方向有一群穿着格子衫背着电脑包的人在围观什么，直觉告诉我，此事不简单，于是就走过去一看究竟。

一打听，原来是两个人在争论谁给“天上地下，唯我独萌”的血小板拍照效果好。



不过这两人长得好像啊，是亲兄弟吧，一时间分不清谁是谁，等下，两人头上帽子上的文字好像不一样，一个是“淫”，一个是“污”。我摸了摸我性感的下巴沉思了下，哦，立马豁然开朗。“淫”指的是 `for..in`，`in` 读音和“淫”近似；“污”指的是 `for..of`，`of` 读音和“污”近似。哈哈哈，我可真是“淫才”，不过听起来好像不太和谐，我还是用IN和OF代替他们吧。

IN说：“我第一届ChinaJoy就过来拍照了，整个圈子谁不认识我！”众人点头。

OF说：“大叔，现在都什么年代了，细胞都谈恋爱了，拍照这种事不是越老就越厉害的！”

IN说：“我可以360度无死角拍美照！”

OF说：“好巧啊，我也可以360度无死角拍美照！”

.....

就这样，IN和OF你一句我一句，吵得不可开交，完全没有个头，我实在看不下去了，直接走上前，大手一摆，劝架道：“内个，你们二位，先冷静下！”

“你是哪根葱啊？”

“我.....张思聪！”说完，两人脸色缓和了些，我立即继续说道：“你们这样争论是没有什么结论的，要不这样，你们直接就把你们那个360度技术当场展示下，我们这一圈观众给你们评判，放心，你瞧我们打扮，一看就是程序员，程序员是出了门的老实，一定会评判公正的。”

IN和OF一脸不可信的表情。

我没办法，直接抓住旁边人就问：“你是程序员吗？”

“我是，我是腾讯的，我做前端的。”

“哎呀，好巧啊，我也是做前端的，我是阅文的。”

“我是阿里的.....我从北京来的.....我做区块链的，我做金融的.....”大家纷纷表明了自己的身份。

“你看现在围观的全部都是程序员，技术男，绝对靠谱，不知你们意下如何？”

IN和OF面面相觑了一下，说道：“好吧，我们同意了，接下来我们依次展示我们技能，你们可都要看仔细了。”

//zxx: 双方接下来使出的技能我都加粗了

“IN前辈，你是前辈，尊老爱幼，你先来！”OF说道。

“好，恭敬不如从命！”IN回道，然后开始展示自己才艺，只见一个跃起，360度大回环，口中喊道：“我可以枚举对象。”然后一顿咔咔咔闪光灯，然后照出了下面的画面：

```
var obj = {
  a: 1,
  b: [],
  c: function () {}
};
for (var key in obj) {
  console.log(key);
}
// 结果是:
// a
// b
// c
```

“怎么样，厉不厉害？”围观众人啧啧称赞。

“呵呵！”OF冷笑了下，“这有啥，看我的！”听这口气，OF要鸟起了，众人投来期盼目光。

只见OF从背后抽出一个40米的相机，也是一顿咔咔咔，然后一个跃起，360度大回环，只听见哎呀一声，OF摔了个大跟头，画面如下：

```
var obj = {
  a: 1,
  b: [],
  c: function () {}
```

```
};  
for (var key of obj) {  
    console.log(key);  
}  
// 出错:  
// Uncaught TypeError: obj is not iterable
```

众人纷纷摇头，嘘声四起。

IN在那里哈哈仰头大笑：“OF你胆儿还真肥，连个简单的对象枚举都搞不定，也敢和我争高下！实话告诉你，我不仅可以枚举纯对象，我还可以枚举数组。”

只见一顿操作后浮现出下面的画面：

```
var arr = [3, 5, 7];  
for (var i in arr) {  
    console.log(i);  
}  
// 结果是:  
// 0  
// 1  
// 2
```

众人纷纷点头表示get到了点。

“呵呵！”OF冷笑了下，“这有啥，看我的！我可以迭代数组。”同样的口气，不过这次众人已经提前开始摇头了。只见从左往右一顿拍，然后出现了下面的画面：

```
var arr = [3, 5, 7];  
for (var i of arr) {  
    console.log(i);  
}  
// 结果是:  
// 3  
// 5  
// 7
```

围观众人似乎之前没见过这样的场景，纷纷交头接耳。

也不禁激起了我自己的兴趣：“有意思！`for...of` 原来是直接输出了数组的值。只是……”

“哈哈！”IN又得意地笑了起来，“OF啊OF，你走的是哪一出啊，你只能‘枚举’出数组的值，数组的索引你就这样放弃了？你看我，能输出索引，也能输出值。”

```
var arr = [3, 5, 7];  
for (var i in arr) {  
    console.log(i, arr[i]);  
}  
// 结果是:  
// 0 3  
// 1 5  
// 2 7
```

众人再次啧啧称赞，看来`for...in` 胜出的概率较大啊。

IN似乎有些得意过头，继续得瑟道：“我不仅可以枚举数组自身。数组的原型对象，以及数组对象本身属

性值，我都可以枚举出来！OF，你可以吗？”

```
Object.prototype.objCustom = function() {};  
Array.prototype.arrCustom = function() {};  
  
var arr = [3, 5, 7];  
arr.foo = 'hello';  
  
for (var i in arr) {  
    console.log(i);  
}  
// 结果是:  
// 0  
// 1  
// 2  
// foo  
// arrCustom  
// objCustom
```

这不得瑟还好，这一得瑟反而让OF抓住了小把柄：“哟哟哟，连原型对象都出来了，这实际工作开发，这些对象很可能是不需要的，你不管三七二十一全部枚举出来，你就不担心出问题吗？”

这一问一瞬间让IN语塞，好在还机灵，立马想到一个东西，假装淡定回复道：“这个不用担心，我有另外一件利器，名为 `hasOwnProperty`，可以避免出现你说的问题，大家请看~”

```
Object.prototype.objCustom = function() {};  
Array.prototype.arrCustom = function() {};  
  
var arr = [3, 5, 7];  
arr.foo = 'hello';  
  
for (var i in arr) {  
    if (arr.hasOwnProperty(i)) {  
        console.log(i);  
    }  
}  
// 结果是:  
// 0  
// 1  
// 2  
// foo
```

OF先是一惊，IN这老家伙还有这一手，可定睛一看，哈哈大笑了起来，“哟哟哟，你看看你，数组本身的属性看来你是怎么也摆脱不掉了哦！”

IN一瞅，哇擦，怎么还有个 `foo`，顿时尴尬不已，冷汗四起，支支吾吾，蹦了一句：“这种情况，我...我可以让我的好朋友 `forEach` 过来帮忙，他的表现是这样的~”

```
Object.prototype.objCustom = function() {};  
Array.prototype.arrCustom = function() {};  
  
var arr = [3, 5, 7];  
arr.foo = 'hello';  
  
arr.forEach(function (value, i) {  
    console.log(i);  
});
```

```
// 结果是:  
// 0  
// 1  
// 2
```

说完IN立马脸色好转，面露微笑，暗自庆幸自己人脉广，人机灵。谁知OF不断哈哈大笑，笑得IN心里毛毛的，难道又要搞出什么幺蛾子。

只见OF慢悠悠向前踱了两步，一脸轻蔑地说道：“forEach，我知道的，呵呵呵，前段时间我们刚笔试过，我的手下败将，不值一提。”然后对着围观的群众大声说道：“这个forEach啊，只能遍历数组，怎么能跟我比；就算比试‘遍历’数组，forEach跟我比也是逊色几分的。你们可能不知道吧，forEach遍历数组的时候是无法 `break` 或者 `return false` 中断的。比方说~”

```
var arr = [3, 5, 7];  
  
arr.forEach(function (value) {  
    console.log(value);  
    if (value == 5) {  
        return false;  
    }  
});  
// 结果是:  
// 3  
// 5  
// 7
```

“大家看到没，`return false` 根本就没作用，一直循环到底。”众人纷纷点头表示认同，OF继续说道：“那我就一样了，我直接一个 `break` 就能中断循环，可谓能进能出，我给大家表演下。”

只见OF腰间抽出一把40CM长刀，对着自己腰部直接就砍啊，眼睛都不眨一下，众人纷纷侧目不忍直视，胆大的人瞄了瞄，看到了下面的画面：

```
var arr = [3, 5, 7];  
  
for (let value of arr) {  
    console.log(value);  
    if (value == 5) {  
        break;  
    }  
}  
// 结果是:  
// 3  
// 5
```

大家一看，果然数组最后一项 `7` 没有输出，果然循环中断了，而且.....大家仔细看了看，OF虽然砍了自己一刀，不过看上去并没有任何伤害！众人不禁纷纷喝彩鼓掌起来。

IN在一旁看傻眼了，OF则得意洋洋，开始乘胜追击：“哎呀，有些人啊，没见过世面，不知道动漫的世界里先手都是输嘛，哈哈。现在，是时候展现真正的技术了！”只见OF口中念念有词，突然，手中白光一闪，出现了几件宝物，众人眼睛都看直了。

只见OF不紧不慢地说道：“洒家我的技能叫做‘迭代’，跟IN那种‘枚举’不是一个level的。我呢不仅可以迭代数组，我还可以直接迭代字符串，大家可以围过来看一看啊.....”

我凑上前行，果然可以啊！

```
let str = 'boo';

for (let value of str) {
  console.log(value);
}
// 结果是:
// "b"
// "o"
// "o"
```

IN支支吾吾说不出话来了，因为这个自己真的不支持，强行执行也没反应（更新：这是因为IN的记忆还停留到过去，曾经的自己是不支持字符串遍历，IE9+之后才支持）。

OF继续说道：“我还可以直接迭代**arguments**类数组对象，且看~”

```
(function() {
  for (let argument of arguments) {
    console.log(argument);
  }
})(1, 2, 3);
// 结果是:
// 1
// 2
// 3
```

“喔噢，这个好！”围观的程序员们纷纷竖起了大拇指。

“还没完呢，嘻嘻……我还可以迭代**NodeList**这类**DOM**集合，无需  `[].slice.call()`，也不需要 `Array.from()` 进行数组转化，我可以直接用起来~”

```
let elements = document.querySelectorAll('body');

for (let element of elements) {
  console.log(element.tagName);
}
// 结果是:
// "BODY"
```

“喔哦哦喔，这个更好！”围观的程序员开始欢呼，开始激动，人群涌动，已经把IN挤到了圈外。

IN在圈外，大声疾喊，“我……我也可以遍历类数组的……”，只是一个人的声音如何跟人群比，早已湮没在众人的喊叫声中。

我心想：看来，这场比试，如果没有奇迹的话，`for..of` 应该就赢了！身为发起这场比试的我已经打算宣布 `for..of` 下半场翻盘获胜，结果 `for..of` 继续不依不饶秀自己技能。

“我还可以迭代类型数组~”

```
let typeArr = new Uint8Array([0x00, 0xff]);

for (let value of typeArr) {
  console.log(value);
}
// 结果是:
// 0
// 255
```

“我还可以迭代Map~”

```
let mapData = new Map([['a', 1], ['b', 2], ['c', 3]]);

for (let [key, value] of mapData) {
  console.log(value);
}
// 结果是:
// 1
// 2
// 3
```

“我还可以迭代Set~”

```
let setData = new Set([1, 1, 2, 2, 3, 3]);

for (let value of setData) {
  console.log(value);
}
// 结果是:
// 1
// 2
// 3
```

“我还可以迭代generators.....”

完了完了，已经完全成了 `for..of` 专场了，我看了看一旁灰头土脸，毫无生气的 `for..in`，不禁担忧，这场比试会不会让 `for..in` 怀疑人生，从此从CJ圈消失啊，甚至直接从大悦城一跃而下.....我越想越后怕，赶快打住：“够了够了！”

OF见我打断其表演有些不开心，我就说了这么一句：“少侠，得饶人处且饶人。我们在场所有人都认为你武艺高强，相比 `for..in` 要更胜一筹。”OF立马脸色好了很多，一脸趾高气昂。

我平生最看不惯那些拽拽的人，OF和IN都不知道谦逊，我都没什么好感。不过身为裁判，还是要公正，于是，我就说道：“在宣布比赛结果之前，我先对两位精彩的比试做个总结。对于纯对象的遍历，`for..in` 要厉害一些，大家相信都看到了；对于数组遍历，如果不需要知道索引，`for..of` 迭代更合适，因为还可以中断；如果需要知道索引，则 `forEach()` 更合适；对于其他字符串，类数组，类型数组的迭代，虽然 `for..in` 也有这方面能力，只是从今天的比赛场面来看，`for..of` 更占上风，因此，我宣布，本次比赛的获胜者是.....”

OF在舞台中央，洋洋得意，准备享受宣布胜利时候的荣光，就在此时，人群一阵哗然，什么情况，卧槽，一群血小板突然走了过来。“卡哇伊.....”众人不禁齐呼，眼睛都放出了40瓦灯泡亮度的光，身子直接都软掉了！

“血小板，你来得正好，我们正在给 `for..in` 和 `for..of` 评判谁拍照技术好，有资格可以做你的御用摄影师。”

结果血小板用超萌的小奶音回复我：“帅气的小哥哥，可不可以我们自己选？”

“当然，当然可以！你选，我想IN和OF两位大人应该不会有意见的。”

“对的，没有异议。”自信的OF率先附和。

“我也没意见。”远处的IN随便举了个手无精打采回复道。

“那好！”只见血小板说完，走到了 `for..in` 那里，举起了 `for..in` 的双手，说道：“我要 `for..in` 做我的摄影师！”

“啊？？？”众人下巴都惊得掉在了地上，“为啥？？”

“那个，那个……”血小板萌奶音回答道，“这次的照片要在微软名为IE的杂志和网站上展示，`for..of` 大人虽然技能更广，但是毕竟是ES6新特性，IE浏览器不支持，因此，这次我选兼容性更好的 `for..in` ！”

然后，众人就眼睁睁看着 `for..in` 在众多萌萌的血小板的包围中离开了！而 `for..of` 双手撑地归在冰冷的地面上，留下了悲伤的泪水。

“唉……”众人纷纷叹息，“世事难料啊！”然后人群逐渐散开了。

我驻足了片刻，也无能为力，准备离开，突然想到一件事情，我一看手机，我拉个擦，出门到现在已经过去1个小时60分钟了！想到晚上爽脆的搓衣板，我也双手撑地归在冰冷的地面上，留下了悲伤的泪水。

此刻，偌大的展馆中央，有两人悲伤的人儿，他们的泪水，让今年的ChinaJoy显得格外得动人！

## 参考文章

- [MDN for...of](#)

感谢阅读，如果你觉得故事写得不错，欢迎分享给其他小伙伴一起来过来围观 `for..in` 和 `for..of` 吵架。

《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

(本篇完) // 想要打赏？[点击这里](#)。有话要说？[点击这里](#)。



« [原来浏览器原生支持JS Base64编码解码](#)

[这回试试使用CSS实现抛物线运动效果»](#)

## 猜你喜欢

- [ES5中新增的Array方法详细说明](#)
- [ES6 JavaScript Promise的感性认知](#)
- [HTMLUnknownElement与HTML5自定义元素的故事](#)
- [简单了解ES6/ES2015 Symbol\(\) 方法](#)
- [JS字符串补全方法padStart\(\)和padEnd\(\)简介](#)
- [万岁，浏览器原生支持ES6 export和import模块啦！](#)

分享到： 1



## 发表评论（目前60条评论）

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

### 1. luogeger说道:

2018年11月10日 16:33

画风依旧，还是那么骚气，应该是媳妇灌输的，总结是真特么棒，看的是真特么爽

[回复](#)



### 2. 啲唧说道:

2018年11月1日 15:11

大佬戏好多

[回复](#)



### 3. tamashii说道:

2018年10月21日 00:25

大佬，我之前一直关注您的空间。

今天再回来大佬您的空间画风咋还不对了啊！！？？！！！！到底这些年发生了啥！！！！！！！！

[回复](#)



#### 张鑫旭说道:

2018年10月21日 21:26

以前太粗犷了，稍微精致了下。

[回复](#)



### 4. ?说道:

2018年09月23日 19:06

作者是b站的把！

[回复](#)



5.

^^说道：  
2018年09月19日 18:22

大大不亏是阅文的啊，这文笔足以让许多作者汗颜了，哈哈

回复



6.

重形说道：  
2018年09月13日 16:51

看不懂，什么语言写的？

回复



7.

mars说道：  
2018年09月8日 20:41

不正经中看正经

回复



8.

神秘说道：  
2018年09月7日 16:11

哈哈，这真是我看得最入迷的一篇文章了

回复




9.

zzy说道：  
2018年09月4日 16:45

请问，我再chrome控制台中使用for...in遍历字符串，表现跟遍历数组一样，但是文中为什么说“IN支支吾吾说不出话来了，因为这个自己真的不支持，强行执行只会报错。”


回复



ading说道：  
2018年09月12日 16:59

你能用for ... in ... 遍历出字符串的每个字符？


回复



hello-zy说道：  
2018年09月18日 11:28

确实可以啊，，，，，


回复



肉山大魔王说道：  
2018年09月15日 22:48

我也是.....感觉有点懵逼.....我用for...in遍历arguments和NodeList也都可以


回复



zmq说道：  
2018年12月29日 11:38

你确定出来是字符不是下标

回复




10.

mask说道:

2018年09月3日 10:23

这就是传说中的跨界吧，用写小说的功力写js教程，佩服佩服！

回复



11.

zzzzKK说道:

2018年08月31日 16:46

最不正经的人永远内心是最缜密的。旭哥就真滴厉害

回复



12.

leslieHe说道:

2018年08月29日 11:01

有意思

回复


13.

华仔说道:

2018年08月29日 10:21

```
function getMenu(d) {
  var a = '';
  for (var m in relAliasData) {
    a += m + ' ' + relAliasData[m] + ' '
    if (d.wife) {
      relAliasData['wife'] = ''
    }
    if (d.father) {
      relAliasData['father'] = ''
    }
    if (d.mother) {
      relAliasData['mother'] = ''
    }
    if (d.male) {
      relAliasData['husband'] = ''
    }
    if (d.female) {
      relAliasData['wife'] = ''
    }
  }
  S('#demo').html(a);
}
getMenu()
```

这样写有用吗?

回复



14.

华仔说道:

2018年08月28日 09:57

for in循环的时候能不能加判断呢?


回复


15.

系统管理员说道:

2018年08月27日 17:30

forin可以遍历字符串吧



[回复](#)

华仔说道:

2018年08月29日 09:27

可以加个QQ咨询问题吗?

[回复](#)



16. **MJPiero**说道:

2018年08月24日 18:31

就喜欢老师不正经的说着一本正经的事情。

[回复](#)



17. **Will**说道:

2018年08月24日 10:05

遍历 object 需要配合 Object.keys / Object.values / Object.entries 服用啦。

```
for(let [key, value] of Object.entries(obj)) {  
  console.log(key, value);  
}
```

[回复](#)



18. **WangNianyi2001**说道:

2018年08月24日 02:03

md 谁说`for of`不能迭代 key 了  
`Object.entries()`了解一下

[回复](#)



19. **WJT**说道:

2018年08月23日 16:36

受教了!

[回复](#)



20. **三娃**说道:

2018年08月21日 16:04

感到愉快

[回复](#)



21. **三月三荷**说道:

2018年08月20日 17:05

我们课本上如果能这么写就好玩啦

[回复](#)



22. **manguo**说道:

2018年08月16日 18:01

好搞笑啊, 又能学习东西

[回复](#)



23.

后知后觉说道：  
2018年08月16日 09:06

大神厉害啊，这么一讲，记忆深刻啊。浏览器兼容果然是前端最大的痛点

回复

weapon张说道：  
2018年08月21日 05:34

不,是IE,IE是最大的痛点

回复

飘渺说道：  
2018年09月6日 17:09

UC怎么说

回复

24.

牛逼说道：  
2018年08月16日 00:58

ggaag

回复

25.

盖大楼说道：  
2018年08月14日 17:35

淫才胸

回复

牛逼说道：  
2018年08月16日 00:58

hahh

回复

gfhfg说道：  
2018年09月9日 21:53

看见客户款进口黄金客户

回复

gfhfg说道：  
2018年09月9日 21:53

回国后就赶紧赶紧改价格会经过加工会根据规划

回复

26.

想做自己喜欢的事说道：  
2018年08月14日 17:14

懂的不多的我默默打卡

回复

27.

alllle说道：  
2018年08月13日 18:16

总体来说

for..in迭代可枚举属性，数组本身也是以key-value形式存储的只不过key是index索引  
因此for in 出来的只有index

for..of只能迭代可转为数组的对象，类似先用 [].slice.call()操作一番，处理的是数组元素

至于break，forEach不可以break但是可以换every或者some，同样可以中断循环

for in 取数组的索引，for of 取数组的值，只是这个for in 的腿伸的有点长，而for of 的嘴有点短

[回复](#)

**Will**说道：

2018年08月24日 10:13



其实不是哦，for..of能遍历一切实现了Symbol.iterator的对象，数组正好实现了罢了。for-of最直观的是可以遍历生成器的返回值。

```
var obj = {
  *[Symbol.iterator]() {
    yield 1;
    yield [];
    yield "aa";
  }
}
```

```
for(let a of obj) {
  console.log(a)
}
```

— or —

```
var G = function*() {
  yield 1;
  yield [];
  yield "aa";
}
```

```
for(let a of G()) {
  console.log(a)
}
```

[回复](#)

28. **心满**说道：

2018年08月13日 00:29



真他娘人才 d====(〃▽〃\*)b

[回复](#)

29. **anran758**说道：

2018年08月12日 22:57



对象只能用for in吗...如果有babel支持的话，用新特性哪个性能会更好？

[回复](#)

30. **jinjing**说道：

2018年08月10日 15:59



我写的好肤浅

[回复](#)

31. **恋红尘→枫**说道：

2018年08月10日 14:14



说了这么多，只为介绍for of。。。

[回复](#)

32.

hello说道：

2018年08月10日 14:08

真的很nice

回复


33.

MerryGirl说道：

2018年08月8日 22:55

被血小板炸出来的我

回复


34.

何碧小小说道：

2018年08月8日 21:04

牛逼。。。

回复


35.

入里说道：

2018年08月8日 17:22

大大一如既往的幽默

回复


36.

falseLuffy说道：

2018年08月8日 15:16

前端去了阅文，都会写小说了； ^~

回复


37.

蒲公英说道：

2018年08月8日 13:04

这种风格很有趣

回复


38.

Tony说道：

2018年08月7日 18:56

说得好！我选择babel成...emmm... of会被编译成啥来着

回复


39.

九月说道：

2018年08月7日 16:49

赞，皮。

回复


40.

赞！说道：

2018年08月7日 15:12

get

回复



41.

xxx说道:

2018年08月7日 14:17

厉害厉害

回复


42.

我自向天笑说道:

2018年08月7日 11:03

等您上传跪搓衣板照我保证转发。。。

回复


43.

菜鸟说道:

2018年08月7日 10:09

每天开班第一件事打开鑫空间看看有没有最新更新.....

回复

mfk说道:

2018年08月14日 17:41

为什么不用RSS订阅?

<https://www.zhangxinxu.com/wordpress/feed/>

回复


44.

广建说道:

2018年08月7日 10:02

回家跪爽脆的搓衣板吧

回复


45.

webster说道:

2018年08月7日 09:25

geliabile

回复


46.

kk说道:

2018年08月7日 07:54

哈哈哈哈哈，猜到结局

回复


47.

chenmeng说道:

2018年08月7日 01:23

张大大还是一贯的幽默生动的风格，笔芯(●'◡'●)/❤

回复



#### 最新文章

- » 常见的CSS图形绘制合集
- » 粉丝群第1期CSS小测点评与答疑
- » 分享三个纯CSS实现26个英文字母的案例



- » [小tips: 纯CSS实现打字动画效果](#)
- » [CSS/CSS3 box-decoration-break属性简介](#)
- » [CSS :placeholder-shown伪类实现Material Design占位符交互效果](#)
- » [从天猫某活动视频不必要的3次请求说起](#)
- » [CSS vector-effect与SVG stroke描边缩放](#)
- » [CSS ::backdrop伪元素是干嘛用的?](#)
- » [周知: CSS -webkit-伪元素选择器不再导致整行无效](#)

#### 今日热门

- » [常见的CSS图形绘制合集](#) <sup>(178)</sup>
- » [粉丝群第1期CSS小测点评与答疑](#) <sup>(113)</sup>
- » [未来必热: SVG Sprite技术介绍](#) <sup>(111)</sup>
- » [HTML5终极备忘大全 \(图片版+文字版\)](#) <sup>(85)</sup>
- » [让所有浏览器支持HTML5 video视频标签](#) <sup>(83)</sup>
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) <sup>(80)</sup>
- » [CSS3下的147个颜色名称及对应颜色值](#) <sup>(78)</sup>
- » [小tips: 纯CSS实现打字动画效果](#) <sup>(72)</sup>
- » [写给自己看的display: flex布局教程](#) <sup>(69)</sup>
- » [分享三个纯CSS实现26个英文字母的案例](#) <sup>(69)</sup>

#### 今年热议

- » [《CSS世界》女主角诚寻靠谱一起奋斗之人](#) <sup>(76)</sup>
- » [不借助Echarts等图形框架原生JS快速实现折线图效果](#) <sup>(64)</sup>
- » [看, for..in和for..of在那里吵架!](#) <sup>(60)</sup>
- » [是时候好好安利下LuLu UI框架了!](#) <sup>(47)</sup>
- » [原来浏览器原生支持JS Base64编码解码](#) <sup>(35)</sup>
- » [妙法攻略: 渐变虚框及边框滚动动画的纯CSS实现](#) <sup>(33)</sup>
- » [炫酷H5中序列图片视频化播放的高性能实现](#) <sup>(31)</sup>
- » [CSS scroll-behavior和JS scrollIntoView让页面滚动平滑](#) <sup>(30)</sup>
- » [windows系统下批量删除OS X系统.DS\\_Store文件](#) <sup>(26)</sup>
- » [写给自己看的display: flex布局教程](#) <sup>(26)</sup>

#### 猜你喜欢

- [ES5中新增的Array方法详细说明](#)
- [ES6 JavaScript Promise的感性认知](#)
- [HTMLUnknownElement与HTML5自定义元素的故事](#)
- [简单了解ES6/ES2015 Symbol\(\) 方法](#)
- [JS字符串补全方法padStart\(\)和padEnd\(\)简介](#)
- [万岁, 浏览器原生支持ES6 export和import模块啦!](#)