

借助HTML5 details,summary无JS实现各种交互效果

这篇文章发布于 2018年01月15日，星期一，01:52，归类于 [HTML相关](#)。阅读 16203 次, 今日 7 次 [25 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=7312>

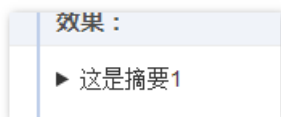
本文可全文转载，但需得到原作者书面许可，同时保留原作者和出处，摘要引流则随意。

一、了解HTML5 details, summary默认交互行为

`<details>` 标签在Chrome，Firefox等浏览器下默认是有展开收起行为的，例如下面HTML：

```
<details>
  <summary>这是摘要1</summary>
  <p>这里具体描述，标签相对随意，例如这里使用的<lt;p>标签。</p>
</details>
```

结果UI表现为：



具体描述为：

1. 只显示了 `<summary>` 标签内容，而 `<p>` 默认隐藏了；
2. `<summary>` 标签前面出现了一个小三角；

小三角图形的隐喻是：我是可点击的，点击我可能会出现宝箱。

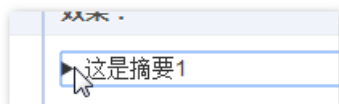
OK，我们不妨就点击一下，结果如下图：



具体描述为：

1. 原本隐藏的 `<p>` 标签显示出来了；
2. `<summary>` 标签前面的小三角方向朝下了；

此时我们再一次点击，`<p>` 标签内容又会隐藏收起，箭头方向还原，如下图：



活脱脱一个天然的展开收起效果。

展开与收起是通过open属性控制的

通过在 `<details>` 标签上添加布尔类型的 `open` 属性，可以让我们的详情信息默认就是展开状态，如下HTML示意：

```
<details open>
  <summary>这是摘要2</summary>
  <content>这里<details>;标签设置了HTML布尔属性open，因此，默认是展开状态。</content>
</details>
```

结果如下截图：



如果我们使用JS脚本手动移除这个 `open` 属性，即使没有点击行为的发生，我们内容也会收起。

`<summary>`如果缺省

`<summary>` 标签如果缺省，则 `<details>` 元素会在内部自动创建一个 `<summary>` 内容，默认文案是“详细信息”。如下HTML代码：

```
<details open>
  <p>如果<summary>;缺省，则会自动补上，文案是“详细信息”。</p>
</details>
```

结果如下截图所示：



您可以狠狠地点击这里：[HTML details,summary基本效果demo](#)

二、details浏览器内置UI可以自定义

`<details>` 标签默认的小三角样式有些简陋，在实际应用的时候，往往不是我们希望的样子，不用担心，我们是可以进行自定义的。在Chrome等浏览器下使用 `::-webkit-details-marker`，在Firefox浏览器下使用 `::-moz-list-bullet` 可以对小三角进行UI控制，例如改变颜色，改变大小，使用自定义的图形代替，或者直接隐藏等，我们来看几个简单的案例。

案例1：小三角右侧显示同时颜色变淡

HTML代码如下：

```
<details class="details-1" open>
  <summary>这是示例1</summary>
  <content>本案例展示对小三角UI重定义：包括显示在右侧，颜色减淡等。</content>
</details>
```

CSS如下：

```
.details-1 summary {
  width: -moz-fit-content;
  width: fit-content;
  direction: rtl;
}
.details-1 ::-webkit-details-marker {
  direction: ltr;
```

```

    color: gray;
    margin-left: .5ch;
}
.details-1 ::-moz-list-bullet {
    direction: ltr;
    color: gray;
    margin-left: .5ch;
}

```

结果如下图所示：



当我们点击摘要标题升起的时候，表现为下图（截自Firefox）：



眼见为实，您可以狠狠地点击[这里](#)：[HTML details小三角位置颜色改变demo](#)

而实际上实际开发的时候，对小三角UI更便捷的定制方法是：隐藏浏览器原生的小三角，然后借助 `::before` 或 `::after` 伪元素重新生成我们想要的UI效果，下面这个案例就将展示相关的处理。

案例2：隐藏浏览器原生的小三角并使用自定义三角替换

HTML结构还是类似的：

```

<details class="details-2" open>
  <summary>这是示例2</summary>
  <content>本案例隐藏原生小三角，使用自定义小三角。</content>
</details>

```

CSS主要分为2部分，一部分是隐藏浏览器原生的小三角，另外一部分是使用伪元素生成自定义的三角效果。

首先看一下隐藏 `<details>` 标签默认的小三角的CSS：

```

/* 隐藏默认三角 */
.details-2 ::-webkit-details-marker {
    display: none;
}
.details-2 ::-moz-list-bullet {
    font-size: 0;
}

```

可以看到Chrome浏览器和Firefox浏览器的小三角隐藏采用的是不同的策略。在Chrome浏览器下，我们可以直接设置 `display:none` 进行隐藏，但是这一招在Firefox浏览器下确实没有效果的，即使设置 `display:none!important` 也是如此，根据我的测试，只有 `font-size:0` 能够比较完美的隐藏。类似 `position:absolute;visibility:hidden` 这种常见的隐藏也是不行的，因为 `position:absolute` 无法生效。

然后是自定义小三角显示的CSS，这里采用的是 `::after` 伪元素模拟的：

```

/* 自定义的三角 */
.details-2 summary::after {
    content: '';
    position: absolute;
}

```

```
width: 1em; height: 1em;
margin: .2em 0 0 .5ch;
background: url(./arrow-on.svg) no-repeat;
background-size: 100% 100%;
transition: transform .2s;
}
.details-2:not([open]) summary::after {
margin-top: .25em;
transform: rotate(90deg);
}
```

最终效果如下图所示：



收起时候：



眼见为实，您可以狠狠地点击这里：[HTML details小三角自定义demo](#)

最后有一点需要注意一下，就是如果 `<details>` 标签内并没有 `<summary>` 元素，则我们的对三角的自定义代码都是无效的，可以使用一个空的 `<summary>` 元素占位，类似这样：

```
<details>
  <summary></summary>
  <content>内容。</content>
</details>
```

三、Chrome浏览器下点击时候outline轮廓等体验处理

UI可以定制了，但是还有个不容忽视的体验问题，那就是在Chrome浏览器下点击时候会出现 `outline` 轮廓，如下图所示：



在实际项目开发的时候，产品和设计一定会让你把这个效果去掉的。以及，当我们 `<summary>` 元素点击较快的时候，文本会被选中，也不是我们想看到的。

阻止文本选中，我们可以：

```
summary {
  -webkit-user-select: none;
  -moz-user-select: none;
  -ms-user-select: none;
  user-select: none;
}
```

对于 `outline` 轮廓，比较直接的做法是：

```
summary {
  outline: 0;
}
```

但是这样处理对无障碍访问而是非常不友好的，那有没有什么办法兼顾视觉体验和无障碍访问体验呢？

我的做法是这样子的：

1. 利用<a>标签的outline交互体验

浏览器对 `<a>` 标签元素的 `outline` 轮廓进行了专门的体验优化处理，鼠标点击的时候不显示轮廓，键盘访问时候显示轮廓。于是我们可采用李代桃僵策略，让 `<summary>` 元素的 `outline` 交给 `<a>` 元素，方法就是在 `<summary>` 中再内嵌一个 `<a>`，同时通过 `tabindex` 属性remove掉 `<summary>` 原本的可访问性。HTML代码示意如下：

```
<details open>
  <summary tabindex="-1"><a href="javascript:">这是示例</a></summary>
  <content>点击无外框，键盘focus有。</content>
</details>
```

CSS如下：

```
summary {
  user-select: none;
  outline: 0;
}
summary a {
  color: inherit;
}
```

此时，在Chrome浏览器下，我们点击摘要信息，没有任何 `outline` 轮廓出现；但是当我们使用Tab键索引时候，可以看到下图所示的轮廓效果：



轮廓区域比原生的 `<summary>` 要小，但这无伤大雅，而且实际项目开发的时候，我们会去掉小箭头，此时只要设置 `<a>` 标签 `display:block`，则轮廓就可以和 `<summary>` 保持一致了。

接下来，我们按下Space空格键，就会发现 `<details>` 元素内的内容信息不断的展开与收起：



您可以狠狠地点击这里：[HTML5 summary outline轮廓a标签处理demo](#)

然后上面实现并不完美，相比原生的 `<summary>` 元素，Enter回车键展开收起效果丢失了。这是因为HTML元素中如果多个 `focusable` 同时带 `click` 浏览器行为元素嵌套的时候，点击里面的元素，外部元素的浏览器行为是不会触发的。类似的有 `<label>` 内嵌 `<a>` 标签。

对于 `<a>` 标签，其浏览器行为只能通过回车键触发，空格键是无效的；但是对于 `<summary>`，回车键和空格键都能触发展开收起行为，这就是为什么上面代码空格键有效，回车键无效的原因。

如果想要同时支持回车键展开与收起，可以对HTML如下处理：

```
<details open>
  <summary tabindex="-1"><a href="javascript:" onClick="this.parentNode.click();">这是示例</a></summary>
  <content>点击无外框，键盘focus有。</content>
</details>
```

需要注意的是上面处理在 `<summary>` 自己额外绑定 `click` 事件时候可能会有double触发的问题，此时

，阻止 `<a>` 元素的冒泡即可。

2. JS捕获键盘行为手动设置outline

这个方法不需要对HTML进行任何的改动，是通过CSS和JS配合对全局的 `<summary>` 元素进行 `outline` 优化。

CSS如下：

```
summary {
  user-select: none;
  outline: 0;
}
summary:focus {
  outline: 1px dotted;
  outline: 5px auto -webkit-focus-ring-color;
}
```

JS如下：

```
window.addEventListener('keydown', function () {
  window.isKeyEvent = true;
  setTimeout(function () {
    window.isKeyEvent = false;
  }, 100);
});

document.addEventListener('focusin', function (event) {
  var target = event.target;
  if (target && target.tagName.toLowerCase() == 'summary' && window.isKeyEvent == true) {
    target.setAttribute('focus', '');
  }
});

document.addEventListener('focusout', function (event) {
  var eleFocusAll = document.querySelectorAll('summary:focus');
  [].slice.call(eleFocusAll).forEach(function (summary) {
    summary.removeAttribute('focus');
  });
});
```

只要把上面的CSS和JS复制到页面中，视觉体验和交互体验完美支持的 `<summary>` 元素 `outline` 效果就有了。

表现为，点击 `<summary>` 没有任何 `outline`，键盘 `focus` 时候出现，且和浏览器原生 `outline` 效果一模一样，Space键和Enter键展开与收起访问完全保留。

眼见为实，您可以狠狠的点击这里：[HTML5 summary outline轮廓JS优化demo](#)

例如下图就是键盘Tab键 `focus` 后回车后的效果：



每每看到如此极致的用户体验处理，心情都大好。

原理：

关键是全局监听 `keydown` 事件，如果有发生，则认为此100毫秒内的页面 `focus` 行为均是键盘产生，

从而有效区分是点击触发的 `focus` 行为还是键盘触发的 `focus` 行为，如果是键盘触发，给 `<summary>` 元素手动增加 `outline` 效果。

四、基于details元素行为的各种交互效果案例

了解了 `<details>` 元素的点击交互行为；解决了UI定制难题；解决了 `outline` 的体验问题，下面我们就可以付诸实践，不借助任何JS来实现各种我们平常见到的交互效果。

案例1：“更多”展开与收起效果

实现最终效果如下gif:



因为“更多”元素是在底部，因此效果实现的要点的所有的内容信息都放在 `<summary>` 元素内部，然后通过 `<details>` 元素的 `open` 属性控制UI的变化。

HTML和CSS代码如下，其中，最核心部分已经红色高亮：

```
<details>
  <summary>
    <p>据台媒报道，大...青睐。</p>
    <div class="more">
      <p>其他几首歌曲...</p>
    </div>
    <a>更多</a>
  </summary>
</details>
```

```
::-webkit-details-marker {
  display: none;
}
::-moz-list-bullet {
  font-size: 0;
  float: left;
}
.more {
  display: none;
}
[open] .more {
  display: block;
}
[open] summary a {
  font-size: 0;
}
[open] summary a::before {
  content: '收起';
  font-size: 14px;
}
```

把“更多”对应的信息放在 `.more` 元素内，然后通过 `[open]` 属性选择器控制器显示，效果即达成。

您可以狠狠的点击这里：[HTML5 details/summary更多展开收起demo](#)

案例2：无JS实现点击显示悬浮菜单，自定义下拉框等效果

效果如下gif:

没有任何JS参与。HTML结构如下：

```
<details>
  <summary>我的消息</summary>
  <div class="box">
    <a href>我的回答<sup>12</sup></a>
    <a href>我的私信</a>
    <a href>未评价订单<sup>2</sup></a>
    <a href>我的关注</a>
  </div>
</details>
```

然后CSS让 `.box` 元素绝对定位即可，显示和隐藏 `<details>` 元素内置行为就搞定了。

您可以狠狠地点击这里：[HTML5 details/summary悬浮菜单demo](#)

案例3：accordion多项折叠效果

此效果常见于条目比较多的垂直导航栏，新闻条目等。

例如下面实现的效果：



这个更加简单了，就是一堆 `<details>` 元素并排放置就可以了，如下HTML：

```
<details open>
  <summary><dt>订单中心</dt></summary>
  <dd><a href>我的订单</a></dd>
  <dd><a href>我的活动</a></dd>
  <dd><a href>评价晒单</a></dd>
  <dd><a href>购物助手</a></dd>
</details>
<details open>
  <summary><dt>关注中心</dt></summary>
  <dd><a href>关注的商品</a></dd>
  ...
</details>
<details open>
  ...
</details>
```

计算CSS没有任何设置，效果也天然达成。

您可以狠狠地点击这里：[HTML5 details/summary多列菜单折叠demo](#)

案例4：带slideUp/slideDown效果的多项折叠菜单

案例3中的展开项显示的时候是非常生硬的突然显示，实际上我们可以借助一些选择器技巧以及CSS3 `transition` 属性让菜单展开收起的时候是有动画效果的，效果如下gif截图：



此效果实现原理核心是 `[open]` 属性选择器，和加号 `+` 相邻兄弟选择器。

首先看下HTML，展开列表结构发生了变化，不是作为 `<details>` 的子元素，而是作为其相邻兄弟元

素存在，HTML示意：

```
<details open><summary>订单中心</summary></details>
<dl>
  <dd><a href>我的订单</a></dd>
  <dd><a href>我的活动</a></dd>
  <dd><a href>评价晒单</a></dd>
  <dd><a href>购物助手</a></dd>
</dl>
...
```

上面 `<dl>` 定义列表就是展开收起的内容，其作为兄弟元素和 `<details>` 元素平起平坐，于是，我们就可以利用点击 `<summary>` 元素 `<details>` 元素的 `open` 属性会变化的特性实现我们想要的动画效果，CSS如下：

```
details + dl {
  max-height: 0;
  transition: max-height .25s;
  overflow: hidden;
}
[open] + dl {
  max-height: 100px;
}
```

借助相邻兄弟选择器以及 `max-height` 任意元素slideUp/slideDown技术就可以效果达成。

您可以狠狠地点击这里：[HTML5 details/summary多列菜单滑入滑出demo](#)

案例5：多级嵌套的树形菜单交互效果

这里的树形菜单效果实现也很简单，多个 `<details>` 元素相互嵌套就可以，效果Gif如下：



HTML结构大致如下：

```
<details>
  <summary>我的视频</summary>
  <details>
    <summary>爆肝工程师的异世界狂想曲</summary>
    <div>tv1-720p.mp4</div>
    <div>tv2-720p.mp4</div>
    ...
    <div>tv10-720p.mp4</div>
  </details>
  <details>
    <summary>七大罪</summary>
    <div>七大罪B站00合集.mp4</div>
  </details>
  <div>珍藏动漫网盘地址.txt</div>
  <div>我们的小美好.mp4</div>
</details>
```

CSS的主要工作就是绘制菜单前面的加号和减号图形，例如我们可以借助background线性渐变，具体可参见文章“[CSS3 linear-gradient线性渐变实现虚线等简单实用图形](#)”，相关CSS如下：

```
details {
```

```
padding-left: 20px;
}
summary::before {
  content: '';
  display: inline-block;
  width: 12px; height: 12px;
  border: 1px solid #999;
  background: linear-gradient(to right, #999, #999) no-repeat center, linear-gradient(to
top, #999, #999) no-repeat center;
  background-size: 2px 10px, 10px 2px;
  vertical-align: -2px;
  margin-right: 6px;
  margin-left: -20px;
}
[open] > summary::before {
  background: linear-gradient(to right, #999, #999) no-repeat center;
  background-size: 10px 2px;
}
```

效果即达成！

您可以狠狠地点击这里：[HTML5 details/summary树形菜单demo](#)

五、如果只想要details/summary的语义不要行为

如果只想要 `<details>` 元素，`<summary>` 元素的语义，但是并不需要点击展开收起的行为，该怎么处理呢？

例如，某评论，或者某帖子有标题和正文，非常符合详情-概要-内容的语义，但是希望是纯展示的，点击时候不收起，可以这么处理：

1. `<summary>` 标签设置 `tabindex="-1"` 让键盘无法访问；
2. 设置CSS：

```
summary {
  outline: 0;
  pointer-events: none;
}
```

这样就不能点，也不会有 `outline` 轮廓。

六、兼容性以及Polyfill

兼容性如下图：



除了IE和Edge浏览器，大好河山一片绿，至少移动端可以用得比较开心。

如果想要在桌面web网页使用 `<details>` 元素的棒棒哒特性，我们可以对其进行Polyfill，可以参见此项目：<https://github.com/javan/details-element-polyfill>

对键盘访问，事件toggle都做了兼容。

如果开发策略是对不支持的IE进行特异处理，则下面的JS判断是否支持 `<details>` 元素的脚本可能对你有帮助：

```
var isSupportDetails = 'open' in document.createElement('details');
```

六、结束语么么哒

无JS实现的好处有：

1. 省了代码，加载快了；
2. 实现更简单了，开发快了；
3. JS还没加载交互也能进行，体验好了；
4. 键盘无障碍和aria阅读设备无障碍天然支持，体验档次高了。
5. 可以跟同事炫，逼格上去了。

好，感谢阅读，行为仓促，欢迎纠错。

《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](#)。



« 瞎折腾：把JS,CSS任意文本文件加密成一张图片

CSS :focus-within伪类选择器及纯CSS下拉等应用举例 »

猜你喜欢

- HTML tabindex属性与web网页键盘无障碍访问
- 实力科普：为什么浮层或弹框一定要有叉叉关闭按钮?
- CSS :focus伪类JS focus事件提高网站键盘可访问性
- span与a元素的键盘聚焦性以及键盘点击性研究
- HTML accesskey属性与web自定义键盘快捷访问
- 翻译-盲人如何使用互联网的8个误区
- 页面可用性之outline轮廓外框的一些研究
- js下拉菜单实现与可访问性问题的一些思考
- 小卖弄：纯CSS实现的outline切换transition动画效果
- 是时候好好安利下LuLu UI框架了！
- 说说HTML5中label标签的可访问性问题

分享到： 1

标签： details, focus, HTML5, outline, summary, tabindex, 可访问性, 键盘

发表评论（目前25条评论）

名称 (必须)

邮件地址(不会被公开) (必须)

网站

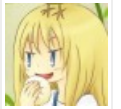
提交评论

1. **suu**说道:

2018年12月3日 23:46

想在为知笔记里实现折叠却不能用JS 差点放弃了
没想到还有乳齿实用且简单的方法
老年人并不想系统学习 决定在这蹲点了
以及感到ie以后要没盆友

[回复](#)



2. **XboxYan**说道:

2018年10月18日 22:38

旭哥，有人抄袭你的博客(o'ω'o)

<https://segmentfault.com/a/1190000016718878>

[回复](#)



3. **ali**、说道:

2018年04月25日 14:31

好像content标签不识别！

[回复](#)



4. **GMBUG**说道:

2018年02月17日 16:35

我觉得未来几年我都在这里混了，感谢大神!
我是全栈型培训生，但是比较喜欢html!
未来10年打算进军Android或者全息领域!
一个中职放弃上大专选择IT的我！然后培训完还想继续回到大专，在专升本（主要是和大神一样，来自乡村，所以后面不说就知道了）!
今年年底培训完才19不过20，天哪，好慌张!
隐藏的技术都是靠个人从网上自己扒出来的!
培训完我觉得还是闭关在修炼到20吧!
膜拜大神

[回复](#)



5. **yugasun**说道:

2018年01月25日 10:41

不知道为什么，每次看鑫大大的文章，总有一种提神醒脑的感觉

[回复](#)



6.

鑒金圣手火麒麟说道:

2018年01月23日 14:11

鑫大大, 请问可以转载到自己的小团队分享吗? (内网)

回复

张 鑫旭说道:

2018年01月23日 20:25

可以, 保留原出处即可。

回复

鑒金圣手火麒麟说道:

2018年01月30日 11:02

谢谢!

回复


7.

getup说道:

2018年01月23日 10:48

请收下在下的膝盖

回复


8.

八阿哥说道:

2018年01月20日 16:53

《css世界》论坛发不了帖子, 邮箱验证不了。建议给书中的例子做一个索引页面, 方便打开。还有最好建个QQ群, 方便大家交流沟通。

回复


9.

cshenger说道:

2018年01月19日 17:35

论奇技淫巧我只服张老师

回复


10.

雪中华尔兹说道:

2018年01月19日 10:39

睡前看篇文章也是不错

回复


11.

我是谁说道:

2018年01月19日 10:38

不错学习了

回复


12.

范特西说道:

2018年01月19日 10:32

推荐给我们的前端人员


回复



13. 王熊喵说道:
2018年01月16日 12:32
啥都别说了, 先把七大罪交出来
[回复](#)




14. 后知后觉说道:
2018年01月15日 16:25
点击收起, 那个, 果断用起来。其实这个标签在刚接触h5时就用过, 但是时间长了, 完全忘记他了, 因为这种效果的js也很简单, 所以一直没用。
[回复](#)



15. 天凉好个秋说道:
2018年01月15日 15:12
<http://bbs.cssworld.cn>
论坛无法完成发帖步骤啊~~卡在邮箱验证了
[回复](#)

张鑫旭说道:
2018年01月21日 15:38
已取消了邮箱限制。
[回复](#)




16. 夜深未眠说道:
2018年01月15日 09:27
我只想献上我的膝盖。
佩服老师:

- 深入, 深入骨髓的探索
- 细致, 细致入微的讲解

[回复](#)




17. 怡红公子说道:
2018年01月15日 08:46
其实后面的几个案例都可以用 checkbox 加相邻选择器实现, 兼容性还比这个好。
[回复](#)





18. edx说道:
2018年01月15日 02:26
睡前看篇文章也是不错的。还有, 为什么要在我买了css世界之后搞个签名版? 而且还比我买的便宜.....
[回复](#)

LGGG说道:
2018年01月15日 11:27
哈哈 我也是 前段时间买了css世界, 没想到还有签名版, 不过都能看. 哈哈
[回复](#)

jack说道:
2018年12月24日 14:43
哪有签名版
[回复](#)







张 鑫旭说道:

2018年12月24日 14:57

这里: <https://weidian.com/item/?itemID=2225312451>



web说道:

2018年01月17日 11:08

哪里有签名版

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

今年热议

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

猜你喜欢

- [HTML tabindex属性与web网页键盘无障碍访问](#)
- [实力科普：为什么浮层或弹框一定要有叉叉关闭按钮？](#)
- [CSS :focus伪类JS focus事件提高网站键盘可访问性](#)
- [span与a元素的键盘聚焦性以及键盘点击性研究](#)
- [HTML accesskey属性与web自定义键盘快捷访问](#)
- [翻译-盲人如何使用互联网的8个误区](#)
- [页面可用性之outline轮廓外框的一些研究](#)
- [js下拉菜单实现与可访问性问题的一些思考](#)
- [小卖弄：纯CSS实现的outline切换transition动画效果](#)
- [是时候好好安利下LuLu UI框架了！](#)
- [说说HTML5中label标签的可访问性问题](#)