

CSS :focus伪类JS focus事件提高网站键盘可访问性

这篇文章发布于 2017年04月23日，星期日，22:22，归类于 [Web综合](#)。阅读 16523 次, 今日 8 次 11 条评论

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=6102>

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、关于键盘可访问性

所谓“键盘可访问性”，指的是用户只使用键盘访问网站的能力。

例如，我们的iMac鼠标没电了，或者鼠标坏了，或者在智能电视中访问我们的网站，此时就只能依赖于键盘访问了。

键盘访问网站的常用操作包括：

- Tab键索引控件元素；
- Enter键触发当前处于focus状态的点击行为；
- 上下键上下滚动网页；
- Space空白键滚动一屏网页；
- Home键返回顶部；
- End键滚动到底部；

等等。

其中Tab键索引和Enter键点击是与交互相关的最重要的两个键盘行为。

一般的操作行为是这样的，先Tab键按次序不断 `focus` 控件元素，包括链接，按钮，输入框等表单元素或者 `focus` 设置了 `tabindex` 的普通元素，处于 `focus` 状态元素，浏览器一般会通过虚框或者外发光的形式进行区分和提示，此时我们在按下Enter回车键，就相当于鼠标点击了这个元素，从而可以前往我们想去的目的地，或者执行我们想要的交互效果。

而 `focus` 状态元素的标记默认全部都是使用 `outline` 属性，因此，类似下面的代码其实是不专业的：

```
* {  
  outline: 0 none;  
}
```

```
a {  
  outline: 0 none;  
}
```

在全局去除元素的 `outline`，会导致控件元素被 `focus` 时候UI完全没有任何改变，用户根本就无法知道现在到底哪个元素是处于 `focus` 状态，键盘可访问性几乎就是 `0`，在我看看是非常不专业的做法。

其实我们只要平时注意HTML语义化，例如按钮不要使用 ``，`<div>` 等标签，不要重置 `outline`，基本上键盘可访问性就已经及格了。

若想再进一步，例如一些交互效果也能键盘访问，则需要做一些其他的额外工作了，一般是使用CSS `:focus`伪类或者JS focus事件提高网站键盘可访问性。

眼见为实，您可以狠狠的点击这里：[label标签和表单元素之间的键盘可访问性demo](#)

小tips: `:focus` 伪类和 `outline` 都是IE8浏览器开始支持的。

三、CSS hover显示隐藏内容的键盘可访问性

举个例子，我列表元素信息很多，为了防止视觉干扰，一些操作按钮在鼠标hover当前列表的时候才显示，如下图所示：



很多小伙伴在实现的时候，并没有考虑很多，就直接使用 `display:none` 隐藏，或者 `visibility:hidden` 隐藏，于是会导致隐藏的控件元素压根没法通过键盘让其显示，因为这两种隐藏方式会让元素无法被 `focus`，那该怎么办呢？可以试试使用透明度 `opacity` 控制内容的显隐，于是，我们就可以通过 `:focus` 伪类让按钮 `focus` 时候可见，例如：

```
tr .btn {
    opacity: 0;
    filter: alpha(opacity=0);
}
tr:hover .btn,
tr .btn:focus {
    opacity: 1;
    filter: none;
}
```

效果示意图如下：



眼见为实，您可以狠狠的点击这里：[CSS hover显示元素交互的的键盘可访问性demo](#)

小tips: 先点击右侧展示空白区域，在Tab键索引，可以快速 `focus` 到目标元素。

四、CSS hover显示下拉内容的键盘可访问性

经常有下拉内容需要点击或者鼠标hover时候显示。

如果是这类交互，同时希望键盘可访问性良好，那该怎么办呢？

首先一定要有键盘可访问的触发源，也就是无论是点击区还是hover区域，一定要有个 `<a>` 标签，或者原生按钮，或者设置了 `tabindex` 的普通元素。

然后，嗯……情况还有点小复杂，为了便于理解，我们把交互形式和实现原理，分为下面四类：

1. 列表HTML结构依赖，使用CSS定位，hover显示；
2. 列表HTML结构依赖，使用CSS定位，click显示；
3. 列表HTML结构不依赖，使用JS定位，hover显示；

4. 列表HTML结构不依赖，使用JS定位，click显示；

例如，导航上的二级菜单常使用CSS进行定位，对HTML结构有要求；而搜索的自动下拉提示列表则几乎都使用JS进行定位，列表直接创建于 `<body>` 标签下，对HTML结构无依赖。

针对上面四种情况，我需要额外进行的处理分别是：

1. 增加:focus控制；
2. 无需额外处理；
3. 增加JS `focus` 事件处理，处理细节同 `mouseenter` ；
4. 无需额外处理；

拿第一种情况举例，如下简易的hover显示列表HTML和CSS：

```
<a href="javascript:" class="trigger">更多操作▼</a>
<div class="list">
  <a href="javascript:">编辑</a>
  <a href="javascript:">删除</a>
</div>
```

CSS交互相关代码如下：

```
.list {
  position: absolute;
  visibility: hidden;
}
.trigger:hover + .list,
.trigger:focus + .list {
  visibility: visible;
}
```

其中，上面CSS代码标红高亮的就是增加的 `:focus` 控制，于是当 `focus` 我们的“更多操作”对应的 `<a>` 元素的时候，列表也会显示出来，完全用不到鼠标出场。



然而，事情并没有结束，通过 `:focus` 让原本隐藏的列表出现仅仅是第一步，因为列表中的元素往往也是可以 `focus` 的，链接或者操作按钮之类的，该如何使用键盘访问这些元素呢！如果还是使用 `Tab` 键一定会出问题，因为触发源失焦，列表就会隐藏，而且DOM顺序并不能保证，尤其当下拉浮层内容是使用JS创建和定位的时候。

我们可以从下拉框中寻找灵感，也就是当这些浮层显示的时候，通过上下左右键进行控制，但是，对于普通元素而言，上下键是不会自动选中控件元素的，只会滚动页面，面对这种场景，我们需要一段额外的脚本对 `:focus` 出现的下拉内容进行索引控制和回车支持。

增强键盘访问的JS代码

我刚花了一会儿时间把这段JS写好了，很短，包括注释100行不到，我已经放到本站的微码中去了，您可以狠狠地点击这里：[微码-键盘可访问性之focus显示的下拉或浮层的索引和回车支持](#)

原生JS书写，兼容IE9+浏览器，IE8浏览器调用不会报错，放心引入。

CSS和HTML需要一点点的配合，下拉或浮层中按钮或链接的选中样式是通过增加类名 `outline` 实现的，所以需要：

```
.outline {
  outline: 1px dotted Highlight;
  outline: 5px auto -webkit-focus-ring-color;
}
```

HTML这块，JS需要知道当前 `focus` 元素对应的下拉或浮层是哪一个，因此，需要手动标记下，语法为，使用 `target` 或者 `data-target` 属性，属性值和下拉或浮层元素的 `id` 保持一致就可以。例如：

```
<a href="javascript:" class="trigger" data-target="list">更多操作▼</a>
<div id="list" class="list">
  <a href="javascript:">编辑</a>
  <a href="javascript:">删除</a>
</div>
```

正如上面示意的，通过 `data-target="list"` 让触发元素和我们的列表发生关联。很简单的处理，然后键盘访问效果就达成了。

您可以狠狠地点击这里：[CSS hover显示元素交互的的键盘可访问性demo](#)

例如，focus按钮，然后往下键按两下，就会这样：



然后再回车，就相当于点击了这个 `focus` 选中元素，什么按钮事件啊，链接跳转都可以预期执行。于是，我们从头到尾的交互都不需要鼠标参与了，键盘可访问性一级棒！

最后总结下就是：

1. CSS定位增加 `:focus` 显示，JS定位增加 `focus` 事件显示；
2. 引入下拉和浮层索引和回车支持JS；
3. 粘贴 `.outline` 样式，以及简单的HTML关联属性设置；

即可让网站的键盘可访问性达到顶级，成本又不高，可以说是高品质网站必备良药。

五、更多关于键盘可访问性

实际上，键盘可访问性的价值就算用户的鼠标是正常的，也可以大大地体现。

举个例子，输入框的回车执行，虽然说小白用户还是习惯点按钮，但是对于一些资深网虫用户而言，如果你的表单例如搜索居然不支持回车功能，一定会在内心咒骂，这个产品体验做的真差。

或者说搜索输入框的自动提示下拉不支持上下键切换，也会让用户不爽的，明明可以键盘搞定的事情，非要逼我让我的尊手移到不知何方的鼠标上去，肯定心里难受啊。

但是对于本文所介绍的键盘可访问性支持，你的网站就算不支持，用户也不会有什么意见的，不是说用户没这个需求，而是说99%的网站对键盘支持都是这幅尿性，普通用户自然会认为这是常态，这是局限，自然也就不会在心里产生咒骂。

然而，有些需求虽然用户没有提出来，并表示用户不需要这些东西，就像iPhone一样，把好的东西出现之后，用户才会发现，原来我之前认为无所谓的事情，实际上是很糟糕的，体验很差的。

如果我们把网站作为一个产品，就键盘可访问性这一块而言，实际上是一个可以产生深远影响的功能点，因为就这么一小步，就甩了99%网站一条街。

第一步，对整站的键盘可访问性以及无障碍访问做良好的适配和兼容；第二步，通过运营手段对用户加

以宣传和推广（虐惯的用户是不会主动尝试是否键盘支持的）；第三步，当用户使用得非常爽之后，再使用同类产品，你一定会觉得其他产品做的真**的，产品的口碑就会慢慢起来，对整个企业品牌形象，甚至运营收入提高都会产生广泛而积极的影响。

所谓用户体验，我就是把这些跟用户打交道的事情做到极致嘛！

趁今天时间多，我再扯远一点。

本文的键盘可访问性，无可厚非，一定是非常有价值的用户体验提升点。我们可以仔细看一下，我们从头到尾都使用了哪些技术？

案例1主要是HTML行为和CSS，案例2纯CSS，案例3是CSS和JS。

请问有没有使用框架？有没有使用ES6？有没有使用Vue？有没有使用Node.js？以及其他什么乱七八糟的新鲜事物？没有，都没有，就是简单的HTML，CSS和JS。

但是这个简单，只是你看上去的简单，若是安排一个让网站支持键盘访问的需求，那些HTML和CSS自我感觉还不错的人不一定弄的出来。差别在哪里呢，就是基础。看到的“简单”实际上是“简-繁-简”后的简单。只有当你基础知识足够扎实和深入之后，你才会知道有哪些解决之道，以及哪些才是最好的解决之道。

很多人迷茫，前端变化很快，不知道学什么。本文案例其实可以做一点解答：如果你希望你的技术日后有所成，希望以后有机会去进BAT团队做海量用户的产品，最重要的不是会使用Vue.js之类的上层的東西，而是你的前端基本功，一定要有扎实的前端基础知识，并在这个基础上不断纵向横向进步，学会通过扎实的基础创造东西，而不是使用东西。

从我去年“[找到适合自己的前端发展方向](#)”这篇招聘文章发表到现在，收到的简历已经有两三百封了，有很多人会摆上很多Vue, React, AngularJS, Sass等等之类的关键字，跟你们讲，在我这里，鸟用都没有，反而减分，其实更希望看到的是扎实的基本功和潜力，说实话，就算你JS一点都不会，只有HTML和CSS基本功足够好，我们也会要你的，不过可惜的是，从各种作品代码或者笔试结果来看，大多数人就是一个简简单单，普普通通的写页面的，根本就意识不到自己的问题所在，我页面效果都实现了呀，兼容性也不错啊，我觉得我CSS和HTML蛮好的呀！入门简单的东西反而限制了日后的成长，困于庐山之中。

偏体验方向的前端，一是关注点不仅限于功能实现，二是CSS和HTML一定要扎实，JS基础是强力加分项，三是积极向上，以及项目推动力等职业方面的东西。目的就是希望可以给整个集团产品带来类似“键盘可访问性”这类前端这块体验上的提升，这些提升并不需要什么高深的技术，需要的是关注点以及扎实的基础，而这些工作才是原本前端这个职位的定义和职责所在。

偏开发的前端好招，偏设计的前端好难招，难道是被整个行业大环境给带坏了？

《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](#)。



猜你喜欢

- [span与a元素的键盘聚焦性以及键盘点击性研究](#)
- [借助HTML5 details,summary无JS实现各种交互效果](#)
- [翻译-盲人如何使用互联网的8个误区](#)
- [js下拉菜单实现与可访问性问题的一些思考](#)
- [HTML tabindex属性与web网页键盘无障碍访问](#)
- [HTML input type=file文件选择表单元素二三事](#)
- [实力科普：为什么浮层或弹框一定要有叉叉关闭按钮？](#)
- [HTML可访问性fieldset,legend元素及CSS布局应用](#)
- [说说HTML5中label标签的可访问性问题](#)
- [CSS radio/checkbox单复选框元素显隐技术](#)
- [基于VoiceOver的移动web站无障碍访问实战](#)

分享到：        0

标签：[focus](#), [form](#), [label](#), [outline](#), [tabindex](#), [可访问性](#), [键盘事件](#)

发表评论（目前11条评论）

<input type="text"/>	名称 (必须)
<input type="text"/>	邮件地址(不会被公开) (必须)
<input type="text"/>	网站
<div></div>	
<input type="button" value="提交评论"/>	

1.

手机端键盘弹出页面上移动会出现一点卡顿说道：

2017年06月30日 14:31

<https://aq.yuewen.com/welcome/validateuser>

这个页面在输入验证码的时候，手机键盘弹出，页面上移动的过程中出现卡顿，请问这是什么原因导致的，测试机版本是安卓5.1.1，ios不会出现这个问题，求大神解惑

[回复](#)


2.

哦了说道：

2017年05月10日 15:45

刻盘按时

[回复](#)


3.

落叶树说道：

2017年05月4日 23:54



基础很重要，非常同意，框架这些东西只要用到，花些时间总能熟练使用，最终要解决的还是业务场景，优化用户体验。

[回复](#)

4. **cshenger**说道：

2017年04月28日 20:09

基础是前进的基石，所以，贵公司还要人不，就是本人只有大专学历，哎~~

[回复](#)



5. **严文彬**说道：

2017年04月28日 09:33

目前写过几个电视端的页面，就是纯键盘操作事件，每个区域每个按钮都要判断，有点复杂，还好正常完成功能了。之前想过能否直接通过tab的方式直接获取表单元素，但是只能依次访问，无法实现用方向键取得最近的按钮。

[回复](#)



哦了说道：

2017年05月10日 15:46

54125

[回复](#)



6. **yqshuai**说道：

2017年04月26日 17:21

看了这么多博主的文章，最佩服的还是博主的灵魂画风

[回复](#)



7. **afei**说道：

2017年04月26日 11:09

赞同博主的看法，基础很重要，高级框架也要按需加载才是

[回复](#)



8. **小小林**说道：

2017年04月25日 15:43

白色背景看起来费眼睛，建议网站背景用易读性的颜色

[回复](#)



9. **meepo**说道：

2017年04月24日 08:38

好设计+超强编码能力 = 不存在

[回复](#)



10. **cheeseyu**说道：

2017年04月23日 23:46

任何东西学精了都别有一番奇妙展现出来。

[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 (190)
- » 未来必热: SVG Sprite技术介绍 (119)
- » 粉丝群第1期CSS小测点评与答疑 (115)
- » HTML5终极备忘大全 (图片版+文字版) (93)
- » 让所有浏览器支持HTML5 video视频标签 (86)
- » Selectivizr-让IE6~8支持CSS3伪类和属性选择器 (82)
- » CSS3下的147个颜色名称及对应颜色值 (79)
- » 视区相关单位vw, vh..简介以及可实际应用场景 (76)
- » 写给自己看的display: flex布局教程 (76)
- » 小tips: 纯CSS实现打字动画效果 (76)



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 scrollToView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

- span与a元素的键盘聚焦性以及键盘点击性研究
- 借助HTML5 details,summary无JS实现各种交互效果
- 翻译-盲人如何使用互联网的8个误区
- js下拉菜单实现与可访问性问题的一些思考
- HTML tabindex属性与web网页键盘无障碍访问
- HTML input type=file文件选择表单元素二三事
- 实力科普: 为什么浮层或弹框一定要有叉叉关闭按钮?
- HTML可访问性fieldset,legend元素及CSS布局应用
- 说说HTML5中label标签的可访问性问题
- CSS radio/checkbox单复选框元素显隐技术
- 基于VoiceOver的移动web站无障碍访问实战

