HTML tabindex属性与web网页键盘无障碍访问

这篇文章发布于 2017年05月13日，星期六，23:04，归类于 HTML相关。 阅读 12289 次, 今日 10 次 13 条评论

HTML虽然入门简单，但是，要想日后深入，却要花费非常大量的时间，因为一些看似非常简单的属性，实际上牵扯到另外一个完整领域的知识。例如本文要介绍的这属性 `tabindex` 和下一篇要介绍的 `accesskey` ，与web网页无障碍访问息息相关，而且是键盘访问领域的。

这些属性不仅可以触发浏览器层面的行为，本身对HTML的元素的交互特性甚至UI表现都会有影响。

一、HTML tabindex属性的原本作用，行为与表现

HTML `tabindex` 属性是一个“全局属性”，所谓“全局属性”，可以理解为“公共汽车属性”，也就是所有HTML标签都可以用的属性，比方说 `id` ， `class` 属性等。

HTML `tabindex` 属性是一个非常老的属性，浏览器支持多年，基本特性没有兼容性问题，老少皆宜，童叟无欺，放心使用。

HTML `tabindex` 属性是一个与键盘访问行为息息相关的属性，或者换种说法吧，是个用户体验息息相关的属性。平常我们可能感觉不到它的价值，但是一旦我们的鼠标坏掉了或者没电了，我们就只能使用键盘。亦或者在电视机上，或者投影设备上访问我们的网页的时候，我们只能使用遥控器。就算设备都完全正常，对于资深用户而言，键盘访问可以大大提高我们的使用效率，例如表单回车提交，下拉列表的上下键选取等，就像《全职高手》里面的叶修一样，用鼠标用键盘，尤其键盘手速玩的飞起。

下面来简单解释一下，为什么 `tabindex` 被称作 `tabindex` ？

`tabindex` 其实可以看成是一个单词组合，就是 `tab + index` ， `tab` 指的就是键盘上的Tab键，如下图（本人键盘实拍，请忽略岁月的痕迹）：

仔细观察可以发现Tab键上除了有字母Tab还有左右箭头指向图形，其实已经隐隐透露出了这个键的作用，对 `focusable` 元素进行索引切换。

至于 `index` 就更好理解了，我们遍历数组的时候，或者数据库中都会有 `index` 这个东西，表示“索引”，一般都是数值，表示当前内容是第几个第几个第几个。

所以 `tabindex` 表示的意思就是“元素使用Tab键进行focus时候的顺序值”，因此，理论上， `tabindex` 属性值只能是数值。但是实际的表现却有差异，在Firefox浏览器下， `tabindex` 属性值缺省，或者属性值非数值，例如 `tabindex="xxx"` ，其行为表现等同于设置了 `tabindex="-1"` ，而IE和Chrome则忽

略该属性。

`tabindex` 支持正值和负值，但是对负值而言，没有任何理由使用 `-1` 以外的其它属性值。这并不是说其它负值就不起作用，，作用还是有的，只是没有意义，为什么这么说呢？

当一个元素设置 `tabindex` 属性值为 `-1` 的时候，元素会变得 `focusable` ，所谓 `focusable` 指的是元素可以被鼠标或者JS `focus` ，在Chrome浏览器下表现为会有 `outline` 发光效果，IE浏览器下是虚框，同时能够响应 `focus` 事件。但是，却不能被键盘 `focus` 。

```
tabindex
```

这种鼠标可以 `focus` ，但是键盘却不能 `focus` 的状态，只要 `tabindex` 属性值为负值，因此， `tabindex="-2"` 和 `tabindex="-1"` 没有任何本质区别，既然如此，那就没有必要冒险在玩类似 `tabindex="-2"` 的新花样，万一哪天浏览器认为这种用法是不合法的呢？在JavaScript中，字符串和数组都有 `indexOf` 方法，当无法匹配的时候返回的值就是 `-1` ，对吧，足够了，并没有 `-2` 之类的返回值，和 `tabindex` 的路数其实是一致的。

在MDN文档上有这么一句话：

> Note: The maximum value for tabIndex should not exceed 32767. If not specified, it takes the default value -1.

我翻译下就是： `tabindex` 属性值的最大值不能超过 `32767` 。如果 `tabindex` 缺省，则使用默认值 `-1` 。

其中后半句的表述似乎仅Firefox浏览器符合，至于前半句……

我特意在我电脑的各个浏览器下测试了一下，发现当设置的 `tabindex` 超过 `32767` 到时候，不同浏览器下的表现是不一样的：

- 在IE浏览器下，如果 `tabindex` 值正好超出最大限制，也就是 `tabindex="32768"` 的时候， `tabindex` 属性值无效被忽略；但是如果值继续往上超出，例如 `tabindex="32768"` 或者 `tabindex="123456"` ，其行为表现和 `tabindex="-1"` 是一致的，可以被鼠标 `focus` 无法被键盘 `focus` ，真是非常怪异的表现。
- 在Firefox和Chrome浏览器下，如果 `tabindex` 值正好超出最大限制，就跟完全没事似的，可以鼠标 `focus` 也能键盘 `focus` 。出现这种现象有两种可能：一是Firefox和Chrome没有 `32767` 的限制，二是有 `32767` 限制，但是浏览器按照最大值 `32767` 进行解析了。究竟是哪一种可能呢？经过我的测试，发现是第一种可能，Firefox和Chrome没有 `32767` 的限制，或者有限制，但是绝对比 `32767` 要大。测试方法很简单：

  ```
  <div tabindex="32769">tabindex="32769"</div>
  <div tabindex="32768">tabindex="32768"</div>
  <div tabindex="32767">tabindex="32767"</div>
  <div tabindex="32766">tabindex="32766"</div>
  ```

  使用Tab键进行 `focus` 切换，看顺序。最后测试结果是 `outline` 是从下往上依次出现的。

认识tabindex=”0″

我先抛一个问题问问大家，如下两个 `<div>` ，请问，当使用Tab键进行索引的时候，哪个先被 `focus` ？哪个后被 `focus` ？

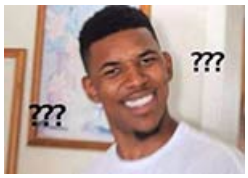
```
<div tabindex="0">tabindex="0"</div>
```

```
<div tabindex="1">tabindex="1"</div>
```

//zxx: 假设三分钟过去了…

一定有很多很多小伙伴会认为第一个 `<div>` 首先被 `focus`，然后才是第二个 `<div>`，因为 `0` 比 `1` 小，索引的顺序也自然更加的靠前。

但实际上，所有浏览器都是第二个 `<div>` 先被 `focus`，然后才是第一个 `<div>` ！



不要摆出上面的表情，事实就是如此。

`tabindex` 属性的键盘索引顺序其实是从数值 `1` 开始的，不是 `0`， `1` 索引顺序是最靠前的。也就是说哪怕你在页面的最底部、文档流的最后一个元素设置了 `tabindex="1"`，当按下Tab键的时候，首先 `focus` 就是这最后一个元素。

那 `tabindex="0"` 又是怎么回事呢?

前文说过，元素设置 `tabindex="-1"`，可以鼠标和JS可以 `focus`，但键盘不能 `focus`； `tabindex="0"` 和 `tabindex="-1"` 的唯一区别就是键盘也能 `focus`，索引的顺序没有任何的变化。或者你可以这么理解， `<div>` 设置了 `tabindex="0"`，从键盘访问的角度来讲，相对于 `<div>` 元素变成了 `<button>` 元素。

因此，实际上，我们是可以使用 `<div>` 或者 `<span>` 元素模拟按钮的，但是千万不能忘记设置 `tabindex` 属性，如下示意:

```
<div class="button" tabindex="0" role="button">按钮</div>
```

如果你是 `tabindex` 属性重度使用患者，免不了会遇到 `tabindex` 属性值一样的情况，这时候的索引属性是怎样的呢?

此时顺序是根据DOM元素在文档中的位置决定的，越靠前越外层的元素索引顺序更高。

好，以上就是HTML `tabindex` 等一些基础知识。我经常跟新人前端小伙伴答疑的时候讲，当务之急不用追风逐影，而是想办法扎实基础。比如说上面一整节都是基础知识，只有你的基础知识扎实了，你才能高屋建瓴，举一反三，并在实际项目开发中自如应用。

二、HTML tabindex的使用场景

在通常的web开发中，你只要不犯一些HTML常识性错误，例如使用 `<div>` 或者 `<span>` 元素作为按钮，或者全局去除 `outline` 的错误， `tabindex` 属性就算不出场，网站的键盘无障碍访问支持已经及格了。

如果想更进一步，可以增加CSS `:focus` 或JS `focus` 事件的交互支持，具体可参考不久前我的这篇文章：“CSS :focus伪类JS focus事件提高网站键盘可访问性”。

但是仅仅额外增加事件支持，并不能让网站 `100%` 键盘支持，或者说可以 `100%` 支持，但交互和体验不一定完美，此时可能就要轮到 `tabindex` 属性出马了！

场景一：`hover` `<div>`显示下拉键盘支持

这是网页中非常常见的一种交互，当我们内容比较静态的时候，常常直接使用CSS `:hover`伪类实现内容的显隐效果，通常，我们都会使用一个大大的`<div>`元素包在外面，从功能上讲，似乎满足了我们的需求，但实际上，对键盘以及其它辅助设备的访问却不友好，因为默认隐藏的下拉内容只能通过鼠标操作显示，键盘永远显示不出来，自然屏幕阅读器等辅助设备也不能读取，此时，就轮到`tabindex`属性来提高我们站点的可访问性了。就是给`<div>`加一个`tabindex="0"`，对吧，小手一抖，体验就有，皆大欢喜，何乐不为！然后再加个`:focus`伪类显示下拉内容就好了！都是举手之劳的事情。

场景二：对网站模块进行自定义的键盘支持

写文章这么多年，第一次拿新浪微博做正面的例子。虽然说新浪微博PC端最重要的微博页，加载慢性能差，但是，数年前却专门增加了自定义的键盘访问支持，比方说按下键盘上的J键，则会自动索引每条微博信息，并使用JS进行`focus`高亮，如下截图所示：

其原理就是设置`tabindex="-1"`然后使用JS进行`focus`，如下JS示意：

```
div.setAttribute('tabindex', '-1');
div.focus();
```

由于设置了`tabindex="-1"`的元素鼠标点击可以`focus`并`outline`轮廓显示，因此，这里的`tabindex="-1"`并不是默认就有，而是按下对应的快捷键后临时动态添加的。

场景三：临时改变页面索引起始位置

当我们点击按钮并显示一个弹框的时候，Tab键的索引起始位置应该是从弹框元素开始的，但是如果我们不做任何处理，索引起始位置还是弹框背后的页面，此时想要通过Tab键一个一个索引到弹框元素，估计天都已经黑了，很显然，为了达到完美的键盘交互体验，我们就需要额外做点事情。

1. 给弹框容器元素设置`tabindex="-1"`；
2. 弹框显示的时候容器元素`DOM.focus()`使其获取焦点；
3. 容器元素CSS设置`outline:none`；

简简单单三步即可大功告成。

眼见为实，您可以狠狠的点击这里：弹框出现与Tab键索引起始位置变化demo

demo页面中，点击按钮出现弹框后，你再按一下Tab键，会发现第一个被`focus`的是关闭按钮，如下截图：

依次Tab，就可以`focus`弹框内所有`focusable`元素了。

三、HTML tabindex结束语

我突然意识到一个问题，当我翻阅我很多年前写的文章的时候，会发现现在的我没有以前“嘻嘻哈哈”了，换句话说，那些野生的棱角和个性已经在不知不觉中被慢慢磨掉了，而且想回去也回去不了了，心态已经变了，更加沉淀，更加稳重，哎，其实是变得更像大叔了。

想了想，其实也不是什么问题，还是顺其自然吧，没有必要非要做作成什么样子。

如果大家足够敏锐，应该会发现，我前两篇文章包括这篇以及下一篇其实都是与键盘无障碍访问相关的，说明我最近学习和关注点在这方面。

经常会见到私信或者留言说“可不可以写写某某方面的文章”，我总是委婉拒绝的。个人站点的好处就在于自由，以后我写小说肯定也是独立运作，否则动不动就被别编辑被读者催稿，那就压力大了！其实类似的，我写技术文章从来不带有功利性，不是说什么东西火就写什么，也不是说什么别人让我写什么就写什么，仅仅是纯粹记录自己学习与工作中研究、总结和积累，在自我成长同时帮助他人成长，是一件很棒很自在的事情，否则也不可能一直坚持这么多年。

好了，就这些，感谢阅读，欢迎交流！

> 《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。

« HTML <area><map>标签及在实际开发中的应用

HTML accesskey属性与web自定义键盘快捷访问 »

猜你喜欢

- 借助HTML5 details,summary无JS实现各种交互效果
- 实力科普：为什么浮层或弹框一定要有叉叉关闭按钮?
- HTML accesskey属性与web自定义键盘快捷访问
- CSS :focus伪类JS focus事件提高网站键盘可访问性
- 是时候好好安利下LuLu UI框架了！
- span与a元素的键盘聚焦性以及键盘点击性研究
- 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
- 翻译-盲人如何使用互联网的8个误区
- 请使用千位分隔符(逗号)表示web网页中的大数字

标签： focus, html, outline, tabindex, 无障碍网页应用, 键盘

发表评论（目前13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. 关于taxindex设置为负值时说道：
2018年08月3日 11:17

当链接元素或表单元素，设置tabindex的值为-1的时候，也能通过键盘，让她获焦点。

回复

> 张 鑫旭说道：
> 2018年08月3日 11:32
>
> 确定？
>
> 回复

2. 墨轩mo说道：
2017年08月21日 14:58

嗯嗯，get到了，又可以稍稍修正自己的毛病了

回复

3. 晨曦说道：
2017年07月14日 14:02

今天看bootstrap的代码，才发现有这个东西，get到了

回复

4. mgser说道：
2017年06月23日 16:50

tabindex 记得还有一个很好用的地方，就是在于能够将任何元素变成跟表单元素一样的表现（效果也如同上面focus），这个属性加在div上可以模拟select等表单元素，实现聚焦显示，失焦隐藏的效果

回复

> wells说道：
> 2018年08月23日 10:48
>
> pc有效，移动端不行
>
> 回复

5. Lee说道：
2017年05月24日 11:22

我也发现你不再嘻嘻哈哈了，大叔【难过】

回复

6. **yuan说道：**
2017年05月15日 14:27

我自己试了一下，我们点击弹窗之后，tab的选取会从点击的按钮开始往下，直接把弹窗元素放在点击的按钮后面，就会直接索引到弹窗元素，当然这样就不能随意排版了

回复

7. **xiaojunjor说道：**
2017年05月15日 14:21

“在Firefox浏览器下，tabindex属性值缺省，或者属性值非数值，例如tabindex=”xxx”，其行为表现等同于设置了tabindex=”-1″，而IE和Chrome则忽略该属性。”

旭哥，我在FF里测试结果是，默认可以获取焦点的元素（比如input，select等）表现为忽略，而默认不可以获取焦点的元素（比如div，form等）表现等同于设置为-1

回复

8. **xiaojunjor说道：**
2017年05月15日 10:24

真的细，自愧不如，像前辈学习

回复

9. 严文彬说道：
2017年05月15日 09:40

目前在做电视的网页，都是全键盘事件，完全通过js模拟实现控制焦点的，一直在想有没有类似使用tab键的那种写法?

回复

> 神秘博士说道：
> 2017年05月24日 16:39
>
> 我之前做了个安卓电视的应用（cordova打包网页做的混合app），就是利用的tabindex，才支持遥控器的方向键操控的
>
> 回复

10. 严文彬说道：
2017年05月15日 09:37

这个只能通过tab键依次查找吧，可不可以通过方向键来控制呢（好像不行）

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 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

- 借助HTML5 details,summary无JS实现各种交互效果
- 实力科普：为什么浮层或弹框一定要有叉叉关闭按钮?
- HTML accesskey属性与web自定义键盘快捷访问
- CSS :focus伪类JS focus事件提高网站键盘可访问性
- 是时候好好安利下LuLu UI框架了！
- span与a元素的键盘聚焦性以及键盘点击性研究
- 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
- 翻译-盲人如何使用互联网的8个误区
- 请使用千位分隔符(逗号)表示web网页中的大数字