

# 基于VoiceOver的移动web站无障碍访问实战

这篇文章发布于 2017年01月20日, 星期五, 23:26, 归类于 [HTML相关](#)。阅读 12607 次, 今日 7 次 [8 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=5893>

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、了解什么是“无障碍访问”

首先要了解下什么是“无障碍访问”。关于“无障碍访问”大家或多或少都有所耳闻, 于是可能会产生这么一个误解, 认为所谓“无障碍访问”就是让势力有缺陷的人可以自如使用访问自己的产品。实际上, “无障碍访问”涵盖的范围要广泛的多。

所谓“无障碍访问”, 指的是各类设备均可以无障碍访问。例如鼠标、键盘、读屏软件或设备等。

所以, 大家不妨思考这么几个问题?

1. 如果用户键盘坏了, 只有鼠标贵站还能正常访问吗?
2. 如果鼠标坏了, 只使用键盘贵站还能正常访问吗?
3. 如果鼠标和键盘都坏了, 只有麦克风, 贵站还能正常访问吗?
4. 如果用户视力障碍, 通过声音识别, 贵站还能正常访问吗?

以上这种场景的适配和支持就属于“无障碍访问”技术范畴。于是, 请继续思考:

1. 为什么按钮习惯使用 `<a>` 标签?
2. 为什么 `<input>` focus时候会外发光?

这其实涉及到“键盘可访问性”, 当用户鼠标出现异常, 比方说我的iMac鼠标多次忘记充电, 只能键盘操作, 此时往往都是使用Tab键按顺序(如果没有设置 `tabindex`)让控件元素获取焦点, 并默认 `outline` 高亮。

所以, 对于to C也就是面向大众用户的网站, 使用如下的CSS是非常业余的:

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

但是, 有小伙伴会吐槽了: “我也不想去掉啊, 但是设计师看不惯默认的发光效果啊!”

如果是对效果不满意, 那好办, 改成设计师喜欢的 `focus` 效果就好了, 例如:

```
input {  
  outline: 0, none;  
}  
input:focus {  
  border-color: HighLight;  
}
```

还有就是“按钮习惯使用 `<a>` 标签”的问题, 按钮使用 `<button>` 是语义化和可访问性最好的, 但是, `<button>` 的UI兼容性不太好, 即使现代浏览器下也是如此。因为需要使用其他非替换元素标签, 如果

单看UI样式，使用 `<span>`，`<div>` 标签都是可以的，然而，这两个标签的问题在于，默认状态下，是无法被键盘 `focus` 的，因此，我们就使用 `<a>` 标签来写按钮，兼顾UI兼容和键盘可访问性。

标签（注意 `href` 属性必须有）。但是，我要转折了，毕竟 `<a>` 标签语义是链接，因此，在屏幕阅读设备或读屏软件那里会被读做“链接”，而非“按钮”，这对于用户而言显然就是误导，那该怎么办呢？试试增加一个 `role="button"`：

```
<a href role="button">
```

这里出现的 `role` 属性就是[ARIA HTML无障碍访问](#)中最常用的属性之一。

## 二、了解无障碍访问中的ARIA

ARIA全称“Accessible Rich Internet Applications（可访问的富互联网应用）”，在当下讨论ARIA基本上都是为读屏设备或软件，而读屏设备或软件的使用者多是视力障碍人员，因此，ARIA约定俗成就成了对视障用户的无障碍访问支持技术的代称了。

首先，了解下常见的读屏软件，如下列表：

移动端：

- Android: TalkBack
- Android: Funtouch
- iPhone: VoiceOver

桌面端：

- Windows: NVDA, JAWS
- Chrome: ChromeVox
- OSX: VoiceOver

由于本人精力有限，因此，目前仅试验了移动端-iPhone-VoiceOver，iOS 10.\*.\*版本下的ARIA支持情况。对于其他软件，我猜想应该都是类似的，毕竟有规范在那里。

ARIA规范还在不断更新中，且相当积极，因此，规范往往会比软件的支持超前，因此，本文下面的内容不涉及新的属性特性。

ARIA总共有3大部分组成，如下：

1. `role`属性值
2. `aria`属性
3. `aria`状态属性

我之前的文章“[WAI-ARIA无障碍网页应用属性完全展示](#)”有非常详尽的翻译展示，我这里列举一些常用的：

**role属性值**

```
role="tab",
role="button",
role="radio",
role="checkbox",
role="link",
...
```

## aria属性

```
aria-haspopup,  
aria-label,  
aria-owns  
...
```

## aria状态属性

```
aria-checked,  
aria-checked,,  
aria-selected,  
aria-expanded,,,  
aria-hidden,,  
aria-invalid,  
...
```

好，热身完毕，下面开始本文的重点了，基于VoiceOver的移动web站无障碍访问实战。

## 三、基于VoiceOver的移动web站无障碍访问实战

### 如何开启iPhone的VoiceOver?

最快捷的是使用siri，“打开VoiceOver”，如果siri卖萌说不知道VoiceOver是什么鬼？那就手动开始，路径为：设置→通用→辅助功能→VoiceOver（在第一个，见下图）



在iPhone上VoiceOver开始后，操作方式有个根本的变化：

1. `touchstart` 或 `touchmove` 行为完全变成了内容识别与读取，没有任何web交互行为的发生！
2. 如果你想点击某元素，必须先 `touchstart` 或 `touchmove` 选中，然后再双击，直接双击是没有用

的；

3. 如果你想滚动，三指滑动；

4. 如果你想拖动滑块，先 `touchstart` 或 `touchmove` 选中，然后连续轻触并滑动（手指不要抬起）；

所以，很多人不小心打开VoiceOver后发现关不了了，就是因为还是老的操作习惯。想要关闭，最快捷sir i；如果只卖萌不行动，那就手动按部就班关闭：（轻触，再双击）设置→（轻触，再双击）通用→（轻触，再双击）辅助功能→（轻触，再双击）VoiceOver->（轻触，再双击）关闭。

## VoiceOver下无障碍访问基础特性

首先，大家应该都知道，HTML元素还可以分为非替换元素和替换元素，常见的替换元素包括 `<img>`，`<input>`，`<img>`，`<button>`，`<iframe>`，`<video>`，`<object>` 等。

除了内容可替换以及一些CSS行为差异外，在ARIA无障碍访问这一领域也是有着巨大的差异的。

### 1. 非替换元素

① `touch` 页面任何区域一定会有信息读取！

非VoiceOver状态，你点击页面空白，是不会有反应的，但是开启VoiceOver后则完全不同，`touch` 页面任何区域一定会有信息读取，包括空白区域，而轻触空白区域的读取遵循下面2个规则：

1. 就近原则；
2. 穿透规则；

所谓“就近原则”，比方说点击下图所示的位置：



结果选中和读取的内容是“免费 链接 导航 标志性内容”，哪个近读哪个。

所谓“穿透规则”，比方说点击下图所示的半透明黑色蒙层位置，请问读出来的信息是？



结果不是“信仰神国”，也不是“读至...”，读取的信息而是蒙层下面的图片列表信息！没错，直接穿透了。这个和正常浏览方式下的点击世界观是不一样的。但是，显然，此处穿透不是我们想要的，怎么避免呢？这个后面会介绍。

## ② 必须有文字内容才会读取

1. 如果元素不含文字内容，是不会读取的，例如，下图图标轻触是被直接忽略的：



相关HTML代码如下：

```
<a href>
  <i class="icon-free"></i>    <!-- VoiceOver忽略 -->
  <h4>免费</h4>
</a>
```

2. 还没完，如果我们增加 `title` 属性描述内容，那会不会读取呢？

```
<i class="icon-free" title="图标"></i>    <!-- VoiceOver读不读呢？ -->
```

答案是：依旧忽略！

3. 但是，注意，但是，有一个例外，那就是 `<a>` 元素，如果这里的 `<i>` 标签换成 `<a>`，则 `title` 属性就可以被VoiceOver宠幸。

```
<a href class="icon" title="图标"></a>    <!-- 读：图标 链接 -->
```

4. 还是 `<i>` 标签，但是，我们不是使用 `title` 属性，还是ARIA原生的规范的 `aria-label` 信息描述属性，那会不会读取呢？

```
<i class="icon-free" aria-label="图标"></i>
```

答案是：忽略，不会读取！

5. 还没完，如果我们里面写入了文字，但是是 `display:none` 隐藏的，那会不会读取呢？

```
<i class="icon-free">
  <span hidden>图标</span>
</i>
```

答案是：忽略，不会读取！

6. 还没完，如果我们里面写入了文字，但是使用的是 `visibility:hidden` 隐藏，那会不会读取呢？

```
<i class="icon-free">
  <span style="visibility:hidden;">图标</span>
</i>
```

答案是：忽略，不会读取！

7. 还没完，假设我们使用 `text-indent` 缩进让文字隐藏到屏幕之外，那总该要读取了吧？

```
<i class="icon" style="text-indent:-999px;">图标</i>
```

答案是：忽略，不会读取！

8. 还没完，假如我们的 `text-indent` 缩进不要那么猛，仅仅是图标容器外，但是在屏幕内，那会不

会读取呢？

```
<i class="icon" style="text-indent:-10%;">图标</i>
```

答案是：**会读取！**但是，选中时候的外框明显不在图标所在的位置，而是文字所在的位置。

9. 还没完，假设我们使用CSS `clip` 属性隐藏，那会不会读取呢？

```
<i class="icon">
  <span style="position:absolute;clip:rect(0,0,0,0);">
    图标
  </span>
</i>
```

答案是：**会读取！**

10. 还没完，假设我们使用绝对定位屏幕外隐藏，那会不会读取呢？

```
<i class="icon">
  <span style="position:absolute;left:-999px;">
    图标
  </span>
</i>
```

答案是：忽略，不会读取！

11. 还没完，如果是相对定位屏幕外隐藏呢？

```
<i class="icon">
  <span style="position:relative;left:-999px;">
    图标
  </span>
</i>
```

答案是：忽略，不会读取！

12. 还没完，我还有最后一口气，终极绝招，如果文字是透明的，那会不会读取呢？

```
<i class="icon" style="color:transparent;">图标</i>
```

恭喜你，**会读取！**

### ③ 文字基于内联盒子片段读取

如下一段简单常见的HTML代码：

```
<p>总共消费<output>500</output>元</p>
```

请问，`touchstart` 或 `touchmove` 该 `<p>` 元素的时候读取的信息是？

结果：要么“总共消费”，要么“五百”，要么“元”。是不会连续读取“总共消费500元”的。完全是基于内联盒子来读取的，例如这里“总共消费”和“元”是两个“匿名内联盒子”，外面有 `output` 标签的 `"500"` 是内联盒子。

但是，如果是 `<h1>` ~ `<h6>` 标题元素，则例外，例如：

```
<h6>总共消费<output>500</output>元</h6>
```

读“总共消费500元 标题级别6”。

## 2. 对于替换元素

### ① 替换元素永远不会穿透

比方说一开始的这个图：



要避免穿透，我们使用 `role` 属性将标签角色变成替换元素，例如设置 `role="button"`，如下截图：



### ② 替换元素 `title` 属性读取

还是上面那个免费图形：



1. 如果HTML是下面这样：

```
<i class="icon" role="img" title="图标"></i>
```

则 `title` 属性值是会读取的，这里会读“图标 图像”。

2. 如果角色是替换元素，则 `aria-label` 描述信息也会读取：

```
<i class="icon" role="img" aria-label="图标"></i>
```

3. `role="img"` 元素里面文本不读取，这个很好理解，原生的 `<img>` 图片标签里面是不能哟文字内容的。

```
<i class="icon" role="img">图标</i>
```

此时只会读“图像”，里面的文字“图标”被忽略了。

4. `role="button"` 非 `display` 隐藏文本均读取，也就是 `color` 透明隐藏，`font-size:0` 隐藏，`visibility:hidden` 隐藏，`text-indent` 缩进隐藏，`absolute` 屏幕外隐藏。。。都是读取的。

但是，如果是 `display:none`，则忽略：

```
<i class="icon" role="button">
  <span hidden>图标</span>    <!-- “图标”不读取 -->
</i>
```

5. 设置 `aria-hidden="true"` 元素不可读不可点。

```
<i class="icon" role="button">
  <span aria-hidden="true">图标</span>    <!-- “图标”不读取 -->
</i>
```

`aria-hidden="true"` 是ARIA无障碍处理售后非常常用的一个属性，所有的装饰性元素或者点击区域太小的元素都应该设置 `aria-hidden="true"`，避免无谓信息对用户的干扰。

6. 当内容，标题等信息同时存在时候的读取顺序是：优先内容，然后类型，然后标题，例如：

```
<i class="icon" role="button" title="示意">图标</i>
```

读：“图标，按钮”，明显停顿后，“示意”。注意，读 `title` 属性之前有个非常非常明显的停顿，停顿时间之长，感觉就像朗读人断片了一样。

7. `aria-label` 作用和文字内容类似，但是优先级更高。也就是同时存在的时候，文字内容读取会被忽略，例如：

```
<i role="button" aria-label="图标1" title="示意">图标2</i>
```

读：“图标1，按钮”，明显停顿后，“示意”。“图标2”不会读取，被忽略！

补充于2017-01-23

如果希望读内部信息，同时又自定义一些信息，可以使用 `aria-describedby`，`aria-describedby` 的行为表现和语义有些类似于 `title` 属性，`aria-describedby` 的属性值只能是元素 `id`，这非常适合保证原始标签语义同时告知辅助信息的场景，例如：

```
<h3 aria-describedby="ariaDesc1">热门小说</h3>
<span id="ariaDesc1" aria-hidden="true">编辑推荐</span>
```

8. 大段描述可以使用 `aria-labelledby` 或 `aria-describedby`，例如：

```
<i role="button" aria-labelledby="id">图标2</i>
<p hidden id="id">这是一个...图标</p>
```

读：“这是一个...图标，按钮”。

可以看到，目前描述元素即使 `display:none` 也是可以读取的。`aria-labelledby` 或 `aria-describedby` 的属性值对应描述信息元素的 `id` 属性值，可以是多个，使用空格分隔，例如：

```
<i role="button" aria-labelledby="id1 id2 id3">图标2</i>
```

9. `<nav>` 会读“导航，标志性内容”，`<header>` 读“横幅，标志性内容”，`<footer>` 读“页脚，标志性内容”。

然后，若触发的是子元素，仅第一次触发读取上面的信息。例如，导航中有两个链接，则轻触第一个的时候，读“xx, 链接，导航，标志性内容”，紧接着轻触第二个链接的时候，仅仅会读““xx, 链接””。

出乎意料的是，`<ul>`，`<ol>` 不会读列表，需要增加 `role="listbox"`，这样，触发 `<li>` 元素的时候会读“列表，第一个”。

所有 `form` 原生控件都能准确读取，包括状态。因此，一定要养成基于元素表单控件实现交互效果的习惯，控件丑没关系，透明度 `opacity:0` 覆盖处理之，例如，单选项，复选框效果等。

10. 后面这些角色设置可让断片文本连读：`role="button"`，`role="tab"`，`role="heading"`，`role="combobox"` 等。



```
<p role="button">总共消费<output>500</output>元</p>
```

读：“共消费500元，按钮”。

虽然很好地解决了多个内联元素读取“断片”的问题，但是，不合语义，明明不是按钮，你说是按钮，问题很大。

后来，经过我的各种尝试，发现了一个无语义文本连读技巧，就是使用：`role="option"`。

`option` 值源自HTML下拉列表的 `<option>` 标签。`option` 可以让里面文字连读的原理是，原生的 `<option>` 标签中只能是纯文本，会忽略一切的HTML标签，于是，设置了 `role="option"`，就会当 `<output>` 标签不存在，从而连读，并且选中的高亮框范围也更加合理。

补充于2017-01-23

使用 `role="option"` 连读的时候会可能会存在内容断句不清的情况。例如：

```
<li class="fans-li" role="option">
  <div class="rel"><aria>粉丝第1名: 我是大帅哥</aria></div>
  <div class="rel">
    <aria>粉丝等级: </aria>LV4
  </div>
</li>
```

此时朗读效果为“粉丝第1名 我是大帅哥粉丝等级 L V 4”（空格表示停顿）。实际上，“我是大帅哥粉丝等级”中间应该有明显的断句。怎么办呢？可以使用英文句号，也就是点结束符，可以产生明显断句，比全角的冒号“：”似乎还要断句明显。修改如下（“我是大帅哥后面加了个点 `.`，隐藏”）：

```
<li class="fans-li" role="option">
  <div class="rel"><aria>粉丝第1名: 我是大帅哥.</aria></div>
  <div class="rel">
    <aria>粉丝等级: </aria>LV4
  </div>
</li>
```

11. `role` 属性值设置会覆盖原始的语义，例如：

```
<a href role="option">
  总共消费<output>500</output>元
</a>
```

读：“共消费500元”，不会提示“链接”。

由于 `<option>` 标签里面只能是纯文本，因此，`role="option"` 连里面的一切语义也会忽略，于是：

```
<li role="option">
  <a href>
    总共消费<output>500</output>元
  </a>
</li>
```

读：“共消费500元”，不会提示“链接”。

如果遇到这种尴尬，可以这么处理：

```
<li role="link">
  <a href role="option">
    总共消费<output>500</output>元
  </a>
</li>
```

也就是原语义外置。此时读：“共消费500元，链接”，这下没毛病了。

### 3. 交互与aria

1. VoiceOver开启时，原来的 `touchmove` 是高成本操作，因此，需要增加详细的描述信息，否则用户根本不知道是个什么鬼？例如下面一段截图示意：



2. 复杂交互场景，或者和原生控件交互不一致的交互场景，需要添加交互场景描述，例如下面示意：



3. 需要同步改变描述和状态，例如面板展开和收起的时候，例如：



此时，先要使用 `role="menuitem"` 定义是菜单项，然后使用 `aria-expanded` 标记是展开还是收起，VoiceOver会自动根据此状态值读出对应的中文状态描述的。

```
<a href class="icon-more" title="更多" role="menuitem" aria-expanded="false"></a>
```



展开时候 `aria-expanded` 设为 `true`：

```
<a href class="icon-more" title="收起更多" role="menuitem" aria-expanded="true"></a>
```

上面例子中的 `aria-expanded` 就是ARIA中的状态属性，常用的其他属性还包括（需要配合特定的 `role` 属性值才有效）：

- `aria-expanded` 展开还是收起，菜单，自定义下拉等用的比较多。
- `aria-checked` 选中还是未选。
- `aria-selected` 选中还是未选。
- `aria-disabled` 禁用还是可用。
- `aria-hidden` 隐藏还是显示。
- `aria-invalid` 验证正确还是错误。

其中 `aria-checked` 和 `aria-selected` 含义类似，那什么时候该用什么呢？

`aria-checked` 多用在单选复选上，`aria-selected` 多用在下拉列表上，或者选项卡（`role="tab"`）上。

说到 `role="tab"` 选项卡，有一个注意点需要提一下，`role="tab"` 一定要加在平级的兄弟元素上，例如：

```
<nav>
  <h3 role="tab">
    <a href>选项卡1</a>
  </h3>
  <h3 role="tab">
    <a href>选项卡2</a>
  </h3>
  <h3 role="tab">
    <a href>选项卡3</a>
  </h3>
</nav>
```

千万不要这样：

```
<nav>
  <h3>
    <a href role="tab">选项卡1</a>
  </h3>
  <h3>
    <a href role="tab">选项卡2</a>
  </h3>
  <h3>
    <a href role="tab">选项卡3</a>
  </h3>
</nav>
```

因为后面每个选项卡会认为就一个单独的选项卡，读“共1个”，前者可以正确读“共3个”。另外，VoiceOver读取的时候不是读“选项卡”，听上去是“标间”，开房吗？

#### aria交互启示

1. 浮层半透明遮罩如果有交互，需分层，并设置 `role`，如果点击整片区域都有行为，不要取巧使用冒泡，因为在无障碍处理的时候会很啰嗦，直接使用一个透明图层覆盖，设置合适的 `role` 以及描述；
2. 浮层右上角要有关闭按钮，哪怕是透明的，千万不要信了设计师的邪，认为例如，弹框上有取消按钮，还搞个关闭干什么，不好看啰嗦。其实对于视力不好的用户，最习惯的是去屏幕右上区域找到关闭按钮。又或者浮层可以直接手指一划移出，不需要关闭按钮，那都是很自以为是的想法。如果设计师坚持，我们做开发的也需要在右上角加一个正常用户看不见的透明的关闭按钮，读屏设备是可以获取的。浮层关不掉是非常非常体验差的一件事情。
3. 自定义控件最好使用原生控件覆盖(opacity:0)，因为你这样处理后，根本不要再有任何额外的HTML或者JS支持就能有完美的无障碍访问了。
4. 交互样式改变可以尝试直接使用aria属性控制，一举两得，例如，使用 `aria-checked` 属性而不是 `.checked` 类名，可以有效降低后期无障碍支持成本。
5. 避免自定义的 `touchmove` 交互，因为开启读屏模式后，`touchmove` 交互成本很高；

#### 4. 其他ARIA无障碍支持经验补充

1. 将视觉信息转换成文字信息

例如下图：



这是个搜索列表，其中，有两个小色块，我们一看就能看明白是“标签”，而后面2个列表右侧是灰色文字，我们可以看出来是作者名。但是，对于靠触摸的盲人用户，颜色，大小包括位置都是看不到的，因此，他就很难区分这些信息是什么意思，VoiceOver会这么读：

巫妖王作者  
武侠分类  
武说谭家三十  
武侠世界独孤起步

估计用户心中是这幅表情：



因此，我们需要将视觉信息转换成文字，我的做法将必要的文字内容使用 `clip` 隐藏放在标签中：

```
aria {  
  position: absolute;  
  clip: rect(0,0,0,0);  
}
```



此时，设备读取就是：

巫妖王 标签：作者  
武侠 标签：分类  
武说 作者：谭家三十  
武侠世界 作者：独孤起步

## 2. 小区域重要标志信息合并

有些信息很重要，但是占据空间很小，或者隐蔽，此时，最好信息合并，例如，图书上的限免标志：



我们可以给“限免”标志设置 `aria-hidden="true"`，然后把该信息和图片信息合并，如下代码截图示意：



## 3. 关于SVG的无障碍访问

请问，有个SVG图标代码如下，清除轻触是否有信息读取？

```
<svg class="icon icon-arrow-1">  
  <title>返回</title>  
  <use xlink:href="#icon-arrow-1"></use>  
</svg>
```

结果，只会读“图像”，然后就没有然后了。这其实跟上面 `role="img"` 的 `<i>` 标签里面文字不读原理是一样的，就是原生图像里面是不会有文字信息的，因此，这里 `<title>` 标签中的“返回”就被忽略了。

还没完，我们看下面的HTML代码：

```
<a href>
  <svg class="icon icon-arrow-1">
    <title>返回</title>
    <use xlink:href="#icon-arrow-1"></use>
  </svg>
</a>
```

请问，会有信息读取吗？

答案是：会读“返回，链接”。因为此时SVG从图像性质变成了链接内容。

有人会问了，如果我想单纯SVG标签读取信息怎么办？可以如下处理：

```
<svg role="img" aria-label="返回">
  <title>返回</title>
  <use xlink:href="#icon-arrow-1"></use>
</svg>
```

此时，会读“图像，返回”。

## 四、不能忽略的结束语

可以看到，本文展示的信息很多，但是，实际上，只是ARIA中的一小部分。都是我在给[起点M站](#)做无障碍支持时候遇到的一些内容总结出来的，要知道，项目千千万，场景千千万，显然我只能覆盖小小部分场景，更多知识tips需要更多的积累。

并且，本文只是移动端VoiceOver的一些小知识，桌面端的VoiceOver又是如何表现怕是又有很多差异。并且，Android端的那些读屏软件行为如何，支持程度如何我都还不知道。本文的内容只是冰山一角。

ARIA的水还是很深的。

ARIA规范支持的新特性越来越多，各类读屏软件也会跟着进行支持，就好比浏览器支持新的CSS3，HTML5特性一样，但是，CSS3，HTML5特性支持如何我们很多人都知道，但是几大读屏软件对ARIA新特性的支持情况你知道吗？

可见完全就是这片领域的新人。

希望可以更多的人沉浸下来，关注这些基础的对体验友好的支持，剑走偏锋，说不定这才是你竞争力的出路所在。

另外，关于VoiceOver有一篇不错的文章，[任寒韬](#)的“[VoiceOver知多少？](#)”，大家有兴趣也可以看看，2012年写的，有些点可能过时了，比如说 `<nav>` 那个时候不读取，但是现在“导航”二次读的不要太响亮等，但是，他的测试要更系统，我则是基于实战，相互补充，比方说这张手势图就很棒：



最后，感谢阅读，行文仓促，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](#)。



« 分享一个即插即用的私藏缓动动画JS小算法

开源移动端元素拖拽惯性弹动以及下拉加载两个JS »

### 猜你喜欢

- WAI-ARIA无障碍网页应用属性完全展示
- 请使用千位分隔符(逗号)表示web网页中的大数字
- 翻译-盲人如何使用互联网的8个误区
- HTML可访问性fieldset,legend元素及CSS布局应用
- HTML5语音合成Speech Synthesis API简介
- i, b, em, strong元素在HTML5中的新语义
- span与a元素的键盘聚焦性以及键盘点击性研究
- 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
- 你到了第几层？图片式标题、按钮与隐藏文本
- absolute绝对定位的非绝对定位用法

分享到： 1

标签： aria, html, 可访问性, 屏幕阅读器, 无障碍网页应用, 语义化

### 发表评论（目前8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. Mihan说道：

2017年04月27日 19:51

查了一下 w3 官方文档，关于 ul 和 ol 标签的默认 role 属性是这样写的：

Allowed ARIA role attribute values: list role (default – do not set), directory, group, listbox, menu, menubar, presentation, tablist, toolbar or tree.

可以看到，ul 和 ol 列表的 role 应为『list』，而不应该改为『listbox』，而『list』和『listbox』是有区别的。

当ul 和 ol 标签加上『role="list"』属性的时候，VoiceOver 也会在 第一个 li 标签读『列表开头』

[回复](#)



2. 成为海贼王的男人说道：

2017年02月15日 14:24

让视力有缺陷的人吧。哈哈，找到一个错别字。

[回复](#)



3. yukun说道：

2017年02月13日 13:42

感谢博主做了这么多详尽的测试，看来VoiceOver对W3C的标准支持情况还是要摸索了才知道。

`<i class="icon-free"></i>`的例子

另一个角度看，“免费”图标语义上是分类“按钮”或者“导航链接”，而按钮和导航是aria的可交互元素。所以这里如果改成：

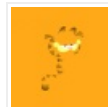
```
<ul>
<!-- 省略 -->
<li>
<button aria-label="免费"><span class="icon"></span></button>
</li>
<!-- 省略 -->
</ul>
```

就可以读取，而换用 nav 标签：

```
<nav>
<ul>
<!-- 省略 -->
<li>
<a aria-label="免费" href="..."><span class="icon"></span></a>
</li>
<!-- 省略 -->
</ul>
</nav>
```

不但可以读取，还能用屏幕阅读器的快捷键导航。

[回复](#)



4. StarBrilliant说道：

2017年01月22日 11:51

现在国内网站对可访问性做得明显不够。

是时候让前端程序员为残疾人（和行为不便的健全人）做些什么了。

另外 GitHub 页面上按 Tab 的第一次，会在页面左上角显示一个小按钮：“Jump to content”，让你省略疯狂按 Tab 才能找到页面内容的烦恼。我觉得这个功能可以借鉴。

[回复](#)



qaq在写aria说道：

2017年02月10日 14:59

我现在在做的项目里面打开voiceover，然后跳转页面发现focus到的地方是在上一个页面点击过的地方，有点类似于穿透了，然而我想让他focus到title的位置，这种方法在前端有办法实现么？Ps：focus()不起作用

[回复](#)



5. nicholasurey说道：

2017年01月22日 10:51

这下没毛病了

[回复](#)



## 6. 杨永全说道:

2017年01月21日 10:15



> ARIA规范支持的新特性越来越多，各类读屏软件也会跟着进行支持，就好比浏览器支持新的CSS3，HTML5特性一样.....

我个人的一点看法是：读屏软件对 ARIA 的支持程度主要取决于浏览器对 ARIA 的支持程度，一般情况下，读屏软件是不直接解析 HTML 的，而是使用浏览器提供的辅助接口（当然，有些情况下，读屏软件也需要直接处理 HTML 文档）。所以读屏软件支持 ARIA 的情况基本上与 CSS 和 HTML5 的情形类似，不同浏览器的支持程度也不尽相同。

[回复](#)

张鑫旭说道:

2017年01月21日 13:02



@杨永全 好的，感谢纠正！

[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)
- » [魔法攻略: 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](#) <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

- [WAI-ARIA无障碍网页应用属性完全展示](#)
- [请使用千位分隔符\(逗号\)表示web网页中的大数字](#)
- [翻译-盲人如何使用互联网的8个误区](#)
- [HTML可访问性fieldset,legend元素及CSS布局应用](#)
- [HTML5语音合成Speech Synthesis API简介](#)
- [i, b, em, strong元素在HTML5中的新语义](#)
- [span与a元素的键盘聚焦性以及键盘点击性研究](#)
- [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](#)
- [你到了第几层？图片式标题、按钮与隐藏文本](#)
- [absolute绝对定位的非绝对定位用法](#)