

Vue Conf

总结分享

Make Your Vue App Accessible @勾三股四

2018-12-10

人工智能事业部 | 通用技术部 贺世宇

Netease AI heshiyu@corp.netease.com

在我们平时使用HTML、CSS和JavaScript开发应用程序时，往往把残疾人士抛在脑后。虽然有一些辅助技术，可是这些应用程序无法提供它所能理解的语义信息。

这场分享，它在一定程度上改善了这种局面。

AGENDA 议程

- background 背景
- web standards 规范
- strategy 策略
- utils & demo 工具 & 示例

背景：为什么要有较高的可访问性？

规范：从规范层面去保证、约束有较高的可访问性？

策略：还从哪些方面去努力？

工具：从最后一公里的角度，来提高“开发较高可访问性”的效率？

原因一：

全球存在某种方面的行为\行动障碍的人群占15%

原因二：

即便正常人，身处某种情况不方便操作
(开车时，不方便触屏)
(开会时，不方便语音)






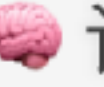
原因三：

鼠标坏了，能否使用所有（或大部分）功能？
偶尔遇到一两个功能，要离开键盘去操作鼠标？
智能电视只能用遥控器？

原因四：

键盘输入元素丰富。如果能给键盘做一个很好的支持，那会给用户提供高效的操作环境。

BACKGROUND 背景

-  /  = 15%
- 2-4% severely so 相对严重
-  图像  行动  声音  认知
- or ...

BACKGROUND 背景

... normal people in such a situation

... 常人在上述方面的特殊环境下

 /  ≈ 100%

MOTOR 关于行动

- PC: mouse 鼠标 only?
- Mobile: touch 触控 only?
- TV: remote control 遥控器 only?
- Keyboard ≈ efficiency 效率

WEB STANDARDS 标准

- WCAG: Web Pages 网页
- UAAG: Browsers 浏览器, Screen Readers
- ATAG: CMS, WYSIWYG 所见即得工具

WCAG: web content, html标签

【← 前端开发者更应留意】

UAAG: 浏览器对“可访问性”能不能提供相应的支持

ATAG: (内容生产工具) 生产内容时能否Accessible

WEB STANDARDS 标准

- **WAI-ARIA**: Web App 富应用
- widget, role, props, states

highly recommended to read

强烈推荐深入学习

ARIA (Accessible Rich Internet Application)

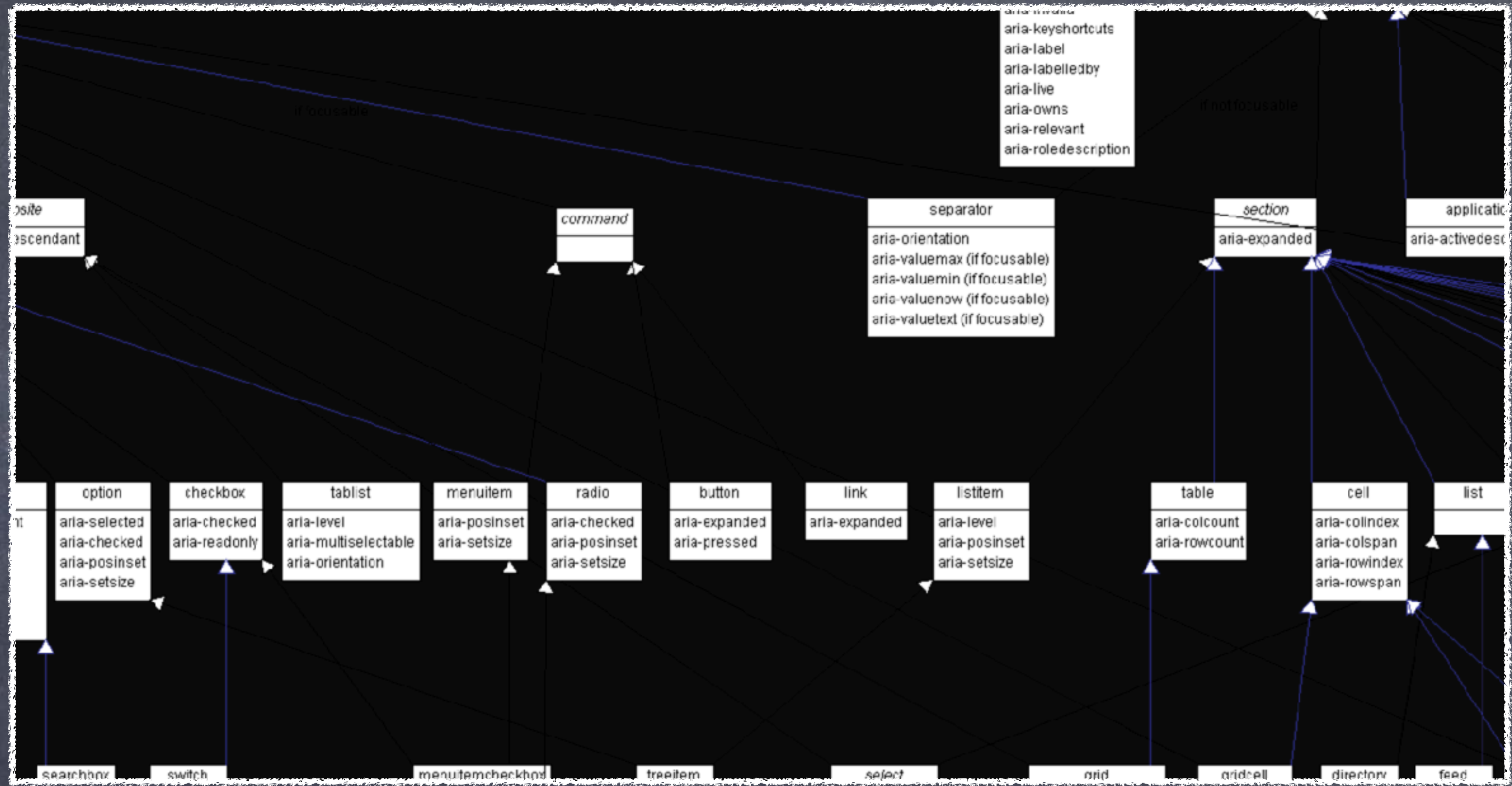
除了静态网页，一般都会有复杂的设计、交互，也带来更多Accessible的问题。

widget : 常用的组件，例：表单、文本框等

role : 告诉浏览器这个组件是干啥的

props : 用到的属性

states : 当前所处的状态



W3C ARIA规范列出的部分widget

aria规范的优点

- 有限的
- 开发更好的Accessible应用
- 方便理解交互设计、组件库的开发思路

使用HTML本身有语义化的标签

例: <header>、<article>

若要用自己的div, 至少把role、props、status设置好

提供必要的文本替代品

为一些缩写或简短的文字配以详细的描述

描述“交互逻辑”关系

描述“更详细的文本”关系

STRATEGY 实践策略

1. Atoms 原子组件
2. Groups 组件容器
3. Relationships 业务关系描述
4. App Layout 应用框架

①

Click Me

②

Click Me

menu / list / table ...

Click Me

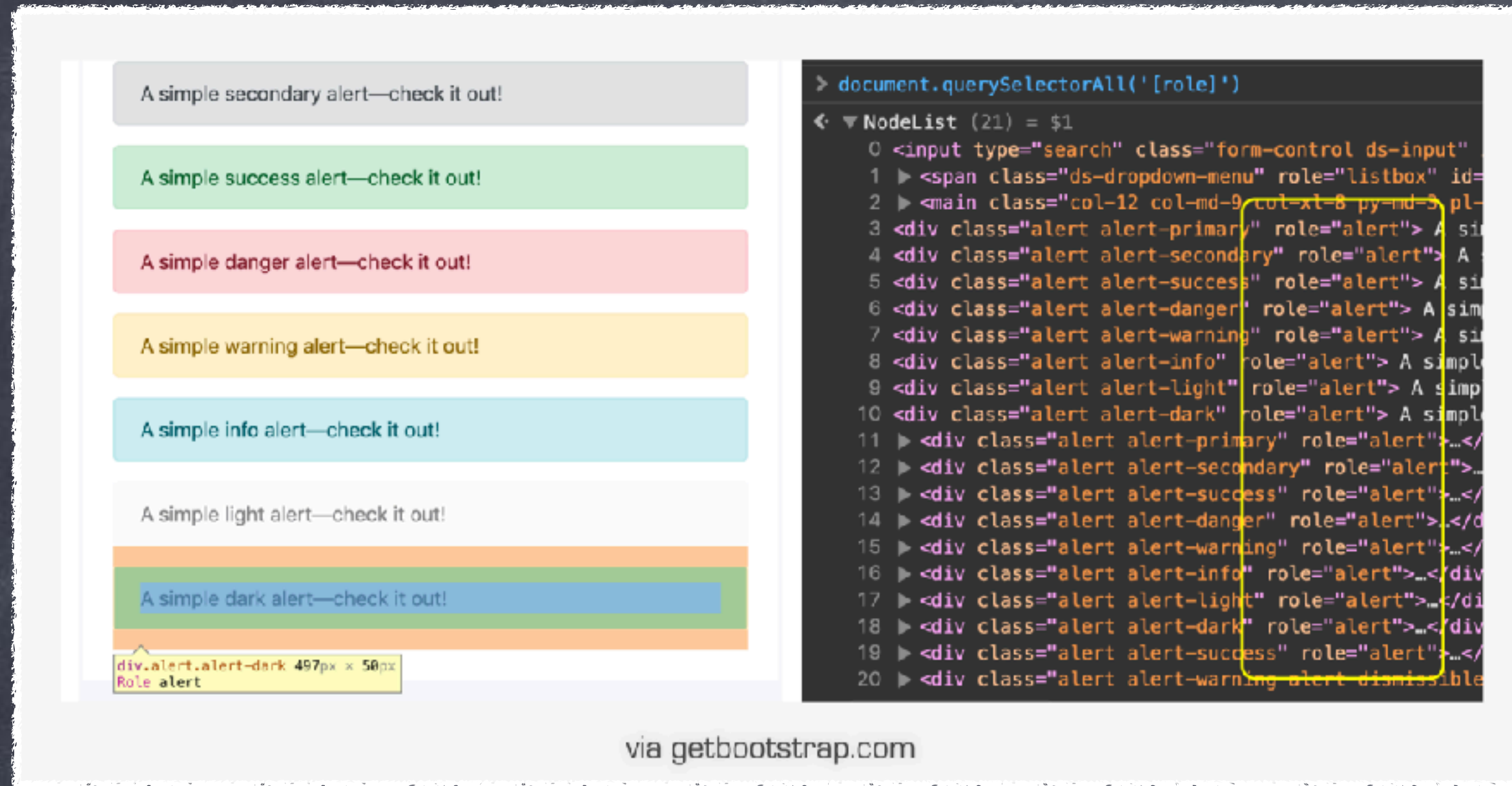
③

Click Me

④

- landmarks
- current
- live region

Back



The image shows a collection of Bootstrap alert boxes in various colors (grey, green, red, yellow, cyan, light blue, orange, dark blue) with the text "A simple [color] alert—check it out!". To the right, a browser's developer console shows the DOM tree for these alerts, with a yellow box highlighting the `role="alert"` attribute on each `<div>` element. Below the screenshot, the text "via getbootstrap.com" is visible.

使用role告诉浏览器这个div是个警告内容，增强可访问性

```
9      <!-- before -->
10     <label for="myID">账号</label>
11     <input id="myID" type="text" />
12
13     <!-- after -->
14     <div id="myLabel">账号</div>
15     <div tabindex="0" role="input" aria-labelledby="myLabel"></div>
```

以上都可以实现：在input获得焦点时，屏幕阅读器会读出相应label值

Back

Example

Subject	Summary	Email
▼ Treegrids are awesome	Want to learn how to use them?	aaron@thegoogle.rocks
re: Treegrids are awesome	I agree with you, they are the shizzle	joe@blahblahblah.blahblah
▶ re: Treegrids are awesome	They are great for showing a lot of ...	billy@dangerous.fish
▶ re: Treegrids are awesome	I hear the Fancytree library is going ...	someone@please-do-it.c...

表格中任何一个格子都可以focus，并且可以利用键盘的方向键进行操作，同时配有回车行为等

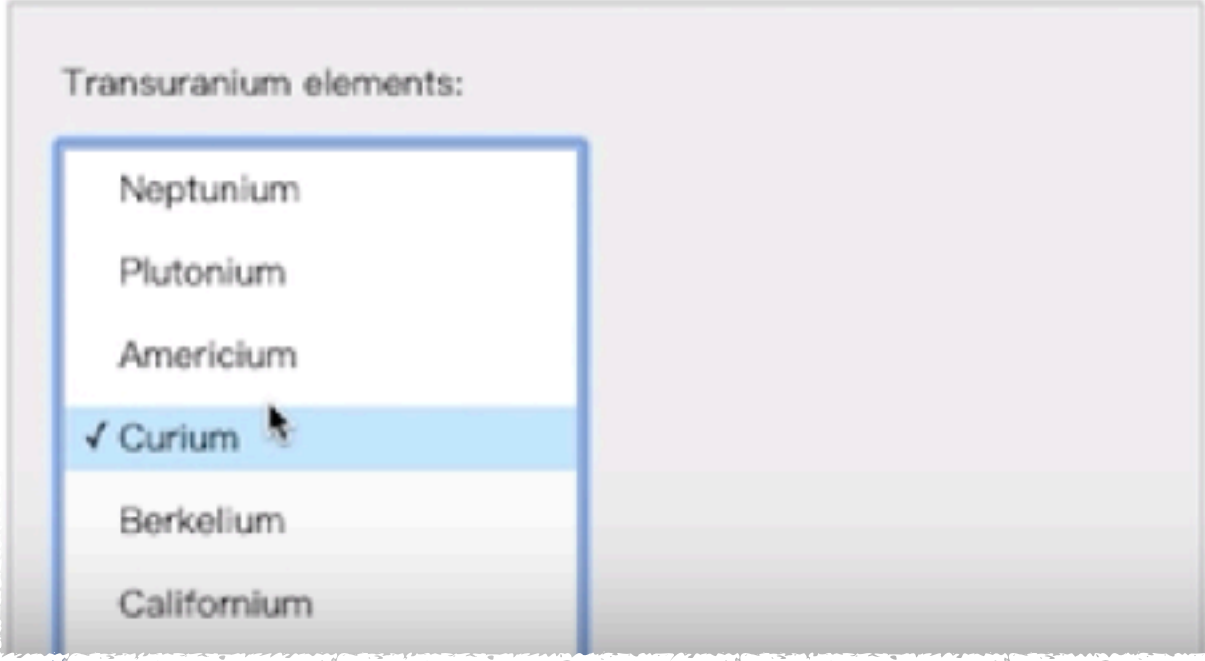
描述“交互逻辑”关系：

- aria-flowto
- aria-controls, aria-owns
- aria-activedescendant
- aria-errormessage

• Collapsible Dropdown Listbox Example: Single-select collapsible listbox that expands when activated, similar to HTML `select` with the attribute `size="1"`.

Example

Choose your favorite transuranic element (actinide or transactinide).



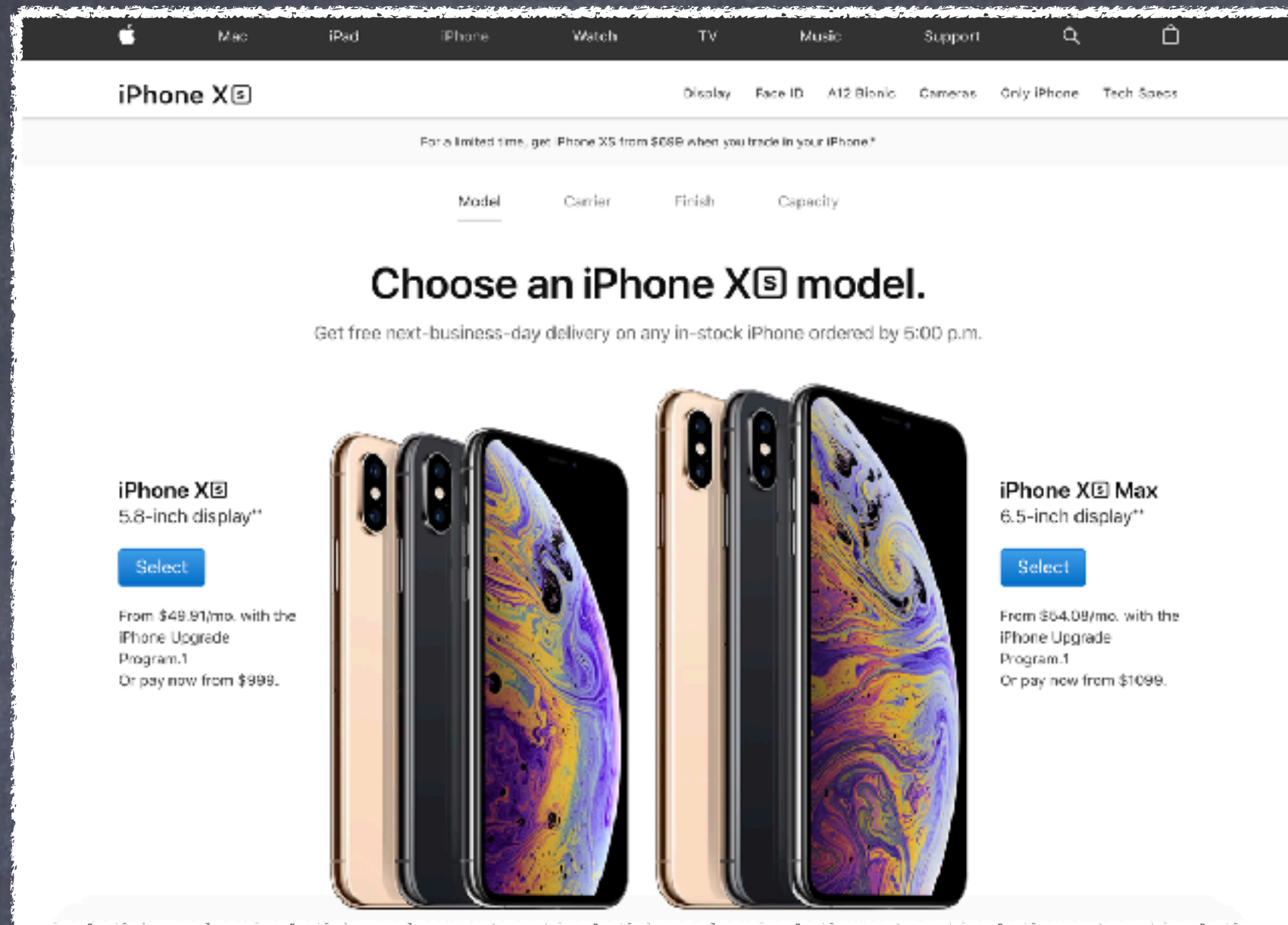
```
<ul id="ss_elem_list"
  tabindex="0" role="listbox"
  aria-labelledby="ss_elem"
  aria-activedescendant=
    "ss_elem_Cm">
  <li id="ss_elem_Np" role=
    "option" class=Neptunium
    </li>
  <li id="ss_elem_Pu" role=
    "option" class=Plutonium
    </li>
  <li id="ss_elem_Am" role=
    "option" class=Americium
    </li>
  <li id="ss_elem_Cm" role=
    "option" class=" focused"
    aria-selected="true">.</li>
  <li id="ss_elem_Bk" role=
    "option" class=Berkelium
    </li>
  <li id="ss_elem_Cf" role=
    "option" class=Californium
    </li>
  <li id="ss_elem_Es" role=
    "option">Einsteinium</li>
  <li id="ss_elem_Fm" role=
    "option">Fermium</li>
  <li id="ss_elem_Md" role=
    "option">Mendelevium</li>
  <li id="ss_elem_No" role=
    "option">Mobelium</li>
```

描述“更详细的文本”关系：

- aria-labelledby
- aria-describedby
- aria-details

给它们传一个ID，然后这个ID指向页面中存在的一大段的文本

landmarks (应用全局布局)



current

```
18 <nav aria-label="breadcrumb" role="navigation">
19   <ol class="breadcrumb">
20     <li class="breadcrumb-item active" aria-current="page">Home</li>
21   </ol>
22 </nav>
23
24 <nav aria-label="breadcrumb" role="navigation">
25   <ol class="breadcrumb">
26     <li class="breadcrumb-item"><a href="#">Home</a></li>
27     <li class="breadcrumb-item active" aria-current="page">Library</li>
28   </ol>
29 </nav>
30
31 <nav aria-label="breadcrumb" role="navigation">
32   <ol class="breadcrumb">
33     <li class="breadcrumb-item"><a href="#">Home</a></li>
34     <li class="breadcrumb-item"><a href="#">Library</a></li>
35     <li class="breadcrumb-item active" aria-current="page">Data</li>
36   </ol>
37 </nav>
```

live region

使用辅助工具的用户可以通过live region来获取文本中变更的部分，并且焦点仍保留在当前活动中。

使用HTML本身有语义化的标签

例: <header>、<article>

若要用自己的div, 至少把role、props、status设置好

提供必要的文本替代品

为一些缩写或简短的文字配以详细的描述

组件库开发者

menu / list / table ...

STRATEGY 实践策略

1. Atoms 原子组件
2. Groups 组件容器
3. Relationships 业务关系描述
4. App Layout 应用框架

业务团队

脚手架开发者

描述“交互逻辑”关系

描述“更详细的文本”关系

- landmarks
- current
- live region

那如果遵循了Aria规范，到了实际写代码的时候会有什么变化呢？

- role, aria-* 特性
- ID, refs 全局唯一引用
- tabindex, focus control 焦点控制
- modal dialog 模态对话框
- keyboard shortcuts 快捷键
- live region 实时播报

理解了前面的Accessibility概念、思想之后，到了实际开发觉得很繁琐、费劲？

The screenshot shows the homepage of the 'Vue A11y Utils' project. The page layout includes a top navigation bar with a search bar and links for 'Home', 'Examples', and 'GitHub'. A left sidebar contains a 'Table of Contents' and a list of project components and mixins. The main content area features the project title 'Vue A11y Utils', a logo consisting of a large 'V' and a person icon, and a row of badges indicating 'NPM' version 'v0.6.1', 'TYPES' as 'TYPESCRIPT', 'LICENSE' as 'MIT', and 'BUILD' as 'PASSING'. Below this, the text 'Utilities for accessibility (a11y) in Vue.js' is displayed, followed by a 'Table of Contents' section with a list of links to various parts of the project.

Vue A11y Utils

Table of Contents

- Why
- Getting Started
- <VueAria> Component
- v-aria Custom Directive
- Travel Mixin
- Id Mixin
- <VueFocusTrap> Component
- Shortcuts Mixin
- <VueLive> Component

Vue A11y Utils

NPM v0.6.1 TYPES TYPESCRIPT LICENSE MIT BUILD PASSING

Utilities for accessibility (a11y) in Vue.js

Table of Contents

- Why
- Getting Started
- <VueAria> Component
- v-aria Custom Directive
- Travel Mixin
- Id Mixin
- <VueFocusTrap> Component
- Shortcuts Mixin
- <VueLive> Component

Click me!!!

AGENDA 议程

- background 背景
- web standards 规范
- strategy 策略
- utils & demo 工具 & 示例

背景：有必要去关注特殊人群的使用体验

规范：W3C规范 aria-*特性

策略：从四个方面去努力

工具：Vue A11y Utils

AND DON'T FORGET 额外的

- pushing designer for 推动设计团队
 - accessible color pattern 配色方案
 - accessible interactions 交互改进
- testing 测试
- continuous iterations 持续迭代

呼吁设计团队一起参加，因为配色方案、交互改进和前端开发效率息息相关

找一些市场、目标用户去进行测试、提供反馈，进行改善

参加感受：

- 门票很贵
- 感受到了来自各个城市的前端开发者的年轻活力，一起为前端圈的良好发展贡献出了自己的青春和努力，觉得自己选择了前端是一件很对的事情
- 要保持不断学习新知识的心态，包括社区、国外的文献
- 要经常总结（特别是零碎的知识点），记录笔记，更新博客
- 不单单只是关注自己眼前的内容，还需要推动设计团队、测试以及一些容易遗忘的使用人群

Thank you