Weex vs. Web

Agenda

- Weex 中的 Web 标准
- Weex 中的 Web 研发模式
- 其它注意事项

Web 标准

HTML

HTML elements

100 个左右 link

Weex Components

~15

 <scroller> <list> <cell>
<refresh> <loading> <text> <image>
<input> <textarea> <switch> <slider>
<indicator> <video> <a> <web>

缺失的内容

- 块级语义化标签
- 内联文本标签
- 表单类标签

块级语义化标签

```
<template>
  <section>
    <text>Hello World</text>
  </section>
</template>
```

实际的效果

```
<template>
  <div>
    <text>Hello World</text>
  </div>
</template>
```

内联语义化标签

```
<template>
  <strong>Hello World</strong>
</template>
```

实际的效果

```
<template>
  <text style="font-weight: bold;">
    Hello World
  </text>
</template>
```

表单类标签

- <input>
- <textarea>
- radio -> <image> + <text>
- checkbox -> <switch>
- <select> -> picker

Using Vue `<template>`

Demo

How to use HTML

富文本组件 wip link

CSS

命题维度

- 选择器
- 属性名
- 属性值 (和单位)

特点

- 单个 class 选择器 (CSS in JS 最佳实践)
- 支持伪类 wip link
- 属性支持情况 link
- 支持 PostCSS / CSS Next

Demo

How to write CSS

JavaScript

- ES -> Babel / Polyfill link
- Native DOM APIs link
- Device APIs link

因为 Weex 在运行时

只有一个

JavaScript 引擎实例

所以我们必须面对

长效内存泄漏

的问题

目前我们采取的措施

- 强制 `"use strict"`
- 强制 `Object.freeze` 全局变量
- and ...

JS Service

wip link

- 在多实例中统一管理
- 拥有完整的生命周期管理
- 甚至跨多种 JS 框架统一管理

能力、数据和内存

得到有效管理和增强

脑洞比较大.....

Polyfill、AMD 实现、第三方库引入、native 回调函数管理、全局数据共享、全局路由管理、Worker...

Demo

How to use AMD with JS Service

额外的，基于

Vue 2.0

开发习惯几乎一致

差异整理 link

传统 Hybrid 扩展?

尽请关注我们今天的最后一个分享

研发模式

单页研发，多页聚合

SPA?

single-page application

SPA 的优势

- 避免多个页面重复加载资源
- 自定义专场效果
- 几乎没有页面之间的等待
- 全局数据共享、全局状态共享

其实用 SPA 也有纠结

- 首次打开的开销
- 内存管理问题
- 原生路由的配合、开放规则的应对
- 所有页面必须基于相同的 JS 框架

What Weex Do?

- 每个页面一个 JS bundle
- 可以支持原生的转场效果
- 运行时优化、缓存和预加载
- 通过 JS Service 进行资源复用
- 通过 JS Service 进行全局数据/状态管理

- 页面秒开
- 子团队 / 个人可以自由选择 JS 框架
- 为内存管理提供更好的引导
- 支持原生 & 开放规则的路由

Goodbye SPA

工程研发

初始化项目

- weex-toolkit
- vue-cli

开发

前端主流编辑器均可

调试

端上预览、hot-reload、devtool

测试

Macaca (wip) / ESLint

打包

webpack / babel / postcss ……

除此之外还有很多选择

发布

搭建、发布、缓存推送
沿用 Web 研发的最佳实践

监控

埋点，曝光埋点、交互埋点

沿用 Web 研发的最佳实践

Demo

一个典型的 Weex 项目

构建整个 App

- 组件生态和市场
- weex-pack

尽请关注我们今天的最后一个分享

其它注意事项

- 长列表性能优化 `<list>` / `<cell>`
- 流式渲染 `append="tree|node"`
- View 嵌套层级不宜过多 < 10
- HTML5 版本支持范围比 Native 大
- 推荐 Devtool 真机调试
- issues / gitter welcome !

Thanks