

日程安排

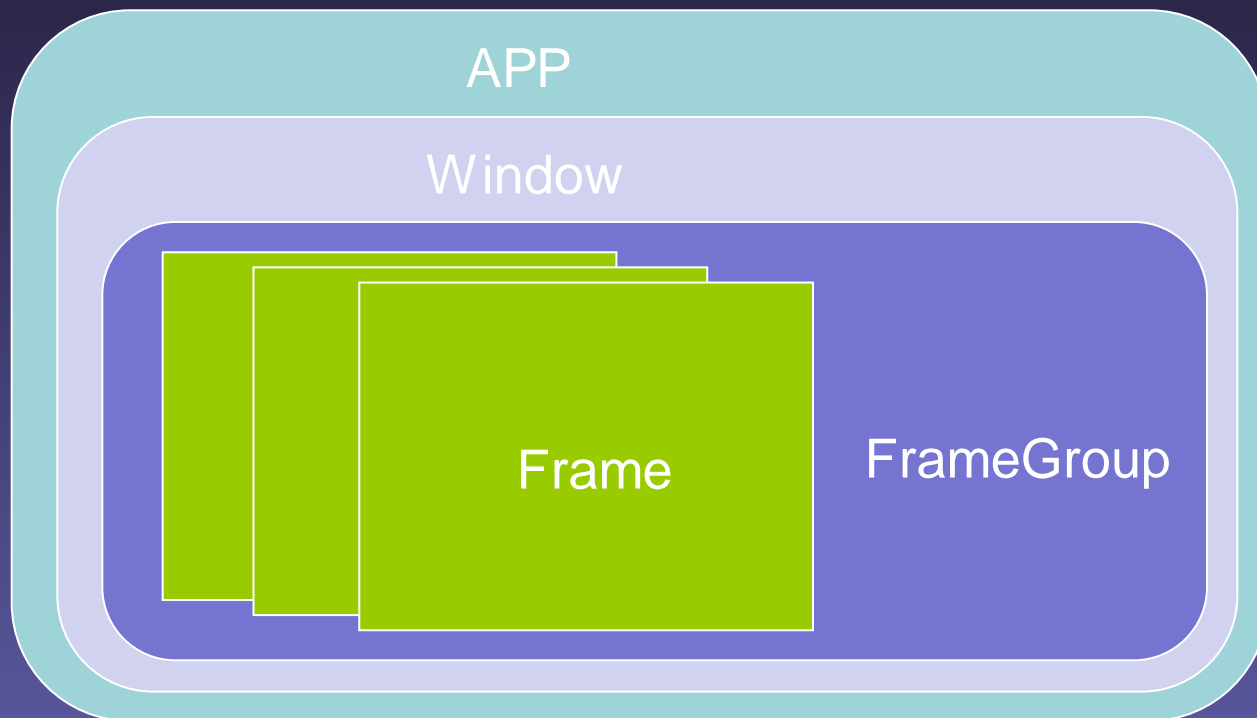


一、使用APICloud开发优质APP

二、快速模块开发

四、答疑交流

Window结构



结构配合
层次配合
显示/隐藏配合

关于api ready函数



- api ready是什么?

...

- 为什么要有api ready

...

- api ready与window.onload

...

API默认值策略



- 公有函数

open、close、show、hide、setXX、getXX、fixedOn、fixed、...

- 可选项策略

openWin: pageParam、bounces、bgColor、animation等均可选，可选值具体开发过程中及时参考文档

- 绝对赋值与 ‘auto’

w、h和赋值时可赋值auto，自动匹配父亲大小

HTML5的默认特性



- 关于viewport

`<meta name="viewport" content="maximum-scale=1.0, minimum-scale=1.0, user-scalable=0, width=device-width" />` 不配置，导致页面被无故放大等问题

- 元素默认样式

a、input标签，可focus元素-会显示黄色、蓝色框或者背景等默认效果，影响体验

- 去除默认样式

`-webkit-appearance:none;`

`-webkit-tap-highlight-color:rgba(0, 0, 0, 0);`

Config配置的妙用



- App/Window/Frame背景

```
<preference name="appBackground" value="color|imageUrl" />  
<preference name="windowBackground" value="color|imageUrl" />  
<preference name="frameBackgroundColor" value="color|imageUrl" />
```

- 自动更新&云修复

```
<preference name="autoUpdate" value="true|false" />  
<preference name="smartUpdate" value="true|false" />
```

- Debug模式

```
<preference name="debug" value="false" />
```

- 软键盘弹出模式

```
<preference name="softInputMode" value="auto|resize|pan"/>
```

- Feature

加载提速-代码集中



- HTML/JS/CSS代码集中在同一页面

公用css/js库尽量小

- 尽量减少过多的link与script标签

每一个link或者script标签浏览器引擎都将有一次同步I/O读写

- 代码集中的优点

降低开发过程中的工作耦合性、符合原生开发的设计思想

加载提速-不依赖框架



- 避免引入JQ、SenchaTouch等“重型框架”

JQ以及SenchaTouch等框架的事件流设计思想及其内部文档模型，严重拖慢UI响应速度，同时，框架内部Timer不断刷新页面，频繁占用CPU/GPU资源，拉低应用性能

- overflow:auto问题

引起DOM高度固定不变，导致立即弹动等问题

- 摆脱对\$函数的依赖

转变思想观念、养成自己动手的习惯、api.js、zepto.js

数据交互-最优AJAX体验



- 合理利用可选项

dataType、charset

headers、returnAll等

- 网络状态监听

online、offline、connectionType

- 异常处理

0 //连接错误

1 //超时

2 //授权错误

3 //数据类型错误

最简单的代码:

```
api.ajax({  
    url: 'http://www.xxx.com/xxx.api'  
}, function(ret, err) {  
    if(err) {  
        //do any  
        return;  
    }  
    //do any  
});
```

数据交互-缓存机制



- 合理利用图片缓存

引擎默认cache策略

`api.imageCache()`

- 速度对比

下载一张3M图片花5s = 下载150张20kb的缩略

- 合理利用数据缓存

AJAX设置cache

本地文件、localStorage、db等

高效渲染-图片处理



- 内存占用

图片占用内存算法： $RAM = \text{图宽} \times \text{图高} \times 4$

分辨率为200的图片： $200 \times 200 \times 4 = 160KB$

分辨率为600的图片： $600 \times 600 \times 4 = 1.44M$

相差9倍！

- 缩放图片的高耗性能

避免发生图片缩放，img标签多大，图片就给多大

- 客户端服务器配合解决

可通过客户端传参，服务器动态返回处理后的图片解决；使用第三方云存储现有的缩略图服务

高效渲染-UI模块



- UI模块的使用

导航类的UI模块具有全局唯一的特点，需合理配合window或者frameGroup使用

- fixedOn和fixed

fixedOn参数用于指定UI模块固定在哪个frame上，不指定则默认为window；fixed参数用于声明UI模块是否固定在页面上，不固定则跟随页面滚动。默认为true，即不跟随页面滚动

点击加速与点击效果



- onclick事件

浏览器的点击事件默认有300ms或更长的一个延迟

- tapmode属性

tapmode属性为APICloud为消除300ms延迟的问题专门定制的私有属性。
使用方式

```
<div tapmode onclick="going()"></div>
```

- 点击效果

给tapmode属性赋值一个css样式，即可实现点击效果；
DOM更新后，需要及时调用api.parseTapmode

高效渲染-尊重系统特性



- 适时更新UI

Google建议动画未完成之前不要进行node更新等操作，让GPU完成动画的过度，不要打搅它

- 充分使用CSS3即HTML5特性

CSS3动画、圆角处理、渐变处理、边框、新的input类型、语义化的标签（header、footer）等

- 避免body级别的背景

可在openWin或openFrame时传入bg，以原生级别的方式高效替代

最佳适配-UI适配



- box线性布局

```
display: -webkit-box;  
display: -webkit-flex;  
-webkit-box-orient: vertical;  
-webkit-box-flex: 1;  
-webkit-box-pack: center;  
-webkit-box-align: center;
```

- 背景/图片

```
background-size: 100% 53px;  
background: url(image/t.png) no-repeat center;
```

最佳适配-窗口结构设计



- window + frame结构

静态的无需频繁更新的title和导航栏部分放在window中，需要实时更新，响应用户及时操作的放在frame中

- frame区域高度最佳适配

JS运算得到的frame区域受诸多实际情况影响，最佳适配算法： $\text{api.winHeight} - \text{header} - \text{footer}$

- 设计图以640 * 960为基准

实际使用过程中，可根据适配程度，使用1280 * 720设计图，开发过程中，实际尺寸除以2

计量单位



- 绝对计量

兼容w3c标准所有计量单位，建议px、em，可根据不同场景使用。Css的px相当于原生的dip，即与设备分辨率无关的单位

- 相对计量

根据实际使用场景，使用百分比

- 设计图以640 * 960为基准

实际使用过程中，可根据适配程度，使用1280 * 720设计图，开发过程中，实际尺寸除以2

命名规范



- 任何资源文件一定避免中文命名
- 小写+下划线

如：order_pay.html；包含大写字母的资源文件在某些手机下可能存在兼容问题，报找不到页面错误

- 避免问号传参

Window与window间，window与frame间等，传参务必使用pageParam，问号传参同样有报找不到页面的兼容风险

设计优化-交互优化



- 合理利用window和frame之间交互
frame是可以和window一样大并且覆盖在window上的；
frameGroup是可以配置预加载的；
显示/隐藏替代打开/关闭操作，加载将更迅速
- 适时使用现有的原生层面接口
下拉刷新，滚动到底部加载更多、SlidLayout等

设计优化- 流程优化



- 先进入后判断登录

逻辑是死的，产品是活的，对于需要判断登录使用的业务，可以先进入主页后进行是否登录判断，未登录则打开登录窗口，登录成功后返回，过程很流畅

- 培养产品意识

Thanks!

www.w3school.com.cn