## 响应式开发的原理

CSS3中的Media Query（媒介查询），通过查询screen的宽度来指定某个宽度区间的网页布局。

|  |  |  |
| --- | --- | --- |
| 超小屏幕（移动设备） | 768px以下 | 768手机 |
| 小屏设备（平板） | 768px-992px | 992舅舅的儿子很小 |
| 中等屏幕（老式台式） | 992px-1200px | 1200十二点 |
| 宽屏设备 | 1200px以上 |  |

## 响应式开发和移动web开发的区别

|  |  |  |
| --- | --- | --- |
| **开发方式** | **移动web开发+PC开发** | **响应式开发** |
| 应用场景 | 一般在已经有PC端的网站，开发移动站的时候，只需单独开发移动端 | 针对新建站的网站，要求适配移动端，一套页面灵活兼容各种终端 |
| 开发 | 针对性强，开发效率高 | 兼容各种终端，效率低 |
| 适配 | 只适配移动设备，pad上体验相对较差 | 可以适配各种终端 |
| 效率 | 代码简洁，加载快 | 代码相对复杂，加载慢 |

@media screen and (max-width: 768px) { } /\*and后一定要有空格\*/

@media screen and (min-width: 992px) and (max-width: 1200px) { }

### **Bootstrap**

### 响应式布局的默认样式

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | html | body | | .container | | .container-fluid | |
| width | | 全屏 | 全屏 | |  | | 全屏 | |
| padding | | 0 | 0 | | 左右15px，上下0 | | | |
|  | <768 | | | <992 | | <1200 | | >1200 | |
| .container | 界限不明确 ~ 720 | | | | | 940 | | 1140 | |

### carousel

|  |  |  |
| --- | --- | --- |
| <div class= "carousel" > | 最大盒子 | position: relative; |
| <ol class="indicators"></ol> | 小圆点 | position: absolute; |
| <div class="inner">  <img src="image1">  <div class="intro"></div>  </div> | 图片盒子 | position: relative;  width: 100%;  overflow: hidden; |
| <a class="left"></a> | 左箭头 | position: absolute; |
| <a class="right"></a> | 右箭头 | position: absolute; |
| </div> |  |  |
| .inner中的div不用设置宽度，自占一行，但img要设width: 100%;才能占满.inner。.inner设position: relative;是为了方便.intro绝对定位 | | |

### jQuery中e返回的originalEvent就是原生js中的touchEvent

var startX = e.originalEvent.touches[0].clientX

### 导航条

|  |  |  |
| --- | --- | --- |
| <nav class="navbar-default"> |  | position: relative; |
| <div class="container-fluid”> |  | 点击button，.collapse展开，把<nav>撑高，若设置  position: absolute;  <nav>高度不变，不会影响其他元素的布局 |
| <div class="navbar-header">  </div> | 左边brand  右边button |
| <div class="navbar-collapse">  </div> | ul/brand |
| </div>  </nav> |  |
| 在导航条框架外面套.container VS 直接引框架，再把里面的.container-fluid改成.container，在移动端会有不同表现。 | | |

### 类名

|  |  |  |
| --- | --- | --- |
| .h1 到 .h6 | 给内联（inline）属性的文本赋予标题样式 | |
| .small | 标记副标题 | |
| .lead | 让段落突出显示 | |
| .text-left / .text-center / .text-right / .text-justify / .text-nowrap | | |
| .text-lowercase / .text-uppercase / .text-capitalize | | |
| .dl-horizontal | 让 <dl> 内的短语及其描述排在一行。开始是像 <dl> 的默认样式堆叠在一起，随着导航条逐渐展开而排列在一行 | |
| .table-striped | 给 <tbody> 之内的每一行增加斑马条纹样式 | |
| .table-bordered | 为表格和其中的每个单元格增加边框 | |
| .table-hover | 让 <tbody> 中的每一行对鼠标悬停状态作出响应 | |
| |  |  | | --- | --- | | .active | 鼠标悬停在行或单元格上时所设置的颜色 | | .success | 标识成功或积极的动作 | | .info | 标识普通的提示信息或动作 | | .warning | 标识警告或需要用户注意 | | .danger | 标识危险或潜在的带来负面影响的动作 | | | |
| 将任何 .table 元素包裹在 .table-responsive 元素内，即可创建响应式表格，其会在移动端水平滚动，在非移动端滚动条消失 | | |
| .pull-left/.pull-right | | 左右浮动 |
| .clearfix | | 清除浮动 |
| .hidden-xs | | 在超小屏消失 |
| .visible-xs-block / .visible-xs-inline-block | | |

### 标签

|  |  |  |  |
| --- | --- | --- | --- |
| <mark> | For highlighting a run of text due to its relevance in another context | <b> | 用于高亮单词或短语，不带有任何着重的意味 |
| <del> | 被删除的文本 | <u> | 为文本添加下划线 |
| <i> | 主要用于发言、技术词汇等 | <var> | 标记变量 |

运行-输入“cmd”，出来命令提示符，再输入“node -v”（空格必须有），跳出5.2.0，再输入npm install -g less安装less，然后输入lessc -v检测是否安装成功。

先把png 转成 SVG格式，然后把SVG上传到阿里的iconfont，然后再下载

$.ajax({

url:'js/index.json', /\*\*js是被html引用的，所以路径要从html开始\*\*/

data:{ },

type:'get',

dataType:'json',

success: function(data){

callback && callback(myData); },

error: function(){ }

});

student记忆法：success, type, url, data, error, dataType

Ajax相当于点菜，先把点的菜告诉服务生（server），服务生去后厨（url()里面的文件里有数据）取菜，然后再端上来。get/post就是点菜的方式，是直接告诉服务生还是在菜单上勾选。

发送请求时的type有两种，一个是get，发送的数据是有限的，并且在地址栏能看到。另一种是Ｐost，可以无限发送，传的内容在地址栏也看不到，但是传的慢一些。

**template** \_.template(templateString, [settings])   
Compiles JavaScript templates into **function**s that can be evaluated for rendering. Useful for rendering complicated bits of HTML from JSON data sources. Template functions can both interpolate values, using <%= … %>, as well as execute arbitrary JavaScript code, with <% … %>. If you wish to interpolate a value, and have it be HTML-escaped, use <%- … %>. When you evaluate a template function, pass in a **data**object that has properties corresponding to the template's free variables. The **settings**argument should be a hash containing any \_.templateSettings that should be overridden.

var compiled = \_.template("hello: <%= name %>");

compiled({name: 'moe'});

=> "hello: moe"

var template = \_.template("<b><%- value %></b>");

template({value: '<script>'});

=> "<b>&lt;script&gt;</b>"

You can also use print from within JavaScript code. This is sometimes more convenient than using <%= ... %>.

模板渲染引擎相当于一个间谍，第一步先把情报解密（把<% %>中的文字按一定的方法翻译过来），第二步按照情报的指示实施暗杀（<% %>中如果是js代码就执行这段代码）。

var compiled = \_.template("<% print('Hello ' + epithet); %>");

compiled({epithet: "stooge"});

=> "Hello stooge

**each** \_.each(list, iteratee, [context]) 依次对集合的每个成员进行某种操作，但不返回值。  
Iterates over a list of elements, yielding each in turn to an iteratee function. The iteratee is bound to the context object, if one is passed. Each invocation of iteratee is called with three arguments: (element, index, list). If list is a JavaScript object, iteratee's arguments will be (value, key, list). Returns the list for chaining.

\_.each([1, 2, 3], alert);

=> alerts each number in turn...

\_.each({one: 1, two: 2, three: 3}, alert);

=> alerts each number value in turn...

Note: Collection functions work on arrays, objects, and array-like objects such as arguments, NodeList and similar. But it works by duck-typing, so avoid passing objects with a numeric length property. It's also good to note that an each loop cannot be broken out of — to break, use **\_.find** instead.

class=”<%=i==0?”active”:””%>”

ctrl+r 调出标尺 ctrl+c切片

APICloud使用

APP有三种：Native（安卓、IOS）、H5APP、混合开发

学习：开发者社区-文档

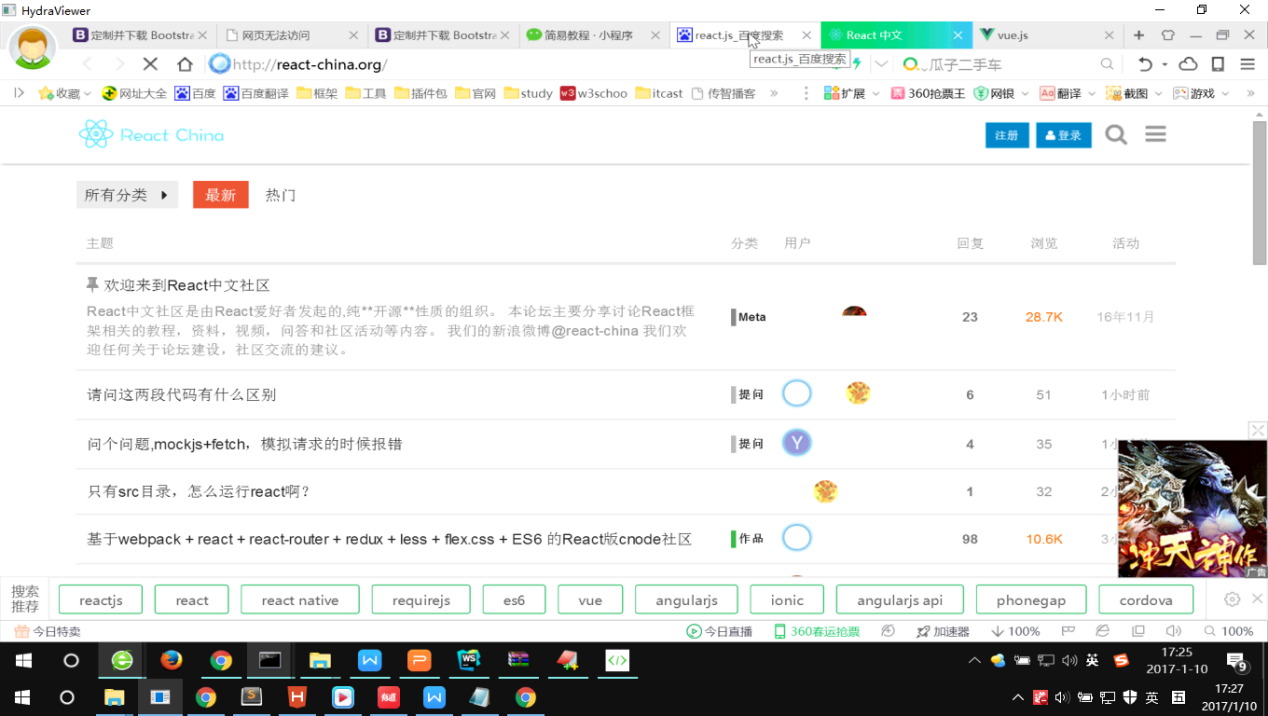
在控制台-创建应用-输入名称-端设置-

在应用商店找上传资料，然后打包完上传到苹果的APP商店，需要审核至少2周。安卓一般需要两天。

代码需要svn账号（俗称小乌龟）

路径要选第三项（在电脑上的路径D：/），不要选第一项。





慕课网

52cto.com

网易云课堂

csdn.net

人人都是产品经理