移动web

1. 移动web的概念

专门为移动端开发的网站

不是所有手机上的网站都是移动web

1. 相比于PC网站
   1. 用户量大
   2. 随时随地使用
   3. 浏览器版本高 可以使用更新更好的技术
2. 相比于移动App
   1. 学习成本低 html+css+js
   2. 以此编码,多个平台通用 只需要一个浏览器即可运行
   3. 可通过工具,将网站打包成不同平台的App
   4. 发布简单,不用经过审核
3. 屏幕知识
   1. 对角线 单位 英寸
   2. 设备分辨率 屏幕里一共拥有的物理像素点的个数

物理像素点:屏幕里面显示图像的最小单位

屏幕大小相等的情况下 分辨率越高越清晰

* 1. 逻辑分辨率 屏幕的宽度,高度 单位像素
  2. PPI 像素密度 每英寸所拥有的物理像素点
  3. Scale factor 放大比例 屏幕的清晰倍数

1. 失真

分辨率低的图片在分辨率高的设备上显示会变得模糊

原因::会把低像素的图片进行拉伸

解决方案: 不同分辨率加载不同清晰度的图片

1. Js 获取window的devicePixelRatio属性 检测当前屏幕的分辨率,根据不同分辨率加载不同分辨率的图片
2. 给img标签加入srcset属性

传参: 图片路径 屏幕倍数 图片路径屏幕倍数

Iphone3以下 1x

Iphone4-iphone6 2x

Ipone6+以上 3x

如:

1. 让美工准备高清图片,直接使用不需要考虑 (使用最多的方案)
2. 视口

一 触屏事件

1.鼠标事件不能模拟多指触控事件

2.鼠标的click事件在移动端上存在延迟 - 用户体验不好

所以有了 触屏事件

Click:

1. pc 单击和双击都会触发点击事件
2. 移动端 单击后会有延迟看是否有第二次点击 有 放大屏幕 无 执行点击事件,所以不管有没有多次点击,click事件都会有延迟 目的 让用户方便放大屏幕

三个新的触屏事件(不能在pc端触发)

建议通过addEventListener绑定, on可能会出现兼容问题

1. touchstart : 按下
2. touchmove: 移动
3. touchend : 离开屏幕

触屏事件的事件源对象 e

因为触屏事件可能多指触发 所以里面的数据是以数组形式储存的

1. touches 屏幕上所有触摸点对象的集合
2. targetTouches 事件源上的所有的触摸点对象的集合
3. changedTouches 事件源发生了状态(进入,离开等)改变的触摸点的集合

每个数组里存着三对数据(主要)

clientX clientY 相对于视口的坐标

pageX pageY 相对于页面的坐标 client+滚动条的距离

screenX screenY 相对于屏幕的坐标

移动端上的点击事件叫做 tap (统一叫法!!!!)

Click比移动端的点击事件要慢多少? 200ms-300ms左右

封装tap的条件 点击

1. 手指的个数不能超过一个
2. 按下时间不能太长 <200ms
3. 判断滑动不能太大 <5px

思路:1在touchstart事件中获取初始时间和初始坐标并进行判断触摸点个数

2 在touchend事件中获取终止时间和终止坐标并进行判断

封装swipe 轻滑

1. 手指的个数不能超过一个
2. 按下时间不能太长 <800ms
3. 判断滑动不能太大 >5px

模块化:

将功能分成一个个模块,需要哪个功能就引入哪个模块

优点: 文件小

缺点: 麻烦

解决方案: 可以自己定制 通过node.js

Bootstrap

栅格系统

页面布局随着宽度的改变发生变化

四部分

1.极小屏幕 xs 768以下

2.小屏幕 sm 768-992之间

3.中等屏幕 md 992-1200之间

4.大屏幕 lg 1200以上

栅格系统的编写步骤

1. 写个容器 .container(版心) .container-fluid(全屏)
2. 写行 .row
3. 再去写栅格
   1. 写较大屏幕的栅格不写小的不会作用于小屏幕
   2. 写小屏幕的栅格不写大的会作用于大屏幕
   3. 默认独占12份
   4. 可以进行嵌套 把一个栅格看做一个容器 再分成12份

会将整个屏幕分成12份 超过的会换行

col-lg-n 大屏幕下每列占n份