

一. 移动端布局及适配方式

1. dpr与viewport

- a. 手机浏览器是把页面放在一个虚拟的“窗口”（viewport）中，窗口可大于或小于手机的可视区域，一般手机默认viewport大于可视区域。

(980px, 黑莓

1024, ie1024)

这样不会破坏没有针对手机浏览器优化的网页的布局，用户可以通过平移和缩放来看网页的

其他部分。

- b. `<meta id="viewport" name="viewport" content="width=device-width,initial-`

`scale=1,user-scalable=no"/>`

- c. iphone3GS 是 320 480，屏幕尺寸为3.5寸，此时device-width = 320; iphone4 640 960, 屏幕尺寸为3.5寸，如果device-width 是640 ,那么之前的布局正好

是一半，会影响其可读性，维持继续是320px,ipad维持在1024px, 此时device-width就不是设备宽了（也就不是分辨率的宽了），是独立像素。

Android也一样，device-width值以360居多，也有540px, 600px.

在设置了`<meta />`标签以后，device-width值可以用`window.innerWidth`来获取device-width

值,其实也是物理分辨率/dpr

2. rem等比缩放方案

```
1 <meta id="viewport" name="viewport"
2 content="width=device-width,initial-scale=1,user-scalable=no"/>
3
4 var ratio = window.devicePixelRatio;
5 var meta = document.getElementById("viewport");
6 meta.setAttribute("content",
7 `width=device-width,initial-scale=${1/ratio},user-scalable=no`);
8
9 document.documentElement.style.fontSize=window.innerWidth/750*100+"px";
```

3. 1px边框问题（为什么移动端css里面写了1px, 实际看起来比1px粗.）

```
1 在devicePixelRatio = 2 时，输出viewport:
```

```
2 <meta name="viewport" content="initial-scale=0.5, maximum-scale=0.5, minimum-scale=0.5, user-scalable=no">
3 在devicePixelRatio = 3 时, 输出viewport:
4 <meta name="viewport" content="initial-scale=0.3333333333333333, maximum-scale=0.3333333333333333, minimum-scale=0.3333333333333333, user-scalable=no">
```

<https://www.cnblogs.com/lunarorbitx/p/5287309.html>

二. 移动端事件相关

1. click事件300ms延迟

移动web页面上的click事件响应都要慢上300ms

移动设备访问web页面时往往需要“双击”或者“捏开”放大页面后来看清页面的具体内容。

正是为了确认用户是“单击”还是“双击”，safari需要300ms的延迟来判断。

后来的iphone也一直沿用这样的设计，同时android也借鉴并沿用了这样的设计。

于是“300毫秒的延迟”就成为了一个默认规范。

解决：(1) 设置meta viewport (2) fastclick

2. Zepto.js

Zepto是一个轻量级的针对现代高级浏览器的JavaScript库，它与jquery有着类似的api。如

果你会用jquery，那么你会用zepto,zepto使用需要注意以下几点：

- (1) 提供jQuery的类似的API，但并不是100%覆盖jQuery；
- (2) Zepto不支持旧版本的Internet Explorer浏览器(<10)；
- (3) zepto添加了完整的touch手势支持;解决300ms延时（tap取代click,

需要引入

zepto.touch.js)

- 移动端浏览器上默认支持四个事件touchstart touchmove touchend touchcancel 四个事件
- tap事件touchstart touchmove touchend 3个事件模拟实现的

jquery与zepto 异同 <https://www.cnblogs.com/colima/p/5289386.html>

3. Hammer.js

[HammerJS](#)是一个优秀的、轻量级的触屏设备手势库

Pan	平移
Pinch	双指缩放
Press	长按
Rotate	旋转
Swipe	滑动
Tap	点击

hammer.js <https://github.com/hammerjs/hammer.js>

vue touch <https://github.com/vuejs/vue-touch/tree/next>

```
<!-- Renders a div element by default -->
<v-touch v-on:swipeleft="onSwipeLeft">Swipe me!</v-touch>

<!-- Render as other elements with the 'tag' prop -->
<v-touch tag="a" v-on:tap="onTap">Tap me!</v-touch>
```

修改后需要重新刷新页面

tag属性可以指定要渲染的元素