### 移动端适配

1. 像素

|  |  |
| --- | --- |
| CSS像素 | 为web开发者创造的，在CSS 和JavaScript 中使用的抽象的层 |
| 设备像素 | 设备屏幕的物理像素，任何设备的物理像素的数量都是固定的，  物理像素是计算机能够显示一种**特定颜色**的最小区域。 |
| 一个css像素相当于多少个设备像素取决于屏幕和用户的缩放 | |

1. 视口viewport

布局视口layout viewport、视觉视口visual viewport、理想视口

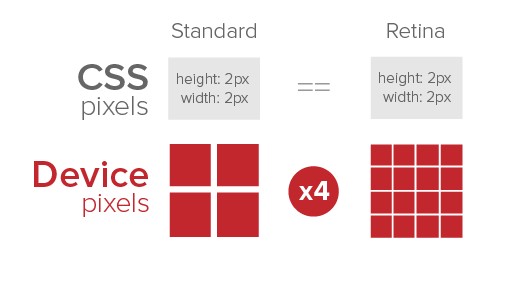
理想视口，对设备来讲，是最理想的视口尺寸。显示在理想视口中的网站拥有最理想的宽度，不需要缩放。

<meta name="viewport" content = "width=device-width, user-scalable = no, initial-scale = 1.0, maximum-scale = 1.0, minimum-scale = 1.0">

这行代码告诉浏览器我要使用理想视口了。

1. 分辨率

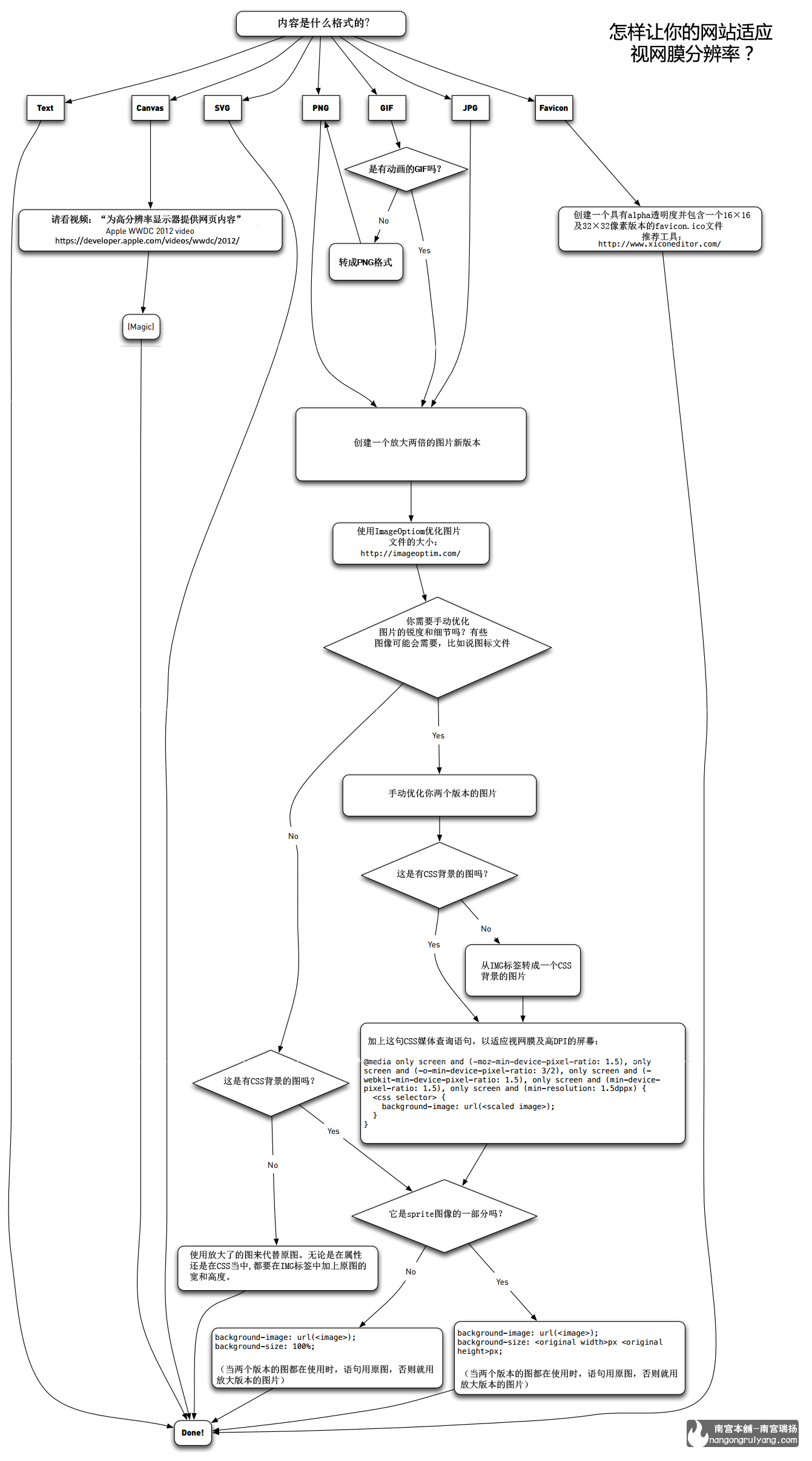
|  |  |
| --- | --- |
| 物理分辨率 | 单位为DPI （dots per inch），开发者无法获取该信息 |
| 设备像素比 | window.devicePixelRatio = 物理像素 / CSS像素  设备独立像素(dips: device-independent pixels) |



1. 字体和尺寸

适配有两种方案：

|  |  |
| --- | --- |
| [媒体查询](http://www.w3cplus.com/content/css3-media-queries" \t "http://www.jianshu.com/p/_blank) |  |
| 字体 | 不推荐使用rem，而是根据DPR 设置字体，  如果用rem，动态修改根元素的font-size。 |
| 布局 | rem |



1. 横竖屏样式

//竖屏时使用的样式

@media all and (orientation:portrait) {

code here ...

}

//横屏时使用的样式

@media all and (orientation:landscape) {

code here ...

}

### 当用户在操作按钮的时候，如果按钮没有任何反应，体验是很差的。在pc端我们都会习惯加上hover属性，来改变按钮状态；但移动端没有鼠标一说，这个时候我们就可以让active上场了。

### lib-flexible

淘宝有个叫[lib-flexible](https://github.com/amfe/lib-flexible" \t "http://www.w3cplus.com/mobile/_blank)的开源库，专门用来解决H5页面终端适配问题。

也可以直接使用阿里CDN：

<script src="http://g.tbcdn.cn/mtb/lib-flexible/{{version}}/??flexible\_css.js,flexible.js"></script>

将代码中的{{version}}换成对应的版本号0.3.4。

### meta用法

// 禁止百度SiteApp转码声明

<meta http-equiv="Cache-Control" content="no-siteapp">

// 禁止自动识别电话和邮箱

<meta name="format-detection" content="telephone=no, email=no">

// 指定iphone中safari顶端的状态条的样式<meta name = "apple-mobile-web-app-status-bar-style" content = "black-translucent">

(default：白色；black：黑色；black-translucent ：半透明)

// 隐藏地址栏，启用 WebApp 全屏模式

<meta name="apple-mobile-web-app-capable" content="yes">

// 优先使用 IE 最新版本和 Chrome

<meta http-equiv="X-UA-Compatible" content="IE=Edge,chrome=1">

// 注明作者

<meta name="author" content="www.jiawin.com">

### html5重力感应事件

if (window.DeviceMotionEvent) {

window.addEventListener('devicemotion',deviceMotionHandler, false);

}

var speed = 30;//speed

var x = y = z = lastX = lastY = lastZ = 0;

function deviceMotionHandler(eventData) {

var acceleration =event.accelerationIncludingGravity;

x = acceleration.x;

y = acceleration.y;

z = acceleration.z;

if(Math.abs(x-lastX) > speed || Math.abs(y-lastY) > speed || Math.abs(z-lastZ) > speed) {

alert('别摇那么大力嘛...');

// your code here

}

lastX = x;

lastY = y;

lastZ = z;

}