█ 基本概念

物理像素（physical pixel）：设备能控制显示的最小的单位

逻辑像素，也叫设备独立像素（DIP, device-independent pixel, density-independent pixel）：独立于设备的用于逻辑上衡量像素的单位（比如CSS的px）

设备像素比（DPR, device pixel ratio）：它的数值体现了物理像素和逻辑像素之间的关系

DPR = 物理像素 / 逻辑像素

JS可以通过window.devicePixelRatio方法来获取设备的DPR

例如

console.log(window.devicePixelRatio)

█ navigator.userAgent方法来识别终端是什么浏览器或者移动设备

●案例一，识别PC终端用的什么浏览器

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Document</title>

</head>

<body>

<script>

function validBrowser(){

var u\_agent = navigator.userAgent;

var browser\_name='Failed to identify the browser';

if(u\_agent.indexOf('Firefox')>-1){

browser\_name='Firefox';

}else if(u\_agent.indexOf('Chrome')>-1){

browser\_name='Chrome';

}else if(u\_agent.indexOf('Trident')>-1&&u\_agent.indexOf('rv:11')>-1){

browser\_name='IE11';

}else if(u\_agent.indexOf('MSIE')>-1&&u\_agent.indexOf('Trident')>-1){

browser\_name='IE(8-10)';

}else if(u\_agent.indexOf('MSIE')>-1){

browser\_name='IE(6-7)';

}else if(u\_agent.indexOf('Opera')>-1){

browser\_name='Opera';

}else{

browser\_name+=',info:'+u\_agent;

}

document.write('browser\_name:'+browser\_name+'<br>');

document.write('u\_agent:'+u\_agent+'<br>');

}

validBrowser()

</script>

</body>

</html>

●移动端，识别是安卓还是IOS

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Document</title>

</head>

<body>

<script>

var u = navigator.userAgent;

var isAndroid = u.indexOf('Android') > -1 || u.indexOf('Linux') > -1;

if (isAndroid) {

console.log('使用的是安卓移动设备')

} else {

console.log('使用的是IOS移动设备')

}

</script>

</body>

</html>

█CSS3的DPR媒体查询

-webkit-device-pixel-ratio

-webkit-min-device-pixel-ratio

 -webkit-max-device-pixel-ratio

例如

<style>

/\* 兼容iphone6\*/

@media (device-width: 375px) and (device-height: 667px) and (-webkit-min-device-pixel-ratio: 2.0) {

html {

font-size: 16px;

}

}

/\* 兼容iphone6 plus \*/

@media (device-width: 414px) and (device-height: 736px) and (-webkit-min-device-pixel-ratio: 3.0) {

html {

font-size: 16px;

}

}

</style>

█根据不同设备的不同DPI来改变逻辑像素值

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Document</title>

<style>

#d1{

background-color: lawngreen;

}

</style>

</head>

<body>

<div id="d1"></div>

<script>

// 获取手机机型

var u = navigator.userAgent;

var isAndroid = u.indexOf('Android') > -1 || u.indexOf('Linux') > -1;

// 函数setDpr可根据不同设备的DPI来自动计算

function setDpr (px){

var dpr = window.devicePixelRatio;

if(isAndroid){

return parseInt(px);

}else{

if(dpr >= 3){

return parseInt(px\*3);

}else if(dpr >=2) {

return parseInt(px\*2);

}else{

return parseInt(px);

}

}

}

// 应用

var d1 = document.getElementById('d1')

d1.style.height = setDpr(100) + 'px'

d1.style.width = setDpr(50) + 'px'

</script>

</body>

</html>

**说明：**

原理：

如果DPI是1 （一般安卓设备），则逻辑像素的值就是本来不变

如果DPI是2（如苹果6），则逻辑像素的值就要乘以2

如果DPI是3（如苹果6plus），则逻辑像素的值就要乘以3

CSS里通过meta标签里的 target-densitydpi=device-dpi来自动计算逻辑像素值，也就是CSS代码里比如宽度等于100px，那就这么写，系统会自动根据设备DPI来进行计算

target-densitydpi=device-dpi

设备会按照真是的DPI来渲染页面。打个比方说，一张320\*480的图片，放在IPHONE4里，默认是占满屏幕的。但是如果设定了target-densitydpi=device-dpi，那么图片只占屏幕的四分之一（二分之一的平方），因为IPHONE4的分辨率是640\*960

width=device-width

页面的宽度自动适配成等于设备屏幕的宽度

user-scalable=no

不允许用户对页面进行缩放

initial-scale=1.0

页面原始比例为1比1（没有缩放）

minimum-scale=1.0, maximum-scale=1.0

如果设置了user-scalable=yes

则需要设置这两个，比如最小缩放到80%，minimum-scale=0.8

最大缩放到120%，maximum-scale=1.2

如果是JS动态改变（比如需要给一个元素使用JS来设置其宽度等于100），则需要根据上例的JS代码所描述的那样的做法

参考网址：

<https://www.jianshu.com/p/b880b48fa028>

<https://www.cnblogs.com/2050/p/3877280.html>