流式布局就是百分比布局，通过盒子的宽度设置成百分比(100%)来根据屏幕的宽度来进行伸缩，特点：不受固定像素的限制，内容向两侧填充。

viewport相当于相纸。

A viewport represents a polygonal (normally rectangular) area in computer graphics that is currently being viewed. In web browser terms, it refers to the part of the document you're viewing which is currently visible in its window. Content outside the viewport is currently not displayed onscreen.

viewport 不包括滚动条、工具栏。

window 也不包括工具栏，但是包括滚动条。

body 的父级是html，html的父级是viewport

|  |  |  |
| --- | --- | --- |
| 代码 | window.innerWidth | body.offsetWidth  html.offsetWidth |
| html,body {  width: 100%;  } | 1366px | 1349px  有滚动条，因为body: margin:0 |
| html,body {  margin: 0;  width: 100%;  } | 1366px | 1349px  无滚动条 |

<meta name = “viewport” content = “width = 320”> 这个标签是让浏览器知道我设置了viewport，它只在移动端识别。width是让屏幕的宽度与viewport的宽度一致。如果content = “width = device-width”, 是让屏幕的宽度与不同设备viewport的宽度一致。

content = “initial-scale = 1.0”，和PC端页面的显示大小一致

content = “width = device-width, initial-scale = 1.0”

user-scalable 是否允许用户自行缩放1，0或yes, no

行业通用的viewport设置：

<meta name = “viewport” content = “width = device-width, initial-scale = 1.0, user-scalable = 0”>

如果不设这一行代码，所有的页面在手机端上html/body都默认是980px。

设置了viewport，使得DOM宽度(layout viewport)与屏幕宽度(visual viewport)一样大，DOM文档主宽度即为屏幕宽度。

在浏览器调试移动端时页面上给定的像素大小就是**理想视口**大小，它的单位正是设备独立像素。

设备像素比device pixel ratio简称dpr，即物理像素和设备独立像素的比值。

在写CSS时，我们用到最多的单位是px，即CSS像素，当页面缩放比例为100%时，一个CSS像素等于一个设备独立像素。

但是CSS像素是很容易被改变的，当用户对浏览器进行了放大，CSS像素会被放大，这时一个CSS像素会跨越更多的物理像素。

页面的缩放系数 = CSS像素 / 设备独立像素 = 理想视口 / 视觉视口。

淘宝是非主流的viewport设置： 先用js判断屏幕尺寸，然后设定不同的initial-scale值来确定屏幕的缩放比。

|  |  |
| --- | --- |
| name = “viewport” |  |
| content = “width = 320” |  |
| content = “width = device-width” |  |
| content = “initial-scale = 1.0” |  |
| content = “width = device-width, initial-scale = 1.0” |  |
| user-scalable = 0 |  |
| maximum-scale: | 最大缩放比 |
| minimum-scale | 最小缩放比 |

meta:vp按tab

-webkit-tap-highlight-color: transparent;

-webkit-appearance: none; 非标准属性，尽量不要使用

屏幕分辨率

指一个屏幕具体由多少个像素点组成。

图片分辨率是指图片分别在水平和垂直方向上所具有的像素点数。同一尺寸的图片，分辨率越高，图片越清晰。

1. 屏幕像素密度PPI(pixel per inch)

PPI可以用于描述屏幕的清晰度以及一张图片的质量。使用PPI描述图片时，PPI越高，图片质量越高，使用PPI描述屏幕时，PPI越高，屏幕越清晰。

1. DPI(dot per inch)

在长为1英寸的范围内能显示的点数。这里的点是一个抽象的单位，可以指打印机输出的点，也可以指LED屏幕等设备上的物理像素点，或者电子墨水屏上的图像单元。

随着媒体查询和响应式设计的出现，描述显示器的分辨率出现了三个新单位。

1. DPPX

每个CSS像素显示的点数。1dppx相当于96dpi，视网膜屏的dppx 为2

*device-pixel-ratio*: Number of device pixels per CSS Pixel这和dppx一致。

手机分辨率越高，同样大小的文字和图片岂不是会越小？

不会。我们现在使用的智能手机，不管分辨率多高，他们所展示的界面大小都是差不多的。

CSS像素称为与设备无关的像素(device-independent pixel)，简称DIPs。

 CSS像素与设备像素的关系依赖于屏幕缩放。 以iPhone6为例，dpr为2，缩放设为0.5，则DOM宽度为750，缩放后显示刚好为屏幕宽度375，而总的CSS像素其实是750，与设备像素一致，这样1px的CSS像素，占用的物理像素也是1；而viewport设置缩放为1的理想视口情况下，DOM宽度为375，显示也刚好是屏幕宽度，然而1px的CSS像素，占用的物理像素是2。

页面的缩放系数 = CSS像素 / 设备独立像素。

分辨率resolution是屏幕图像的精密度，是指显示器所能显示的像素的多少。由于屏幕上的点、线、面都是由像素组成的，显示器可显示的像素越多，画面就越精细，同样的屏幕区域内能显示的信息也越多。可以把整个图像想象成一个大型的棋盘，而分辨率的表示方式就是所有经线和纬线交叉点的数目。

CSS全部（包括元素、边距、字体大小）用rem即可，不用px。

### 设备像素比 DPR(device pixel ratio) ＝ 物理像素 / 设备独立像素

通常所说的二倍屏(retina)的dpr是2, 三倍屏是3。

### 点击高亮效果

-webkit-tap-highlight-color : transparent;/\*清除点击高亮效果\*/

### 搜索按钮调用

<!--在移动端点击弹出输入法 enter键会显示“搜索”-->  
<form action="#">  
 <input type="search" placeholder="提示"/>  
</form>

触摸事件：touchstart touchmove touchend

每个触摸事件都包括三个触摸列表：

|  |  |
| --- | --- |
| originalEvent | jquery 封装的事件 |
| touches | 当前位于屏幕上的所有手指的列表 |
| targetTouches | 位于当前DOM元素上的手指的列表 |
| changedTouches | 涉及当前事件的手指的一个列表  For the touchmove event, it is a list of the touch points that have changed since the last event. |

### 过渡和动画结束事件

The transitionend event is fired when a CSS transition has completed. In the case where a transition is removed before completion, such as if the transition-property is removed or display is set to “none”, then the event will not be generated.

dom.addEventListener(**'webkitTransitionEnd'**,**function**(e){ });

dom.addEventListener(**'transitionEnd'**,**function**(e){ });

动画结束后触发

dom.addEventListener(**'webkitAnimationEnd'**,**function**(e){ });

dom.addEventListener(**'animationEnd'**,**function**(e){ });

### 在谷歌的模拟器会出现一个问题：就是touchend的时候可能会丢失事件，所以绑定事件的时候要绑到window上。

### Gesture 事件（不常用）

|  |  |
| --- | --- |
| gesturestart | 当一个手指触摸屏幕之后，第二个手指再触摸屏幕时触发 |
| gesturechange | 当上面的事件触发后立即触发 |
| gestureend | 第二根手指离开屏幕时触发，之后将不会再次触发gesturechange |

在event当中会返回另外两个参数

scale 根据两个手指的滑动距离计算的缩放比例 初始1

rotation根据两个手指的滑动距离计算的旋转角度 初始 0

动画网站：<https://daneden.github.io/animate.css/>

百度地图

iconfont.com

bootstrap.com