11 对比分析 CSS屏幕适配方案

更新时间：2019-06-24 19:21:29

> 学习要注意到细处，不是粗枝大叶的，这样可以逐步学习、摸索，找到客观规律。
>
> —— 徐特立

我们前端在做页面样式开发的时候，明明是按着设计稿 1：1 的比例做出来的，但提测以后就会有 QA 或者 UI 设计师或者产品经理拿着个手机就找你来了（有时候还是几个人一起来）。

“这样式怎么是乱的？”
“这手机上看着字体怎么那么小？”
“这间距不对啊！”
“你这和设计稿不一致啊！”

作为前端开发人员的我们一下就被问懵了，甚至怀疑是不是自己做错了什么。当镇定下来看了一下，原来 QA 手里拿了个多年前的酷派，设计师用的是 iPhone 8 Plus，产品经理手里不知道哪搞来一个 8 点几寸的大屏手机。这时候作为前端开发人员的我们就知道了，这是页面只按着一个尺寸做的，没有搞屏幕适配啊。

我们这次项目做的就是一个移动端的 UI 框架，作为一个公共的框架，它将来会在各种应用上使用，也会在各种各样的设备上运行，所以屏幕适配就是非常重要且必要的了。这一节我们就专门拿出来一部分篇幅，来分析一下 Web 开发中的屏幕适配问题。通过这节的学习，我们要知道为什么做屏幕的适配，了解和屏幕相关的长度计量单位，最重要的是要了解几种常用的屏幕适配方式。

为什么需要适配

现在市面上的设备尺寸没有统一的标准，考虑到设备的用途不同，将来也不会统一。所以我们前端开发人员就要通过屏幕适配，保证我们开发的页面在绝大多数设备上都可以正常显示。这里说的绝大部分而不是全部设备，是因为确实存在个别偏差比较大的设备，如果这部分用户量很少且适配代价会很大，那么忽略这一小部分的设备也是可以的。适配的最终目的，就是为了让设计稿在大部分的移动设备上看起来有一致的展示效果。

超清视频资料 原价一折 加01762238313 更多

计量单位

在讲适配之前，我们要先介绍一下和设备有关的计量单位。知道这些计量单位以及它们之间的换算关系后，才能更好的理解后面要讲的屏幕适配方案。

一、物理像素

现在的设备基本上用的都是液晶屏，这种屏幕的显示原理就是在一块屏幕区域上规则排布很多个发光二极管，系统通过控制这些发光点的颜色和亮度来达到显示的作用。在一块屏幕区域上，每一个发光点，就是我们说的一个物理像素。一些屏幕厂商标识的分辨率指的就是这个物理像素的多少，比如华为 P30 手机标的屏幕像素是 1080×2340，就表示这块屏幕上每行有 1080 个物理像素点，每列上有 2340 个物理像素点。

> **Tips:**
> 1、物理像素点是屏幕的自身属性，设备一出厂这个数值就是固定的了。
> 2、物理像素只是设备的属性，一般和开发者没什么关系，我们的代码也不会直接去操作物理像素。

二、逻辑像素 px

我们在写样式的时候，用的最多的应该就是 px 这个单位，这个 px 就是逻辑像素。现在的设备宽度最小的有 320 个物理像素，最大的能到一两千，所以设备在显示的时候如果直接使用物理像素来显示，那显示效果间的差异是非常大的。比如一个宽 320 像素的按钮，在宽 320 物理像素的屏幕上会占满正个屏幕，而在宽 1242 物理像素的屏幕上，这个按钮只能占差不多四分之一的宽度，这个差距实在太大，并且不容易抹平。

所以，这些设备厂商为了解决这个问题，提出了逻辑像素的概念。逻辑像素的出现就是要尽量抹平不同物理像素设备间的显示差异。还以刚才的例子来说，320 物理像素宽的设备上，我们让 1 逻辑像素 = 1 物理像素，那整个宽度还是能容下 320px 的内容；而对于 1242 宽度的设备，我们让它 1 逻辑像素 = 3 物理像素，这样相当于这块屏幕可以放下 414px 的内容。这时候再定义一个 320像素宽的按钮，在低分辨率上是占满整个宽度，而在高清屏上也能占四分之三左右的宽度，这种显示效果就已经很相近了。

之所以屏幕的逻辑像素没有统一成一种，一是因为屏幕的物理尺寸不一样，二是因为大的屏幕就是要多显示一点东西，不能完全迁就最低配置的屏幕。厂商可以通过控制逻辑像素和物理像素间的换算系数，来保证同样 px 的元素在不同屏幕上显示出来的大小尽量相似。

> **Tips:**
> 高分辨率屏幕上会把多个物理像素当成 1 个逻辑像素来用，但不会和低清屏幕上一样，把 1 个逻辑像素里的多个物理像素显示成同样的颜色。所以高分辨率屏幕虽然和低分辨率屏幕的逻辑像素差不多，但在显示细节和清晰度上还是有区别的，这就是为什么高清屏的显示效果会更细腻。

下面放几组常见手机型号的物理像素和逻辑像素的值供参考：

设备名称	屏幕尺寸	物理像素(宽*高)	逻辑像素(宽*高)
iPhone X	5.8	1125 x 2436	375 x 812
iPhone 8	4.7	750x1334	375x667
iPhone 8+	5.5	1242 x 2208	414 x 736
Samsung Galaxy S8	5.8	1440 x 2960	360 x 740
HUAWEI P10	5.1	1080 x 1920	360 x 640
红米 4 (4, 4X)	5.0	720 x 1280	360 x 640

超清加密 更多视频资料 原价一折 加Q1762238323

三、设计像素

设计像素是指 UI 设计师做设计稿时使用的像素值，通常情况下是按着设备的物理像素来确定设计稿尺寸的。

这里有的同学会有疑问，屏幕逻辑像素最大也就是四百多一点，何必用那么大的设计稿呢？这是因为这些设备虽然逻辑像素差不多少，但同样 1px 的内容，有的设备用 1 个物理像素点显示，而有的用 9 个物理像素点来显示。如果设计稿使用的分辨率够大，那么高分辨率的屏幕里每 px 里的 9 个物理像素可以显示不同的颜色，而低分辨率屏幕里只能把图片里相近的颜色做合并，显示在一个物理像素上。所以用高清的设计稿，最终做出来的显示效果会更细腻。在公司里，设计像素一般用到 1080 的宽度也就差不多了。

四、相对像素 em 和 rem

我们刚才说的 px 单位，它是一种绝对像素，给一个元素多少 px，那么这个元素的宽就定死了。而在 CSS 里还有一种计量方式，就是相对像素。相对像素在响应式开发的时候会显得更灵活。相对像素的单位有 em 和 rem 两种：

1. **相对像素 em**。em 这个单位是相对于父元素的 font-size 字号值，实际就是父元素 font-size 的几倍。这个字号可以是 CSS 指定的，也可以是从父元素的父元素继承来的，如果前两个都没有，就会使用默认值（一般是 16px）。假如父元素指定了 “font-size:20px;” ,子元素使用了 “width: 8em” ，那么子元素的实际宽度就是 160px。我们在使用 em 单位的时候，如果 HTML 层级太深，这个相对关系就容易乱，并且其中一个节点改动字号的话，里面的子元素的 em 值都要重新调整，会造成不便。
2. **相对像素 rem**。rem 和 em 很相似，只不过它是相对于根节点，也就是 HTML 节点的 font-size 属性指定的 px 值。页面里所有用 rem 作为长度单位的元素，就都以 HTML 为准，这样就不用使用 em 那种比较麻烦的换算了，并且当页面里的元素改变字号时，不会对其他元素造成影响。

> **Tips:**
> em 是一个很早就出现的单位，IE6 都可以兼容。但是 rem 是 CSS3 的属性，使用的时候要注意它的兼容性问题。**目前 rem 通常应用在移动端的 WEB 开发上。**

屏幕适配方案

前面铺垫了这么多，终于要进入这节的正题，开始聊一下屏幕的适配方案。刚才讲过，虽然用逻辑像素代替物理像素抹平了一些屏幕差异，但屏幕的宽度依然不一致，所以还是要对屏幕做适配（我们一般所说的适配，大部分都是在关注页面的宽度）。

在做屏幕适配的时候，通常是选择一个大小适中的设备作为基准尺寸，然后再对其他尺寸的设备做适配。比较常见的适配方法有下面几种：

一、百分比方式适配

用百分比做适配的方式是比较原始的一种适配方式。我们通过给元素设置一个百分比的值，让这个元素自动调整它的大小或位置等。这种方式出现的最早，也是在使用的时候最方便的。Bootstrap 框架在很多地方就采用了这种方案。

百分比适配有如下优点：

1. 简单便捷，不用关心元素的绝对宽度。
2. 基本没有兼容性问题。

但是这种适配方式缺点也是很明显的：

超清加密 更多 视频资料 原价一折 加Q1762238323

1. 使用百分比时候，很容易不是整数，比如设计稿是一行里三个元素等分距离，如果用百分比，那每个的宽度就是 33.33333%。这样百分比是很长的小数，还要自己计算百分比的值，最严重的是经过百分比的计算后有可能最后整行宽度会留有空白。
2. 百分比的值是相对于父元素来计算的。子元素宽度的百分比是相对父元素宽度，子元素的高度的百分比是相对于父元素的高度，子元素的内外边距都是相对父元素的宽度，这些属性是相对谁的容易弄混。
3. 当同一行里同时存在定宽元素和自适应的元素时，使用百分比就不太利于分配空间。
4. 如果父子元素都脱离了文档流，那么它父元素的宽度就可能没有了，这时候子元素用百分比就会失效。

通过用百分比的方式适配的话，在一些情景下是非常方便的，比如一行内等距放 4 个元素，就可以每个元素 25% 的宽度，一条语句就可以搞定，但考虑到百分比适配的缺点，并不推荐所有地方都强制使用百分比进行适配。我们的项目中，用到百分比适配比较多的是需要撑开整个屏幕的情况，如头部标题栏、导航和全屏宽的按钮等设置宽度的时候，就可以直接给元素“width: 100%；”。

二、多套样式分段适配

我们要说的第二种适配方案就是制作多套样式，然后根据屏幕宽度来选择合适的那套样式。这种适配方式一般是在 HTML 或 body 上添加一些 class 来控制。

我们以京东的 PC 端页面为例，当浏览器宽度大于 1190px 时，就会在 HTML 标签上加上 o2_wide 的 class，这时候页面的主要内容区也是定宽 1190px 的一个框子。当浏览器小于 1190px时，就会把 o2_wide 变成 o2_mini，这时候主要显示区的宽度就变成了 990px。京东实际上是实现了两种宽度的页面，然后通过 JS 判断页面宽度来选择应用哪一套。

淘宝的 PC 端会做的更精细一点，会把适配方案分成 990px、1024px、1190px、1279px、1365px、1440px 这些档，对每一种宽度都做了一套样式。然后也是通过 JS 控制 body 上的 class 来控制生效的样式。

上面两个例子都是使用 JS 来控制 DOM 元素上的 class 来实现的，在 CSS3 中还有一种控制方式叫媒体查询。媒体查询可以直接用 CSS 根据浏览器页面宽度判断使用哪套样式方案。原理和上面的一样，只不过用媒体查询代替了上面的 JS 逻辑。

这种适配方案有如下的优点：

1. 更灵活，比如可以借助根节点上的 class 来控制更多的内容。比如在宽屏幕上指定一行显示 4 个元素（每个宽 25%），如果在窄屏幕上可以让一行显示 5 个（每个宽 20%）。
2. 页面宽度是固定的几档，测试的时候比较方便。
3. 使用 JS 控制 class 的方式一般不会出现兼容性问题，通常应用在有兼容性要求的 PC 端页面。

但这种方式的不足也比较明显：

1. 需要制作多套设计稿，也需要多套样式代码，工作量比较大。
2. 要维护多套样式，有改动的时候比较麻烦。
3. 不能在所有设备下都撑满全屏，不适合移动端。

这种适配方案在适配时一般都是设计一套基准宽度的样式，其他宽度下的样式用覆盖的方式来控制。

三、Rem 方式适配

最后就是要说 Rem 方式的适配了。我们之前讲了 Rem 的用法，在做屏幕适配的时候，我们通过控制 HTML 的 font-size 属性就可以了。

超清加密 更多 视频资料 原价一折 加Q1762238325

第一步，限定基准尺寸屏幕下的 HTML 字号。我通常会给 HTML 节点设置上 20px 的字体，这样做 px 到 rem 的转换时比较容易，比如我需要 18px 的宽度，那就可以很快的把 18px 换算成 .9rem。

第二步，在其他尺寸屏幕上，修改 HTML 字号值。在 rem 适配中，因为只需要控制 HTML 字号就可以达到屏幕适配的目的。而在变动这个字号的时候可以像百分比适配那样做连续的变动，直接按着屏幕宽度和标准尺寸的比例关系修改字号即可；也可以用分段适配的方式，在某一个宽度范围内用哪一种字号。对字号的控制还是比较灵活的。

Rem 适配的优点：

1. Rem 的参考值只有一个，就是 HTML 字体大小，所以不会像百分比适配那样要明确参考谁。
2. Rem 使用的范围比较广，元素的高度宽度、border 宽度、字号大小、间距和偏移量等都可以用 rem 做单位
3. Rem 适配的时候不需要做多套样式，可以直接按比例改变 HTML字体的大小。

Rem 适配的缺点：

1. Rem 是 CSS3 的属性，只有比较新的浏览器上可以使用，一般都是用在移动端。目前大厂的 H5 页面，基本都在使用 Rem 方式做屏幕适配了。
2. Rem 最后换算出来的值还是 px，所以对于按比例分配空间这种需求是没法满足的，需要和其他适配方式混合使用。

四、弹性布局适配

最后说一下弹性布局，这也是 CSS3 里出现的一种适配方式。弹性布局融合了百分比布局和固定宽度布局的优点，并且用起来更加简单。弹性布局下盒模型可以不用指定宽度，直接让盒子把可利用空间填满，这样既不用去计算盒子的宽度，也不用担心固定宽度元素和需要适配的元素在同一行显示的问题了。

弹性布局的优点有：

1. 可以弹性地利用页面空间，且不需要计算宽度比例。
2. 每行元素个数变化时，不用重新调整元素的宽度。
3. 弹性布局中的高级属性能带来更多的功能，比如可以指定元素的排列方式和排列顺序等，这些功能用传统的 CSS 是不太好实现的。

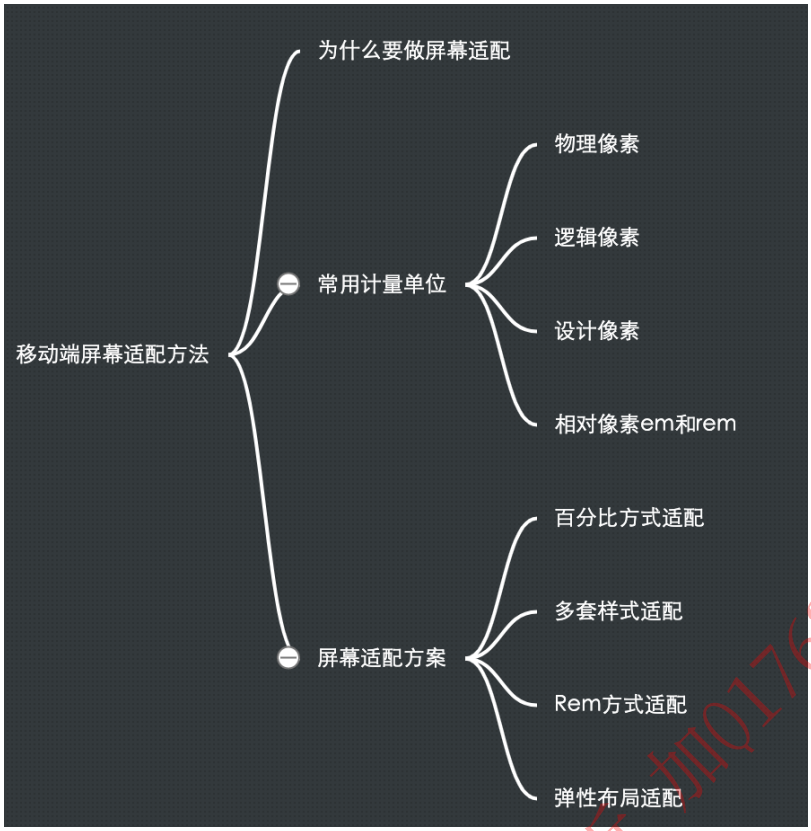
而弹性布局最重要的缺点就是兼容性问题，在移动端有些弹性布局的属性都还没有得到完全的支持，在使用时一定要做详细的测试。这个项目中的标题栏、底部导航、网格等组件中，都使用了弹性布局，到时会带着同学们更详细地了解这种布局方式。

小结

这一节介绍了几种长度单位，然后讲了几种常见的布局方式。这几种布局方式之间并不是对立的关系，而是可以融合放在一起的。比如简单的宽度自适应且比例是整数的，用百分比是最方便的；如果是 PC 端使用，那么分段适配更合适；如果在移动端使用，那么 Rem 一定是主力；如果定宽元素和需要适配元素放在一起，那弹性布局是合适的。

这一节内容的结构如下：

超清加密 更多 视频资料 原价一折 加Q1762238323



我们这个项目中这几种适配方式都有应用。Rem 方式布局是这个项目的主要适配方式；需要撑满整个宽度的时候，简单的会直接用百分比，而复杂一点的就会用到弹性布局；多套适配方式用的比较少，改变 HTML 的字体大小的时候用到了多套适配的思路。我们对各种布局方式就介绍到这里，在后面开发的过程中遇到了还会再提这些概念，并且还会结合着开发过程做详细的讲解。

我们这一节的内容就到这里，下一节我们将介绍一下样式开发的规范。概念最多的一章就快要过去了，同学们挺住！

10 页面渲染机制（二）之渲染树的构建及渲染过程
12 CSS规范：写出一套优雅的CSS 代码

超清加密 更多 视频资料 原价一折 加01762238323