前端布局方案汇总

## 1. 静态布局 Static Layout

**布局概念**：最传统的Web布局方案，内容区域设置固定宽/最小宽，单位是px，在不同大小屏幕下看到的内容一样，屏幕不够宽就出现横向滚动条，多用于纯PC端网站。

例如：PC端百度首页<https://www.baidu.com/>



**优点**：采用css2以前的写法，没有兼容性问题，布局简单，易于实现。

**缺点**：不适用于移动端，需另写一套。

## 2. 流式布局 Fluid / Liquid Layout

**布局概念**：也叫百分比布局，元素宽度设置为百分比，不同屏幕下显示布局一样，但元素宽度不一样。在Web前端开发的早期历史上，用来应对不同尺寸的PC屏幕，多用于屏幕尺寸跨度小，页面布局简单的网页情况，现在在移动端网站也会用到。（现在的移动端多被弹性布局取代）。

例如：某手机端官网<http://www.ccdi.com.cn/m>

**优点**：元素的宽高用百分比做单位，元素宽高按屏幕大小调整，布局不发生变化。

**缺点**：字体单位多用PX不会根据屏幕大小调整，屏幕尺度跨度过大的情况下，页面不能显示不协调。

## 3. 自适应布局 Adaptive Layout

**布局概念**：分别为不同大小屏幕定义布局，即创建多个静态布局，每个静态布局对应一个屏幕分辨率范围。使用 @media 媒体查询给不同尺寸和介质的设备切换不同的样式。常和流式布局一起在PC端使用。

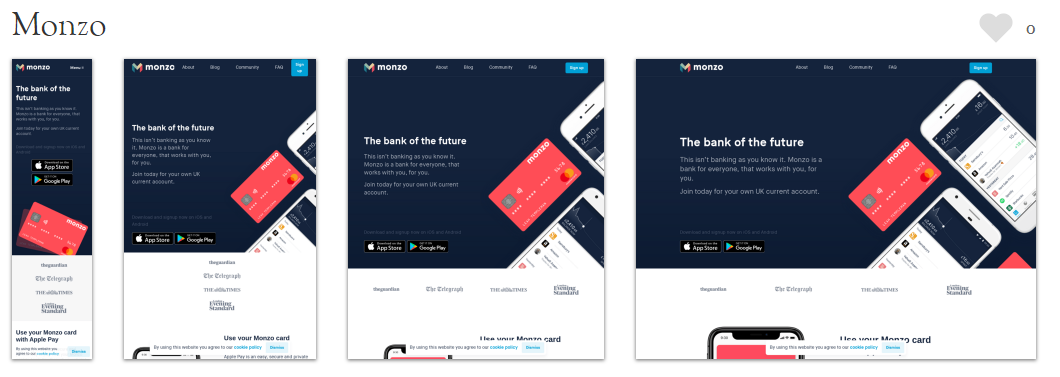
**优点**：可以明确的给某些尺寸的的屏幕设计页面效果。

**缺点**：仅能支持特定尺寸的屏幕，实际使用效果不够精细。

## 4. 响应式布局 Responsive Layout

布局概念：通过流式布局、自适应布局、弹性布局等布局方式，实现一个页面在所有的终端（各种尺寸的PC端和移动端）上都显示出理想设计好的效果。页面尺寸多用百分比写法。可适应屏幕大小跨度大的设备，仅适用于结构简单的页面，对于业务逻辑复杂的页面，工作量大。

例如：某网站<https://monzo.com/，https://www.reactjscn.com/>，https://beta.ionicframework.com/docs/



**优点**：一套代码同时适应移动和PC设备，并且在各种屏幕大小设备下都能显示理想。

**缺点**：要匹配足够多的屏幕大小，设计和开发、测试工作量太大，只能适应主流大小屏幕。

## 5. 弹性布局 rem / em

**布局概念**：通过计算设计和屏幕尺寸，设置不同屏幕下根元素<html>的font-size属性，页面单位使用rem，当屏幕尺寸改变时，页面元素和字体的大小会随之缩放。通常使用的尺寸单位是rem和em，其中rem偏多。因为rem是相对于根元素<html>的单位。目前这种干布局方式在移动端使用较多。

浏览器的默认字体高度一般为16px，即1em:16px。设置1rem便是10px便于计算，如下：

html {font-size: 62.5%;/\*10 ÷ 16 × 100% = 62.5%\*/}

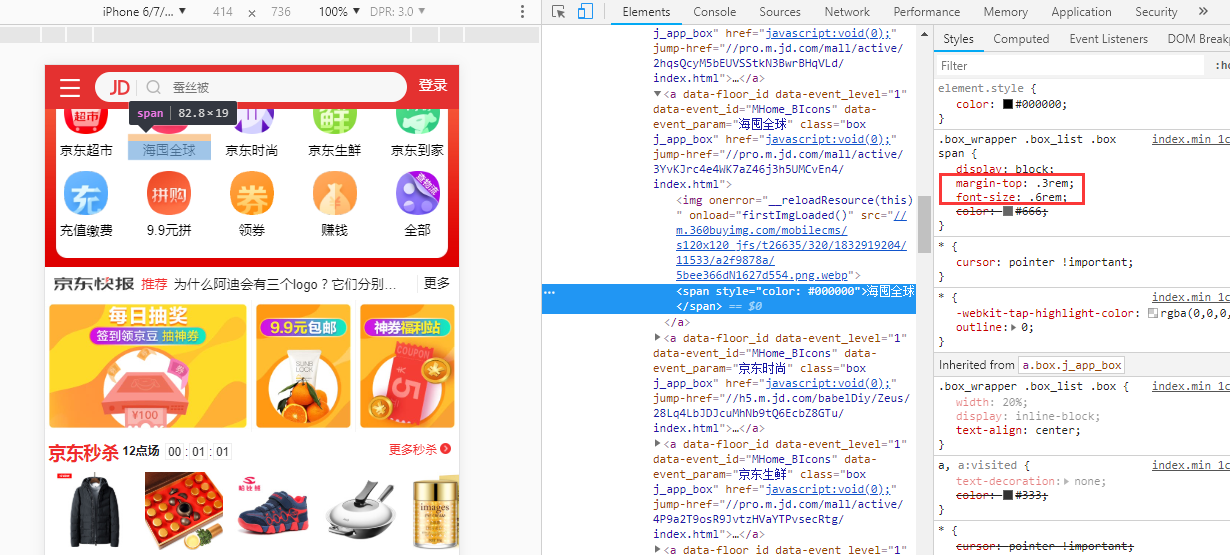
body {font-size: 1.4rem;/\*1.4 × 10px = 14px \*/}

h1 { font-size: 2.4rem;/\*2.4 × 10px = 24px\*/}

**优点**：理想状态是所有屏幕的高宽比和最初的设计高宽比几乎一样。

**缺点**：不适用于屏幕尺寸跨度较大的情况。

例如：京东手机端网站 https://m.jd.com/



某个对比网站<http://wow.techbrood.com/fiddle/1753>

Viewport

viewport 是用户网页的可视区域。viewport 翻译为中文可以叫做"视区"。

手机浏览器是把页面放在一个虚拟的"窗口"（viewport）中，通常这个虚拟的"窗口"（viewport）比屏幕宽，这样就不用把每个网页挤到很小的窗口中（这样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不同部分。

一个常用的针对移动网页优化过的页面的 viewport meta 标签大致如下：

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

width：控制 viewport 的大小，可以指定的一个值，如 600，或者特殊的值，如 device-width 为设备的宽度（单位为缩放为 100% 时的 CSS 的像素）。

height：和 width 相对应，指定高度。

initial-scale：初始缩放比例，也即是当页面第一次 load 的时候缩放比例。

maximum-scale：允许用户缩放到的最大比例。

minimum-scale：允许用户缩放到的最小比例。

user-scalable：用户是否可以手动缩放。

