

网页基本布局方式（5 种）

一、静态布局（static layout）

即传统 Web 设计，网页上的所有元素的尺寸一律使用 px 作为单位。

1、布局特点

不管浏览器尺寸具体是多少，网页布局始终按照最初写代码时的布局来显示。常规的 pc 的网站都是静态（定宽度）布局的，也就是设置了 min-width，这样的话，如果小于这个宽度就会出现滚动条，如果大于这个宽度则内容居中外加背景，这种设计常见于 pc 端。

2、设计方法

PC：居中布局，所有样式使用绝对宽度/高度(px)，设计一个 Layout，在屏幕宽高有调整时，使用横向和竖向的滚动条来查阅被遮掩部分；

移动设备：另外建立移动网站，单独设计一个布局，使用不同的域名如 wap.或 m.。

优点：这种布局方式对设计师和 CSS 编写者来说都是最简单的，亦没有兼容性问题。

缺点：显而易见，即不能根据用户的屏幕尺寸做出不同的表现。当前，大部分门户网站、大部分企业的 PC 宣传站点都采用了这种布局方式。固定像素尺寸的网页是匹配固定像素尺寸显示器的最简单办法。但这种方法不是一种完全兼容未来网页的制作方法，我们需要一些适应未知设备的方法。

二、流式布局（Liquid Layout）

流式布局（Liquid）的特点（也叫"Fluid"）是页面元素的宽度按照屏幕分辨率进行适配调整，但整体布局不变。代表作栅栏系统（网格系统）。

网页中主要的划分区域的尺寸使用百分数（搭配 min-*、max-*属性使用），例如，设置网页主体的宽度为 80%，min-width 为 960px。图片也作类似处理（width:100%, max-width 一般设定为图片本身的尺寸，防止被拉伸而失真）。

1. 布局特点

屏幕分辨率变化时，页面里元素的大小会变化而但布局不变。【这就导致如果屏幕太大或者太小都会导致元素无法正常显示。

2. 设计方法

使用百分比定义宽度，高度大都是用 px 来固定住，可以根据可视区域 (viewport) 和父元素的实时尺寸进行调整，尽可能的适应各种分辨率。往往配合 max-width/min-width 等属性控制尺寸流动范围以免过大或者过小影响阅读。

这种布局方式在 Web 前端开发的早期历史上，用来应对不同尺寸的 PC 屏幕（那时屏幕尺寸的差异不会太大），在如今的移动端开发也是常用布局方式，但缺点明显：主要的问题是如果屏幕尺度跨度太大，那么在相对其原始设计而言过小或过大的屏幕上不能正常显示。因为宽度使用百分比定义，但是高度和文字大小等大都是用 px 来固定，所以在大屏幕的手机下显示效果会变成有些页面元素宽度被拉的很长，但是高度、文字大小还是和原来一样（即，这些东西无法变得“流式”），显示非常不协调

三、自适应布局 (Adaptive Layout)

自适应布局的特点是分别为不同的屏幕分辨率定义布局，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。改变屏幕分辨率可以切换不同的静态局部（页面元素位置发生改变），但在每个静态布局中，页面元素不随窗口大小的调整发生变化。可以把自适应布局看作是静态布局的一个系列。

1、布局特点

屏幕分辨率变化时，页面里面元素的位置会变化而大小不会变化。

2、设计方法

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。在优秀的响应范围设计下可以给适配范围内的设备最好的体验，在同一个设备下实际还是固定的布局。

四、响应式布局 (Responsive Layout)

随着 CSS3 出现了媒体查询技术，又出现了响应式设计的概念。响应式设计的目标是确保一个页面在所有终端上（各种尺寸的 PC、手机、手表、冰箱的 Web 浏览器等等）都能显示出令人满意的效果，对 CSS 编写者而言，在实现上不拘泥于具体手法，但通常是糅合了流式布局+弹性布局，再搭配媒体查询技术使用。——分别为不同的屏幕分辨率定义布局，同时，在每个布局中，应用流式布局的理念，即页面元素宽度随着窗口调整而自动适配。即：创建多个流体式布局，分别对应一个屏幕分辨率范围。可以把响应式布局看作是流式布局 and 自适应布局设计理念的融合。

响应式几乎已经成为优秀页面布局的标准。

1. 布局特点

每个屏幕分辨率下面会有一个布局样式，即元素位置和大小都会变。

2. 设计方法

媒体查询+流式布局。通常使用 @media 媒体查询 和网格系统 (Grid System) 配合相对布局单位进行布局, 实际上就是综合响应式、流动等上述技术通过 CSS 给单一网页不同设备返回不同样式的技术统称。

优点: 适应 pc 和移动端, 如果足够耐心, 效果完美。

缺点: (1) 媒体查询是有限的, 也就是可以枚举出来的, 只能适应主流的宽高。
(2) 要匹配足够多的屏幕大小, 工作量不小, 设计也需要多个版本。

响应式页面在头部会加上这一段代码:

五、弹性布局 (rem/em 布局)

1. rem/em 区别: rem 是相对于 html 元素的 font-size 大小而言的, 而 em 是相对于其父元素。

2. 使用 em 或 rem 单位进行相对布局, 相对百分比更加灵活, 同时可以支持浏览器的字体大小调整和缩放等的正常显示, 因为 em 是相对父级元素的原因没有得到推广。【中国站点制作网页的时候, 习惯用 CSS 强制定义字体大小, 保证每个人都看到一致的效果, 包括网易、搜狐这些门户网站在内的大部分站点, 用的都是绝对单位 px (像素)。但是, 如果从网站易用性方面考虑, 字体大小应该是可变的, 一些视力不是那么好的人需要放大字体才能看得清页面内容。然而, 占据大部分浏览器市场的 IE 无法调整那些使用 px 作为单位的字体大小。国外人士非常重视网站的易用性, 相当一部分外国站点已经使用 em 作为字体单位。

3. 这类布局的特点是, 包裹文字的各元素的尺寸采用 em/rem 做单位, 而页面的主要划分区域的尺寸仍使用百分数或 px 做单位 (同「流式布局」或「静态/固定布局」)。早期浏览器不支持整个页面按比例缩放, 仅支持网页内文字尺寸的放大, 这种情况下。使用 em/rem 做单位, 可以使包裹文字的元素随着文字的缩放而缩放。

4. 浏览器的默认字体高度一般为 16px, 即 1em:16px, 但是 1:16 的比例不方便计算, 为了使单位 em/rem 更直观, CSS 编写者常常将页面跟节点字体设为 62.5%, 比如选择用 rem 控制字体时, 先需要设置根节点 html 的字体大小, 因为浏览器默认字体大小 16px*62.5%=10px。这样 1rem 便是 10px, 方便了计算。

5. 用 em/rem 定义尺寸的另一个好处是更能适应缩进/以字体单位 padding 或 margin / 浏览器设置字体尺寸等情况 (因为 em/rem 相对于字体大小, 会同步改变)。例如: `p{ text-indent: 2em; }`。

6. 使用 rem 单位的弹性布局在移动端也很受欢迎。

7. 其实在移动端使用所谓的弹性布局，是比较勉强的。移动端弹性布局流行起来的原因归根结底是 **rem** 单位对于（根据屏幕尺寸）调整页面的各元素的尺寸、文字大小时比较好用。其实，使用 **vw**、**vh** 等后起之秀的单位，可以实现完美的流式布局（高度和文字大小都可以变得“流式”），弹性布局就不再必要了。

结论

- 1.如果只做 **pc** 端，那么静态布局（定宽度）是最好的选择；
- 2.如果做移动端，且设计对高度和元素间距要求不高，那么弹性布局（**rem+js**）是最好的选择，一份 **css**+一份 **js** 调节 **font-size** 搞定；
- 3.如果 **pc**，移动要兼容，而且要求很高那么响应式布局还是最好的选择，前提是设计根据不同的高宽做不同的设计，响应式根据媒体查询做不同的布局。