

## 多列布局

笔记本: 7.18多列布局

创建时间: 2020/7/20 9:39

更新时间: 2020/7/20 17:29

作者: 805427154@qq.com

URL: file:///D:/千峰前端培训/H5-2005前端班/第14天/13.弹性盒布局、媒体查询/资料/分栏...

---

## 多列布局

css3多列布局可以自动将内容按指定的列数排列，这种特性实现的效果和报纸、杂志类排版非常相似。

### 1、网页瀑布流布局 (columns) :

column-width (添加给子元素) : 栏目宽度

说明: 定义每列列宽; 类似于最小宽度min-width; auto 自适应;

column-count (添加给父元素) : 栏目列数

说明: 定义分列列数; 最多列数, auto自适应 (由列宽、容器宽和列间距决定), 也可固定。

column-gap (添加给父元素) : 栏目间距

说明: 定义列间距; 不能为负数;

column-rule (添加给父元素) : 栏目间隔线样式 (大小、样式、颜色)

说明: 定义列边框; 与定义边框一样: 2px dashed #ccc;

**注意: 写的时候要把三个属性值都写上, 否则不能生效**

column-span (添加给子元素) : 规定元素应横跨多少列

说明: 定义多列布局中子元素的跨列效果; 通常用于标题;

属性: none: 不跨列; all: 跨所有列; 数值: 数值为多少则指定跨多少列 (1:指定跨1列)

break-inside: avoid断点 (添加给子元素)

### 2、关于列宽度、高度的平衡:

- 如果没有设置栏宽或设置的宽度小于默认平分的宽度, 那么默认平分;
- 如果设置的宽度大于默认平分的宽度, 则可能减少列的数量, 依旧平分;

- 父元素设置的高度小于默认生成的高度，则会造成多列出现；
- 父元素设置的高度大于默认生成的高度，则默认生成的高度不会被改变。

### 3、css瀑布流和JS瀑布流的区别在于：

css瀑布流的排列顺序是垂直排列，js瀑布流的排列顺序是横向排列。

## meta标签定义

许多智能手机都使用了一个比实际屏幕尺寸大很多的虚拟可视区域viewpoint(布局视口)，主要目的就是让页面在智能手机端阅读时不会因为实际可视区域变形。所以你看到的页面还是普通样式，即一个全局缩小后的页面。为了让智能手机能根据媒体查询匹配对应样式，让页面在智能手机中正常显示，特意添加了一个meta标签。这个标签的主要作用就是让智能手机浏览页面时能进行优化，并且可以自定义界面可视区域的尺寸和缩放级别。

### 1、三种viewpoint视口

**布局视口：**以屏幕分辨率为基准，实际上布局视口的宽度要比屏幕宽出很多

**视觉视口：**用户看到的网站展示区域，一般视觉视口和设备宽度一致。并且它的CSS像素的数量会随着用户缩放而改变。

**理想视口：**为了使网站在移动端有最理想的浏览和阅读宽度而设定。需要手动添写meta视口标签，一般视口大小都设置为设备大小。(meta:vp)

### 2、设置移动端标准视口

快捷键：meta:vp

```
<meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=1, user-scalable=no" />
```

width：可视区域的宽度；（控制viewpoint的宽度，可以是固定值，也可以是device-width设备宽度）

height：可视区域的高度；

device-width：设备屏幕分辨率的宽度值

initial-scale: 初始的缩放比例 (0-10.0) , 取值为1时页面按实际尺寸显示, 无任何缩放。

minimum-scale: 允许用户缩放到的最小比例

maximum-scale: 允许用户缩放到的最大比例

user-scalable: 设定用户是否可以缩放 (yes/no)

可以写成: <meta name="viewport" content="width=device-width,initial-scale=1.0" />

## 媒体查询

### 备: 知识拓展

- 移动端设计稿的宽度一般都是640px/750px
- bootstrap的屏幕分界点可以让页面类型随着页面宽度的改变而发生变化。

**自适应:** 不同大小设备呈现同样的页面效果, 只是文字、图片等的大小不一样, 但是相对位置一样。例如: 百分比布局、弹性盒布局flex、分栏布局。

**响应式:** 同一页面在不同大小设备(窗口)可能呈现不一样的页面效果, 自适应布局+媒体查询即可实现。(应用: 相对简单、内容较少的页面)

### 1、CSS3响应式布局

#### 屏幕分界点:

超小屏幕xs: (移动设备) 768px以下;

小屏设备sm: 768px-992px;

中屏设备md: 992px-1200px;

宽屏设备lg: 1200px以上。

### 2、语法: 最小宽度 (最大宽度语法一样, 但是max-width的书写要从大到小。)

语法: @media screen and (条件) {css语法}

**注意:**

1、若当前页面宽度大于min-width，则样式生效，所以媒体查询min-width要从小写到大；

2、若当前页面宽度小于max-width，则样式生效，所以媒体查询max-width要从大到小。

3、min/max-width指的是页面宽度;min/max-device-width指的是设备屏幕宽度

例子：

```
div {  
    height: 200px;  
    background: pink;  
}  
@media screen and (min-width: 768px) {  
    div{background: red;}  
}  
@media screen and (min-width: 992px) {  
    div{background: green;}  
}  
@media screen and (min-width: 1200px) {  
    div{background: blue;}  
}
```