# **第16课 移动端布局**

## **一、Css3分栏布局**

ss3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.分栏布局属性**

* **columns**：列宽 列数
  + **column-width** 栏目宽度
  + **column-count** 栏目列数
* **column-gap** 栏目间距，不能为负数
* **column-rule** 栏目间隔线样式，与定义边框一样
* **column-span**: 定义多列布局中子元素的跨列效果，通常用于标题
  + 1: 指定跨1列
  + all: 跨所有列

### **2.列宽度、高度的平衡**

* 如果没有设置栏宽或者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。
* 小于默认生成的高度，则会造成多列出现；
* 当设置高度大于默认生成的高度，则默认生成的高度不会被改变。

## **弹性盒flex-box**

## **1.概述：**Flexbox从本质上就是一个盒模型（Box-model）的延伸，用来实现更复杂 的版面布局。它能让我们的实现响应式布局更加优雅 2.**css属性语法**：**.box{display: flex;}**

## flexbox_axis**PS：弹性盒括中的两个对象：容器、子项目容器设为Flex布局以后，子项目的float、clear和 vertical-align属性将失效。**

## **扩展：**参照右图，弹性盒子按照宽和高分出了以下 8 点： 1. 水平的主轴: main axis 2. 垂直的纵轴: cross axis 3. 纵轴的开始位置和边框的交点: cross start 4. 纵轴的结束位置和边框的交点: cross end 5. 主轴的开始位置和边框的交点: main start 6. 主轴的结束位置和边框的交点: main end 7. 单个项目占据主轴的空间距离: main size 8. 单个项目占据纵轴的空间距离: cross size

## **3.应用在容器（父元素）上的属性**

* **flex-direction：主轴的方向**（即**项目的排列方向**）
  + row（默认） 主轴为水平方向，起点在左端。
  + row-reverse 主轴为水平方向，起点在右端。
  + column 主轴为垂直方向，起点在上沿。
  + column-reverse 主轴为垂直方向，起点在下沿。
* **flex-wrap：**如果一条轴线排不下，**如何换行**。
  + nowrap（默认）：不换行。
  + wrap：换行，第一行在上方。
  + wrap-reverse：换行，第一行在下方
* **flex-flow：**是flex-direction和flex-wrap属性简写形式，默认值为row nowrap。
* **justify-content：**项目在**主轴上的对齐方式**。
  + flex-start（默认） 左对齐
  + flex-end 右对齐
  + center 居中
  + space-between 两端对齐，项目d之间的间隔都相等。
  + space-around 每个项目两侧的间隔相等。所以，项目之间的间隔比项目与边框的间隔大一倍。
* **align-items：**项目在**交叉轴上的对齐方式**。
  + flex-start 交叉轴的起点对齐。
  + flex-end 交叉轴的终点对齐。
  + center 交叉轴的中点对齐。
  + baseline 项目的第一行文字的基线对齐。
  + stretch（默认） 如果项目未设置高度或设为auto，将占满整个容器的高度。
* **align-conten**t：**多根轴线的对齐方式**。项目只有一根轴线，该属性不起作用。
  + flex-start 与交叉轴的起点对齐。
  + flex-end 与交叉轴的终点对齐。
  + center 与交叉轴的中点对齐。
  + space-between 与交叉轴两端对齐，轴线之间的间隔平均分布。
  + space-around 每根轴线两侧的间隔都相等。所以，轴线之间的间隔比轴线与边框的间隔大一倍。
  + stretch（默认值） 轴线占满整个交叉轴。

### **4.应用在项目（子元素）上的属性**

* **order: integer**定义**项目的排列顺序**，属性值为整数（默认0）。数值越小，排列越靠前。

**Flex：**

flex属性是flex-grow, flex-shrink 和 flex-basis的简写，默认值为0 1 auto。后两个属性可选。

**.item { flex: none || auto || <flex-grow> <flex-shrink> <flex-basis>}**

PS：建议优先使用这个属性，而不是单独写三个分离的属性，因为浏览器会推算相关值。  
该属性有两个快捷值：auto(1 1 auto) 和 none(0 0 auto)。

* + flex-grow: number  
    定义项目的放大比例，如果项目没占满容器（存在剩余空间）按比例分配剩余空间。默认为0，即如果存在剩余空间，也不放大
  + flex-shrink: number  
    定义了项目的缩小比例（与flex-grow相反），默认为1，即如果空间不足，该项目将缩小
  + flex-basis  
    定义了在分配多余空间之前，项目占据的主轴空间（main size），可以理解为我们给子元素设置的宽度。浏览器根据这个属性，计算主轴是否有多余空间。它的默认值为auto，即项目的本来大小

.item {flex-basis: <length> | auto; /\* default auto \*/}

**align-self**

允许单个项目有与其他项目不一样的对齐方式，可覆盖align-items属性。默认值为auto，表示继承父元素的align-items属性，**所有属性跟容器的align-items相同**

## **三、媒体查询media query**

将已有的开发技巧（弹性盒、弹性图片、媒体查询等）整合起来，命名为****“响应式布局”****，所谓响应式布局，是一种针对任意设备对网页内容进行“完美”布局的一种显示机制。  
简言之，是一个网站能够兼容多个终端的布局。

### **1.使用方法：**

外联CSS语法

**<link rel="stylesheet" href="wide.css" media="screen and (min-width:1024px)" />**

**<link rel="stylesheet" href="mobile.css" media="screen and (max-width:320px)" />**

内嵌样式的语法

<style>

@media all(screen) and (min-width:500px) { … }

@media (orientation: portrait) { … }

</style>

### **2.媒体查询语句**

一个 Media Query 包含一种媒体类型，如果媒体类型没有指定，那么就是默认类型all

**@media (max-width: 600px){...}**

可以使用 and 表示同时满足这两者时生效

**@media handheld and (min-width:20em) and (max-width:50em){...}**

逗号 , 被用来表示 并列 和 或者

**@media handheld and (max-width:20em),screenand(max-width:30em){...}**

排除符合表达式的设备

**@media not screen and (orientation:portrait){...}**

### **3.媒体介质类型**

* all 所有设备
* screen 电脑显示器
* print 打印用纸或打印预览视图
* handheld 便携设备

### **4.设备特性(media feature)**

* width/height: 定义输出设备中的页面可见区域宽度/高度
* device-width/device-heigh: 定义输出设备的屏幕可见宽度/高度
* orientation: 定义’height’是否大于或等于’width’。
  + portrait：是（竖屏），
  + landscape：否（横屏）

### **5.优势**

1. 多终端视觉和操作体验非常风格统一
2. 兼容当前及未来新设备
3. 响应式web设计中的大部分技术可以用在WebApp开发中
4. 节约了开发成本，维护成本也降低很多

### **6.不足**

1. 兼容性：低版本浏览器兼容性有问题
2. 移动带宽流量：相比较手机定制网站，流量稍大，
3. 但比较加载一个完整pc端网站显然是小得多
4. 代码累赘，会出现隐藏无用的元素，加载时间加长
5. 兼容各种设备工作量大

### **7.自适应与响应式区别**

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可能呈现不一样的页面效果，自适应布局+媒体查询即实现。

**五、[百分比布局](file:///D:/%E4%B8%8A%E8%AF%BE%E8%B5%84%E6%96%99/%E7%AC%AC%E4%B8%80%E9%98%B6%E6%AE%B5/%E7%AC%AC%E5%9B%9B%E5%91%A8%20%E7%A7%BB%E5%8A%A8%E7%AB%AF%E5%B8%83%E5%B1%80/%E8%B0%A2%E7%89%88/day17%E7%A7%BB%E5%8A%A8%E7%AB%AF%E5%B8%83%E5%B1%802/doc/%E7%A7%BB%E5%8A%A8%E7%AB%AF%E5%B8%83%E5%B1%802.html" \l "_2)**

优缺点：好处：充分发挥大手机的优势——显示内容越多

缺点：屏幕过大，间距过大，比例失调。

**六、[rem布局布局](file:///D:/%E4%B8%8A%E8%AF%BE%E8%B5%84%E6%96%99/%E7%AC%AC%E4%B8%80%E9%98%B6%E6%AE%B5/%E7%AC%AC%E5%9B%9B%E5%91%A8%20%E7%A7%BB%E5%8A%A8%E7%AB%AF%E5%B8%83%E5%B1%80/%E8%B0%A2%E7%89%88/day17%E7%A7%BB%E5%8A%A8%E7%AB%AF%E5%B8%83%E5%B1%802/doc/%E7%A7%BB%E5%8A%A8%E7%AB%AF%E5%B8%83%E5%B1%802.html" \l "rem)：**等比缩放布局

**七、布局[单位](file:///D:/%E4%B8%8A%E8%AF%BE%E8%B5%84%E6%96%99/%E7%AC%AC%E4%B8%80%E9%98%B6%E6%AE%B5/%E7%AC%AC%E5%9B%9B%E5%91%A8%20%E7%A7%BB%E5%8A%A8%E7%AB%AF%E5%B8%83%E5%B1%80/%E8%B0%A2%E7%89%88/day17%E7%A7%BB%E5%8A%A8%E7%AB%AF%E5%B8%83%E5%B1%802/doc/%E7%A7%BB%E5%8A%A8%E7%AB%AF%E5%B8%83%E5%B1%802.html" \l "_7)**

**1.Px：**百分比

**2.Em**：相对单位，基于自身字体大小font-size来计算，所以不同的元素 1em代表的具体大小可能不同（部分浏览器默认字体是16px）。

**3.Rem：**相对单位，可理解为”root em”, 相对根节点html的字体大小 来计算，CSS3新加属性，chrome/firefox/IE9+支持

**4.vw** viewport width，视窗宽度，1vw等于视窗宽度的1%

Vh：viewport height，视窗高度，1vh等于视窗高度的1%。

使用vw/vh，需设置meta缩放比1:1  
**IE9+局部支持，chrome/firefox/safari/opera支持，ios safari 8+支持，android browser4.4+支持，chrome for android39支持**

**4.[viewport视口](file:///D:/%E4%B8%8A%E8%AF%BE%E8%B5%84%E6%96%99/%E7%AC%AC%E4%B8%80%E9%98%B6%E6%AE%B5/%E7%AC%AC%E5%9B%9B%E5%91%A8%20%E7%A7%BB%E5%8A%A8%E7%AB%AF%E5%B8%83%E5%B1%80/%E8%B0%A2%E7%89%88/day17%E7%A7%BB%E5%8A%A8%E7%AB%AF%E5%B8%83%E5%B1%802/doc/%E7%A7%BB%E5%8A%A8%E7%AB%AF%E5%B8%83%E5%B1%802.html" \l "viewport)**

移动设备上的viewport就是设备的屏幕上能用来显示我们的网页的那一块区域

* 各大手机的viewport大小
* 布局视口（layout viewport）

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，以上图片显 示的尺寸就是 layout viewport

* 视觉视口（visual viewport）  
  layout viewport 的宽度是大于浏览器可视区域的宽度的，所以我们还需要一个viewport来代表 浏览器可视区域的大小，这个viewport叫做visual viewport。在js中可以通过window.innerWidth来获取
* 理想视口（ideal viewport）

一个能完美适配移动设备的视口，需要手动添加meta标签，一般视口大小都设置为设备大小

**<meta name="viewport" content="width=device-width">**

* **meta标签属性**

利用以下属性对viewport进行控制, 可多个同时使用，并用逗号隔开

width 设置layout viewport 的宽度，为一个正整数，或字符串”device-width”

initial-scale 设置页面的初始缩放值，为一个数字，可以带小数

minimum-scale 允许用户的最小缩放值，为一个数字，可以带小数

maximum-scale 允许用户的最大缩放值，为一个数字，可以带小数

user-scalable 是否允许用户进行缩放(no代表不允许/yes代表允许)。