# 响应式布局

实现响应式布局的方法：em rem % flex 媒体查询 bootstrap等等。

**px**像素（Pixel）。相对长度单位。像素px是相对于显示器屏幕分辨率而言的。  
em是相对长度单位。**相对于当前对象内文本的字体尺寸**。如当前对行内文本的字体尺寸未被人为设置，则相对于浏览器的默认字体尺寸

任意浏览器的默认字体高都是16px。所有未经调整的浏览器都符合: 1em=16px。那么12px=0.75em,10px=0.625em。

为了简化font-size的换算，需要在css中的body选择器中声明Font-size=62.5%，这就使em值变为 16px\*62.5%=10px=1em, 这样12px=1.2em, 16px=1.6em, 也就是说只需要将你的原来的px数值除以10，然后换上em作为单位就行了。  
  
**em**特点  
1. em的值并不是固定的；  
2. em会继承父级元素的字体大小。  
所以我们在写em的时候，需要注意两点：  
1. body选择器中声明Font-size=62.5%;浏览器默认是16px,1em=16\*0.625px=10px;  
2. 将你的原来的px数值除以10，然后换上em作为单位；  
3. 重新计算那些被放大的字体的em数值。避免字体大小的重复声明。  
  
也就是避免1.2 \* 1.2= 1.44的现象。比如说你在#content中声明了字体大小为1.2em，那么在声明p的字体大小时就只能是1em，而不是1.2em, 因为此em非彼em，它因继续#content的字体高而变为了1em=12px。  
  
  
**rem**是CSS3新增的一个相对单位（root em，根em），这区别在于使用rem为元素设定字体大小时，**仍然是相对大小，但相对的只是HTML根元素**。这个单位可谓集相对大小和绝对大小的优点于一身，通过它既可以做到只修改根元素就成比例地调整所有字体大小，又可以避免字体大小逐层复合的连锁反应。目前，除了IE8及更早版本外，所有浏览器均已支持rem,所以用在移动端比较多。

% ：百分比通常用在浮动布局，或者超过两列的布局上比较多，用来设置宽度。

**Flex**  **弹性布局**

**兼容问题：**

新规范"display: flex"

Chrome 21+，safari 6.1+ ,firefox 22+,Opra 12.1 ，ie 11+，Android4.4+,ios7.1+

老规范"display: box"

Chrome 20-，safari 3.1+ ,firefox 2-21 ，Android2.4+,ios3.2+

中间版本"display: flexbox"

ie10

**用法**

任何的容器都可以用flex布局，dispaly:inline-flex;dispaly:flex;

添加前缀例如：-webkit，-moz,-ms

不过注意的是：设为flex布局后，子元素的float,clear,vertical-align属性会失效

Flex容器具有的属性：

项目的排列方向：

flex-direction: row | row-reverse | column | column-reverse;

项目在一条轴线排下的换行方式：

flex-wrap: nowrap | wrap | wrap-reverse;

以上两个属性（方向和换行方式）的简写

flex-flow: <flex-direction> || <flex-wrap>;默认是row nowrap

设置在主轴上（和设置的方向有关）的对齐方式

justify-content: flex-start | flex-end | center | space-between | space-around;

在垂直方向的对齐方式（主要针对方向设置为水平row）

align-items: flex-start | flex-end | center | baseline | stretch;align-items

内容过多的换行方式

align-content: flex-start | flex-end | center | space-between | space-around | stretch;

Flex容器的子元素的属性：

排列顺序。数值越小，排列越靠前，默认为0,可以设置负数

flex-grow: <number>;

Order：number

设置占用的比例，默认为0，即如果存在剩余空间，也不放大。

flex-grow: number;

设置缩小比例，默认为1，即如果空间不足，该项目将缩小。

flex-shrink: <number>;

在分配多余空间之前，项目占据的主轴空间。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。

flex-basis: <length> | auto;

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

flex快捷值：auto (1 1 auto) 和 none (0 0 auto)

flex: none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]

align-self属性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，如果没有父元素，则等同于stretch。

align-self: auto | flex-start | flex-end | center | baseline | stretch;

兼容，添加前缀：

.flex-container {

display: -webkit-box;

display: -moz-box;

display: -ms-flexbox;

display: -webkit-flex;

display: flex;

}

.flex-item {

-webkit-box-flex: auto;

-moz-box-flex: auto;

-webkit-flex: auto;

-ms-flex: auto;

flex: auto;

}

.flex-item {

-webkit-box-ordinal-group: 1;

-moz-box-ordinal-group: 1;

-ms-flex-order: 1;

-webkit-order: 1;

order: 1;

}