

一、弹性布局(flex布局)

1. Flex 布局是什么? <http://www.ruanyfeng.com/blog/2015/07/flex-grammar.html>

Flex 是 Flexible Box的缩写,意为"弹性布局",用来为盒状模型提供最大的灵活性。

任何一个容器都可以指定为 Flex 布局。

```
1 .box{
2     display: flex;
3 }
```

注意: 设为 Flex布局以后, 子元素的float、clear和vertical-align属性将失效。

2. 基本概念

采用 Flex 布局的元素,称为Flex容器,它的所有子元素自动成为容器成员,称为 Flex 项目。

3. 容器的属性

以下6个属性设置在容器上。**父元素:**

1. flex-direction 属性决定主轴的方向(即项目的排列方向)。

```
1 .box {
2     flex-direction: row | row-reverse | column | column-reverse;
3 }
4 row (默认值): 从左到右
5 row-reverse: 从右到左
6 column: 从上到下
7 column-reverse: 从下到上
```

2. flex-wrap 放不下时是否换行

```
1 .box{
2     flex-wrap: nowrap | wrap | wrap-reverse;
3 }
4 nowrap (默认): 不换行
5 wrap: 换行, 第一行在上方
6 wrap-reverse: 换行, 第一行在下方
```

3. flex-flow 是flex-direction属性和flex-wrap属性的简写形式

```
1 .box {
2     flex-flow: <flex-direction> || <flex-wrap>;
3 }
```

4. justify-content 设置项目在主轴上的对齐方式(flex-direction 设置的方向就是主轴)

```
1 .box {
2     justify-content: flex-start | flex-end | center | space-between | space-around;
3 }
4 flex-start (默认值): 左对齐
5 flex-end: 右对齐
6 center: 居中
7 space-between: 两端对齐, 项目之间的间隔都相等
8 space-around: 项目两侧的间隔相等, 项目之间的间隔比项目与边框的间隔大一倍
```

5. align-items 设置项目在交叉轴上的对齐方式(flex-direction 设置的对应方向就是交叉轴)

```
1 .box {
2     align-items: flex-start | flex-end | center | baseline | stretch;
3 }
4 flex-start: 交叉轴的起点对齐
```

```
5 flex-end: 交叉轴的终点对齐
6 center: 交叉轴的中点对齐
7 baseline: 项目的第一行文字的基线对齐
8 stretch (默认值): 如果项目未设置高度或设为auto, 将占满整个容器的高度
```

6. align-content 设置多根轴线的对齐方式。如果项目只有一根轴线, 该属性不起作用。

```
1 .box {
2   align-content: flex-start | flex-end | center | space-between | space-around | stretch;
3 }
4 flex-start: 与交叉轴的起点对齐。
5 flex-end: 与交叉轴的终点对齐。
6 center: 与交叉轴的中点对齐。
7 space-between: 与交叉轴两端对齐, 轴线之间的间隔平均分布。
8 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。
9 stretch (默认值): 轴线占满整个交叉轴。
```

4.项目的属性

以下6个属性设置在项目上。 **子元素:**

1. order 设置项目的排列顺序。数值越小, 排列越靠前, 默认为0。

```
1 .item {
2   order: 数字;
3 }
```

2. flex-grow 设置项目的放大比例, 默认为0, 即如果存在剩余空间, 也不放大。

```
1 .item {
2   flex-grow: 数字;
3 }
4 如果所有项目的flex-grow属性都为1, 则它们将等分剩余空间 (如果有的话)
5 如果一个项目的flex-grow属性为2, 其他项目都为1, 则前者占据的剩余空间将比其他项多一倍
```

3. flex-shrink 设置项目的缩小比例, 默认为1, 即如果空间不足, 该项目将缩小。

```
1 .item {
2   flex-shrink: 数字;
3 }
4 为0的时候, 不收缩
5
6 为1的时候, 会收缩
```

4. flex-basis 设置了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。

```
1 .item {
2   flex-basis: <length> | auto;
3 }
4 它可以设为跟width或height属性一样的值 (比如350px), 则项目将占据固定空间。
```

5. flex 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。

```
1 .item {
2   flex: none | auto;
3 }
4 该属性有两个快捷值: auto (1 1 auto) 和 none (0 0 auto)。
```

6. align-self 设置单个项目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。

```
1
2 .item {
3   align-self: auto | flex-start | flex-end | center | baseline | stretch;
4 }
5 该属性可能取6个值，除了auto，其他都与align-items属性完全一致。
```

二、伪类选择器

1. 考虑其他同级元素的影响

- a. E:first-child 第一个
- b. E:last-child 最后一个
- c. E:only-child 只有一个节点的时候选中
- d. E:nth-child(n) 正数第几个，从1开始，计算所有元素
- e. E:nth-last-child(n) 倒数第几个

2. 不考虑其他同级元素的影响

- a. E:first-of-type 第一个
- b. E:last-of-type 最后一个
- c. E:only-of-type 只有一个同类型的元素
- d. E:nth-of-type(n) 正数第几个，从1开始，只计算同类元素
- e. E:nth-last-of-type(n) 倒数第几个，只计算同类元素

三、字体图标

1. 自定义字体

```
1 @font-face {
2   // 自定义字体的名字
3   font-family: "iconfont";
4   // 要使用的字体文件
5   src: url('iconfont.eot?t=1479642403788'); /* IE9*/
6   src: url('iconfont.eot?t=1479642403788#iefix') format('embedded-opentype'), /* IE6-IE8 */
7   url('iconfont.woff?t=1479642403788') format('woff'), /* chrome, firefox */
8   url('iconfont.ttf?t=1479642403788') format('truetype'), /* chrome, firefox, opera, Safari, Android */
9   url('iconfont.svg?t=1479642403788#iconfont') format('svg'); /* iOS 4.1- */
10 }
11
12 div {
13   font-family: "iconfont";
14 }
```

2. 使用iconfont（阿里巴巴矢量图标库）

图标下载地址：<https://www.iconfont.cn/>

[https://www.xp.cn/css3/](https://www.xp.cn/css3/css参考手册)
css参考手册