

# 1. flex布局体验

## 1.1. 传统布局与flex布局

传统布局：

- 兼容性好
- 布局繁琐
- 局限性，不能在移动端很好的布局
- 

flex布局：

- 操作方便，布局极为简单，移动端应用很广泛
- PC端浏览器支持情况较差
- IE11或更低版本，不支持或仅部分支持

建议：

- PC端页面页面布局，还是传统布局
- 如果是移动端或者不考虑兼容性问题的PC端页面布局，还是使用flex弹性布局

## 1.2. 初体验

(1)搭建HTML结构

```
<div>
  <span></span>
  <span></span>
  <span></span>
</div>
```

# 2. flex布局原理

## 2.1. 布局原理

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

当我们为父盒子设为flex布局之后，子元素的float、clear、和vertical-align属性将失效

## 伸缩布局=弹性布局=伸缩盒布局=弹性盒布局=flex布局

采用flex布局的元素，称为flex容器(flex container)，简称“容器”。它的所有子元素自动成为容器成员，称为flex项目(flex item)简称“项目”

体验找那个div就是flex父容器

体验中span就是子容器flex项目

子容器可以横向排列也可以纵向排列

总结flex布局原理：

就是通过给父盒子添加flex属性，来控制子盒子的位置和排列方式

# 3. flex布局父项常见属性

## 3.1. 常见父项属性

以下由六个属性是对父元素设置的：

- flex-direction:设置主轴的方向
- justify-content:设置主轴上的子元素排列方式
- flex-wrap:设置子元素是否换行
- align-items:设置侧轴上的子元素排列方式（单行）
- align-content:设置侧轴上的子元素的排列方式（多行）
- flex-flow:复合属性，相当于同时设置了flex-direction和flex-wrap

## 3.2. flex-direction设置主轴的方向

（1）主轴和侧轴

在flex布局中，是分为主轴和侧轴两个方向，同样的叫法有：行和列、X轴和Y轴

（2）属性值

flex-direction属性决定主轴的方向（即项目的排列方向）

注意：

主轴和侧轴是会变化的，就看flex-direction设置谁为主轴，剩下的就是侧轴，而我们的子元素就是跟着主轴来排列的

## 3.3. justify-content设置主轴上的子元素排列方式

justify-content属性定义了项目在主轴上的对齐方式

注意：

使用这个值之前一定要确定好主轴是哪个

### 3.4. flex-wrap设置子元素是否换行

默认情况下，项目都排在一条线上（又称“轴线”）。flex-wrap属性定义，flex布局中默认是不换行的

### 3.5. align-items设置侧轴上的子元素排列方式（单行）

该属性是控制子项在侧轴上（默认是Y轴）上的排列方式在子项为单项的时候使用

### 3.6. align-content设置侧轴上的子元素的排列方式(多行)

设置子项在侧轴上的排列方式，并且只能用于子项出现换行的情况（多行），在单行下是没有效果的

#### align-items和align-content的区别

align-items适用于单行情况下，只有上对齐、下对齐、居中和拉伸

align-content适用于换行（多行）的情况下（单行情况下无效），可以设置上对齐、下对齐、居中、拉伸以及平均分配剩余空间等属性值

总结就是单行找align-items、多行找align-content

### 3.7. flex-flow

flex-flow属性是flex-direction和flex-wrap属性的复合属性

## 4. flex布局子项常见属性

flex子项目占的份数

align-self控制子项自己在侧轴的排列方式

order属性定义子项的排列顺序（前后顺序）

### 4.1. flex属性

flex属性定义子项目分配剩余空间，用flex来表示占多少份数

语法：

```
item{
  flex:<number>;/*default 0*/
}
```

## 4.2. align-self控制子项自己在侧轴的排列方式

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

语法：

```
span:nth-child(2){
  /*设置自己在侧轴上的排列方式*/
  align-self:flex-end;
}
```

## 4.3. order属性定义项目的排列顺序

数值越小，排列越靠前，默认是0

注意：和z-index不一样

### 常见的flex布局思路：

### 背景线性渐变gradient background

语法：

```
background:linear-gradient(起始方向,颜色1,颜色2,...);
background:-webkit-linear-gradient(left,red,blue);
background:-webkit-linear-gradient(left,top,red,blue);
```

注意：

- 背景渐变必须添加浏览器私有前缀
- 起始方向是：方位名词或者度数，如果省略默认是top