

选择器优先级

- 通配符 < 标签 < 类 < id < 行内 < !important

优先级 - 叠加计算规则

```
div {
  color: red;
}

.box {
  color: green;
}
```

```
.box div {
  color: red;
}

.box {
  color: green;
}
```

叠加计算：如果是**复合选择器**，则需要权重叠加计算。

公式：（每一级之间**不存在进位**）

（行内样式，id选择器个数，类选择器个数，标签选择器个数）

规则：

- 从左向右依次比较选个数，**同一级个数多的优先级高**，如果个数相同，则向后比较
- !important 权重最高
- 继承权重最低

复合选择器

- 后代选择器
 - 父选择器 子选择器 { CSS属性 }
 - 父子选择器之间用空格隔开
 - 会选中所有后代标签
- 子代选择器
 - 父选择器 > 子选择器 { CSS属性 }
- 并集选择器
 - 选择器1, 选择器2, 选择器3...
- 交集选择器
 - 选择器1选择器2 {CSS属性}
 - 选择器连写，不用隔开
 - 选中同时满足多个条件的元素
- 伪类选择器
 - 选择器:hover {CSS属性}
- 伪类选择器（超链接状态）
 - :link 访问前
 - :visited 访问后
 - :hover 鼠标悬停
 - :active 点击时
- 结构伪类选择器

- `E:first-child` 查找第一个E元素
- `E:last-child` 查找最后一个E元素
- `E:nth-child(N)` 查找第N个E元素（第一个元素N值为1）
- `E:nth-child(公式)`

:nth-child(公式)

作用：根据元素的**结构关系**查找**多个**元素。

功能	公式
偶数标签	$2n$
奇数标签	$2n+1; 2n-1$
找到5的倍数的标签	$5n$
找到第5个以后的标签	$n+5$
找到第5个以前的标签	$-n+5$

- 伪元素选择器
 - `E::before` 在E元素里面**最前面**添加一个伪元素
 - `E::after` 在E元素里面**最后面**添加一个伪元素

伪元素选择器

作用：创建**虚拟**元素（伪元素），用来摆放**装饰性**的内容。

选择器	说明
<code>E::before</code>	在 E 元素 里面最前面 添加一个伪元素
<code>E::after</code>	在 E 元素 里面最后面 添加一个伪元素

注意点：

- 必须设置 `content: " "` 属性，用来 设置**伪元素的内容**，如果没有内容，则引号**留空**即可
- 伪元素默认是**行内**显示模式
- **权重**和**标签选择器**相同



背景属性

- 背景色 `background-color`
- 背景图 `background-image`

- 背景图平铺方式`background-repeat`
 - `no-repeat` 不平铺
 - `repeat` 平铺
 - `repeat-x` 水平方向平铺
 - `repeat-y` 垂直方向平铺
- 背景图位置`background-position`
 - `left` 左侧
 - `right` 右侧
 - `center` 居中
 - `top` 顶部
 - `bottom` 底部
- 背景图缩放`background-size`
 - `cover` 完全缩放
 - `contain` 根据背景区域缩放
 - 百分比、数字+单位
- 背景图固定`background-attachment`
 - `fixed` 固定定位
- 背景复合属性`background`
 - 背景色 背景图 背景图平铺方式 背景图位置/背景图缩放

标签显示模式

- 块级元素
 - 独占一行
 - 宽度默认是父级的100%
 - 添加宽高元素生效
- 行内元素
 - 一行多个
 - 尺寸由内容撑开
 - 添加宽高元素不生效
- 行内块元素
 - 一行多个
 - 尺寸由内容撑开
 - 添加宽高元素生效

转换显示模式`display`

- `block` 块级
- `inline-block` 行内块
- `inline` 行内

盒子模型

盒子模型 - 组成

盒子模型重要组成部分：

- ⁺ 内容区域 - width & height
 - 内边距 - **padding**（出现在内容与盒子边缘之间）
 - 边框线 - **border**
 - 外边距 - **margin**（出现在盒子外面）
- 盒子模型被border和padding撑大解决方法
 1. 手动计算，减去border/padding的尺寸
 2. 内减模式:`box-sizing: border-box`
 - margin不会撑大盒子
 - 左右居中:`margin: 上下 auto`
 - 清除标签默认样式

```
*{
  margin: 0;
  padding: 0;
  /* 盒子内减模式，避免被border和padding撑大 */
  box-sizing: border-box;
  /* 去掉列表的项目符号 */
  list-style: none;
  /* 去掉a标签的下划线 */
  text-decoration: none;
}
```

- 元素溢出`overflow`
 - hidden 溢出隐藏
 - scroll 溢出滚动（无论是否溢出，都显示滚动条）
 - auto 溢出滚动（溢出才显示滚动条）
- 塌陷问题
 - 父子级标签，子级添加上外边距，会产生塌陷；导致父子级一起向下移动
 - 解决方法
 1. 取消父级margin，父级设置padding
 2. 父级设置`overflow: hidden`
 3. 父级设置`border-top`

- 圆角边框**border-radius**
 - 圆形: **border-radius:50%**/高和宽的一半
 - 胶囊: **border-radius:高的一般**
- 盒子阴影**box-shadow**
 - X Y 模糊半径 扩散半径 颜色 内外阴影(内inset/默认外)

浮动**float**

- **left** 左对齐
- **right** 右对齐

清除浮动

1. 额外标签法: 在父元素内容的最后添加一个块级元素**div#clearfix**, 设置CSS属性**clear: both**
2. 单伪元素法:

```
.clearfix::after {  
  /* 伪元素必须要有content属性 */  
  content:"";  
  display:block;  
  clear:both;  
}
```

3. 双伪元素法

```
/* before 解决外边距塌陷 */  
.clearfix::before,  
.clearfix::after {  
  /* 伪元素必须要有content属性 */  
  content:"";  
  display:table;  
}  
/* after 清除浮动 */  
.clearfix::after {  
  clear:both;  
}
```

4. 父级元素添加CSS属性 **overflow:hidden**

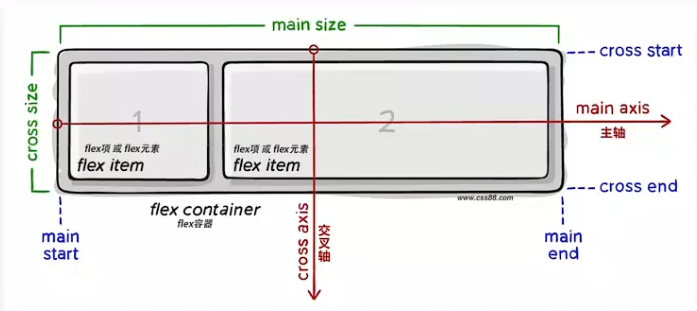
Flex布局**display:flex;**

Flex - 组成

设置方式：给父元素设置 `display: flex`，子元素可以自动挤压或拉伸

组成部分：

- 弹性容器
- 弹性盒子
- 主轴：默认在水平方向
- 侧轴 / 交叉轴：默认在垂直方向



Flex 布局

描述	属性
创建 flex 容器	<code>display: flex</code>
主轴对齐方式	<code>justify-content</code>
侧轴对齐方式	<code>align-items</code>
某个弹性盒子 侧轴对齐方式	<code>align-self</code>
修改主轴方向	<code>flex-direction</code>
弹性伸缩比	<code>flex</code>
弹性盒子换行	<code>flex-wrap</code>
行对齐方式	<code>align-content</code>

- 创建Flex容器 `display: flex`

- 主轴对齐方式 `justify-content`

主轴对齐方式

属性名：`justify-content`

属性值	效果
<code>flex-start</code>	默认值，弹性盒子从 起点 开始依次排列
<code>flex-end</code>	弹性盒子从 终点 开始依次排列
<code>center</code>	弹性盒子沿主轴 居中 排列
<code>space-between</code>	弹性盒子沿主轴均匀排列，空白间距均分在弹性盒子 之间
<code>space-around</code>	弹性盒子沿主轴均匀排列，空白间距均分在弹性盒子 两侧
<code>space-evenly</code>	弹性盒子沿主轴均匀排列，弹性盒子与容器之间间距相等

- 侧轴对齐方式 `align-items`
- 某个弹性盒子侧轴对齐方式 `align-self`

侧轴对齐方式

属性名

- `align-items`：当前弹性容器内**所有**弹性盒子的侧轴对齐方式（给弹性容器设置）
- `align-self`：单独控制**某个**弹性盒子的侧轴对齐方式（给弹性盒子设置）

+

属性值	效果
<code>stretch</code>	弹性盒子沿着侧轴线被 拉伸至铺满容器 （弹性盒子没有设置侧轴方向尺寸则默认拉伸）
<code>center</code>	弹性盒子沿侧轴 居中 排列
<code>flex-start</code>	弹性盒子从 起点 开始依次排列
<code>flex-end</code>	弹性盒子从 终点 开始依次排列

- 修改主轴方向 `flex-direction`

修改主轴方向

主轴默认在水平方向，侧轴默认在垂直方向

属性名：`flex-direction`

属性值

属性值	效果
row	水平方向，从左向右（默认）
column	垂直方向，从上向下
row-reverse	水平方向，从右向左
column-reverse	垂直方向，从下向上

- 弹性伸缩比 `flex`

弹性伸缩比

作用：控制弹性盒子的主轴方向的尺寸。

属性名：`flex`

属性值：**整数数字**，表示占用父级**剩余**尺寸的**份数**。

- 弹性盒子换行 `flex-wrap`

弹性盒子换行

弹性盒子可以自动挤压或拉伸，默认情况下，所有弹性盒子都在一行显示。

属性名：`flex-wrap`

属性值

- `wrap`: 换行
- `nowrap`: 不换行（默认）

- 行对齐方式 `align-content`

行对齐方式

属性名：`align-content`

属性值

属性值	效果
<code>flex-start</code>	默认值，弹性盒子从起点开始依次排列
<code>flex-end</code>	弹性盒子从终点开始依次排列
<code>center</code>	弹性盒子沿主轴居中排列
<code>space-between</code>	弹性盒子沿主轴均匀排列，空白间距均分在弹性盒子之间
<code>space-around</code>	弹性盒子沿主轴均匀排列，空白间距均分在弹性盒子两侧
<code>space-evenly</code>	弹性盒子沿主轴均匀排列，弹性盒子与容器之间间距相等

定位 `position`

- 子级绝对定位，父级相对定位（子绝父相）
- `relative` 相对定位
 - 参照物：自己原来的位置
 - 不脱标，占位
 - 标签的显示模式特点不变化
- `absolute` 绝对定位
 - 参照物：先找最近的已经定位的祖先元素；如果没有，则以浏览器可视区域改变位置
 - 脱标，不占位

- 3. 显示模式改变：宽高生效（具备行内块的特点）
- `fixed` 固定定位
 - 1. 参照物：浏览器窗口
 - 2. 脱标，不占位
 - 3. 显示模式改变：宽高生效（具备行内块的特点）
- `z-index` 堆叠层级
 - 取值是整数，数字越大，越靠上
- 定位居中
 - 1. 绝对定位 `position:absolute;`
 - 2. 水平、垂直边偏移量为50% `left:50%;top:50%;`
 - 3. 子级向左、上移动自己尺寸的一半 `transform:translate(-50%,-50%);`

定位-总结

定位模式	属性值	是否脱标	显示模式	参照物
相对定位	<code>relative</code>	否	保持标签原有显示模式	自己原来位置
绝对定位	<code>absolute</code>	是	行内块特点	1. 已经定位的祖先元素 2. 浏览器可视区
固定定位	<code>fixed</code>	是	行内块特点	浏览器窗口