

# CSS

## CSS

css引入

选择器

权重

设置字体属性

标签归类

行级元素inline

块级元素block

行级元素inline-block

文本类

元素定位position

absolute

relative

absolute和relative比较

fixed固定定位

sticky粘性定位

定位总结

定位的特殊性

## css引入

- 行间样式引入 `div style=""`
- 界面级css `[style][style]`
- 引入外部css文件 `link rel="stylesheet" href="地址"`
- 在css中引入另一个css文件 `@import url(地址)`

## 选择器

注：写样式时因该遵守顺序

1. 布局定位属性：`display / position / float / clear / visibility / overflow`（建议 `display` 第一个写，毕竟关系到模式）
2. 自身属性：`width / height / margin / padding / border / background`
3. 文本属性：`color / font / text-decoration / text-align / vertical-align / white-space / break-word`
4. 其他属性（CSS3）：`content / cursor / border-radius / box-shadow / text-shadow / background:linear-gradient...`

- id选择器 [#]
- class选择器 [.]
- 标签选择器 [div]
- 通配符选择器 [\*]
- 复杂选择器、父子选择器 [最外面元素 外面元素 里面元素]
- 直接子元素选择 [div > em{ }]
- 并列选择器 [div.demo]
- 分组选择器 [div,div,div]{ }
- 伪类选择器 [a:link]

## 权重

是显示样式的比较值谁大就用谁的样式

CSS 权重	
标签名	权重值
!important <i>max important &gt; #id</i>	infinity 正无穷
行间样式	1000
id 选择器 #	100
class 选择器、属性选择器、伪类选择器	10
标签选择器、伪元素选择器 <i>div</i>	1
通配符选择器 * .	0
在计算机中，正无穷+1>正无穷	
如果权重值一样（优先级一样），会显示后面的	
就是先来后到，谁在后面，谁（后面的）覆盖前面的——后面的会覆盖前面的	
在权重中，是 256 进制，是从 0 到 255 后变成 1	
所以这里的 1000 不是一千，100 不是一百	

## 设置字体属性

## 1) 设置字体

属性名	属性值 (举例)	效果/功能
font-size:	12px	设置字体大小, 默认是 16px, 但是互联网字体一般用 12px 或 14px。这个设置的是字体的高度
font-weight:	lighter	细体
	normal	正常 (默认值)
	bold	加粗, 跟<strong>效果是一样的
	bolder	加更粗
	100,200 到 900	从细到粗, 都是整百来表达
font-style:	italic;	斜体, em 这个标签本身就带有 italic 的样式
font-family:	arial;	设置字体包的样式, 这个是乔布斯发明的字体
color:	见下面	注意字体颜色直接写 color, 不写 font-color:red;
字体颜色的 color 三种写法 : 1、纯英文单词 : red, green 等 (一般不用) 2、颜色代码, 如 #ff4400; 其中每两位都是从 00 到 ff, 分别对应红 r (从 00 到 ff), 绿 g (从 00 到 ff), 蓝 b (从 00 到 ff), 这是一个十六进制的数, 代表的是饱和度, 如果没两位都可以重复, 那么简写成 #f40 淘宝红, #ffffff 白色, #000000 黑色。 3、颜色函数, 如 color:rgb(255,255,255); //这是白色 rgb ( 0-255, 0-255, 0-255 ) 里面的 0-255 是十进制数 transparent 是透明色		
border:	1px solid black;	这是一个复合属性。可以给容器加外边框。 第一位是 border-width 代表粗细; 第二位是 border-style 设置实心 solid, 虚线; 第三位是 border-color 设置颜色 还可以单独写成 border-width:1px; border-style:dashed;代表虚线 dotted 是点

**思考：**如何写出一个三角形？

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

```
div{
```

```
width: 0px;
```

```
height: 0px;
```

```
border: 100px solid black;
```

```
border-left-color:red; //此处可以改为 transparent 透明色
```

```
border-top-color:green; //此处可以改为 transparent 透明色
```

```
border-right-color:blue; //此处可以改为 transparent 透明色
```

```
}
```



<b>text-align:</b>	left;	对齐方式：左对齐
	center;	对齐方式：一行居中
	right;	对齐方式：右对齐
<b>line-height:</b>	16px;	单行文本所在的高度。 当 line-height=height （文本所占高度=容器高度）时，单行文本水平垂直居中
	1.2em ;	单行文本所在的高度。意思是文字的行高是 1.2 倍行高
<b>text-indent:</b>	2em;	意思是首行缩进 2em（2 个文本单位）
单位的衡量标准：绝对单位（m，cm 等），相对单位（px，em 等） px 是像素的意思，一个像素只能显示一个颜色； 屏幕的分辨率，就是说的像素，国际标准是每英寸所能容纳的垂直像素点数。 em 是文本单位，1em = 1 * font-size 该文本的字体大小		
<b>text-decoration:</b>	line-through;	中划线
	none;	没有线
	underline;	下划线
	overline;	上划线（基本没用）
<b>cursor:</b>	pointer;	光标定位值（cursor: pointer;鼠标变成一个小手）

## 标签归类

可以通过display属性更改元素性质

## 行级元素inline

特点：

- 内容决定元素的宽高
  - 不可以通过 **CSS** 改变宽高
- 列：span、strong、a、em....

## 块级元素block

特点：

- 独占一行可以被元素撑开
  - 可以通过 **CSS** 改变宽高
- 列：div、p、ul、h1.....

## 行级元素inline-block

特点：

- 内容决定元素的宽高
  - 可以通过 **CSS** 改变宽高
- 列：span、strong、a、em

## 文本类

2 ) inline 和 inline-block 都是文本类元素

凡是带有 inline 属性的元素，都有文本类特点，所以叫文本类元素

## 元素定位position

一般都是子绝父相：子元素绝对定位父元素使用相对定位，父元素用relative相对定位作为参照物

### absolute

**absolute：是绝对定位，脱离原来的位置对、是层模型**

根据离得最近有 **absolute/relative** 的父级来定位如果父级以上没有定位那么根据文档来定位

### relative

**relative:相对定位，保留原来的位置、层模型**

根据出生的位置来定位：灵魂出窍

## absolute和relative比较

### 相同点

- 都会跑到上一层显示会覆盖他底下的元素
- 都需要配合top、left、right、bottom来定位

### 不同点

- absolute：1、脱离了文档跑到了上一层2、根据父级来定位
- relative：1、保留原来的位置2、根据原来的位置定位

## fixed固定定位

广告专用，固定在一个地方，不占有原来的位置

### 中心显示

```
div {  
  position: fixed;  
  top: 50%;  
  left: 50%;  
  /* translate(-50%, -50%)作用是，往上（X轴），左（Y轴）移动自身长度的50%，以使其居于中心位置。 */  
  transform: translate(-50%, -50%);  
  width: 100px;  
  height: 100px;  
  border: 1px solid black;  
}
```

### 固定到中间右侧

**固定定位小技巧：固定在版心右侧位置。**

小算法：

1. 让固定定位的盒子 left: 50%. 走到浏览器可视区（也可以看做版心）的一半位置。
2. 让固定定位的盒子 margin-left: 版心宽度的一半距离。多走 版心宽度的一半位置就可以让固定定位的盒子贴着版心右侧对齐了。

## sticky粘性定位

**粘性定位**可以被认为是相对定位和固定定位的混合。Sticky 粘性的语法：

```
选择器 { position: sticky; top: 10px; }
```

粘性定位的特点：

1. 以浏览器的可视窗口为参照点移动元素（固定定位特点）
2. 粘性定位**占有原先的位置**（相对定位特点）
3. 必须添加 top、left、right、bottom 其中一个才有效

```
/* 当div离盒子顶端只有0px时会固定到顶端 */  
position: sticky;  
top: 0;  
margin-top: 300px;
```

## 定位总结

### 1.9 定位的总结

定位模式	是否脱标	移动位置	是否常用
static 静态定位	否	不能使用边偏移	很少
relative 相对定位	否 (占有位置)	相对于自身位置移动	常用
absolute 绝对定位	是 (不占有位置)	带有定位的父级	常用
fixed 固定定位	是 (不占有位置)	浏览器可视区	常用
sticky 粘性定位	否 (占有位置)	浏览器可视区	当前阶段少

## 定位的特殊性

```
<!-- 浮动不会压着文字 (inline)属性的  
absolute会压着-->  
<!-- <div class="border wh100 fl"></div> -->  
<div class="border wh100 poabs colorgreen"></div>  
<p>  
    Sint eum vel enim adipisci eligendi et aperiam  
    Ut quisquam illo reprehenderit cumque culpa cum  
    mollitia. Animi laborum tempore. Ipsa dolore do  
    autem expedita perferendis illum cupiditate vol  
    ut non. Eos reiciendis sunt laboriosam illum qu  
    eius saepe magnam consequatur.  
</p>
```

