

CSS基础

什么是CSS

层叠样式表 — 也就是CSS — 是在HTML之后应该学习的第二门技术。HTML 用于定义内容的结构和语义，**CSS 用于设计风格和布局**。比如，可以使用 CSS 来更改内容的字体、颜色、大小、间距，将内容分为多列，或者添加动画及其他的装饰效果。

CSS可以通过以下几种方式添加到HTML中：

1.行内样式-在HTML元素中使用“style”属性

```
<div style="color:red;">一些文字</div>
```

缺点：HTML页面不纯净，文件体积大，不利于蜘蛛爬行，后期维护不方便。

注：蜘蛛爬行是指蜘蛛在互联网上收集数据到数据库的过程。整个互联网就像是一张蜘蛛网，蛛丝就是链接，搜索引擎要抓取一个网页就像蜘蛛在蛛网上爬行一样，从一个一个链接上爬过然后发现新的网页。

2.内嵌样式-在HTML文档头部< head >区域使用< style >元素来包含CSS

```
<style>
  p{
    color:red;
  }
  #d1{
    color:green;
  }
  .d2{
    font-size:large;
  }
</style>
```

优缺点：页面使用公共CSS代码，也是每个页面都要定义的，如果一个网站有很多页面，每个文件都会变大，后期维护难度也大，如果文件很少，CSS代码也不多，这种样式还是很不错的。

3.链接样式-使用外部CSS文件

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

优缺点：实现了页面框架代码与表现CSS代码的完全分离，使得前期制作和后期维护都十分方便。

*优先级

行内样式 > 内嵌样式 > 文件导入（链接样式）

可用“就近原则”来概括：

- 1.行内样式就在标签内，离标签最近，所以优先级最高；
- 2.内嵌样式在同一文件内，但是离标签比较远，优先级居中；
- 3.文件导入式（即链接样式）不在同一文件，因此优先级最低。

1.基本语法

两部分组成：1.选择器，2.一或多条声明；

选择器通常是需要改变样式的 HTML 元素。

每条**声明**由一个属性和一个值组成。

属性（property）是希望设置的样式属性（style attribute）。例如：字体大小、颜色、背景等；

每个属性有一个值。属性和值被**冒号**分开。

```
<style>
  p{
    color:red;
  }
  #d1{
    color:green;
  }
  .d2{
    font-size:large;
  }
</style>
```

2.选择器

id选择器

HTML元素以id属性来设置id选择器,CSS 中 id 选择器以 "#" 来定义。

```
<style>
  #d1{
    font-size: medium;
  }
</style>
```

```
<body>
  <div id="d1">
    这是一个盒子，姑且叫它盒子一
  </div>
</body>
```

class选择器

class 选择器在 HTML 中以 class 属性表示, 在 CSS 中, 类选择器以一个点 "." 号显示。

```
<style>
  .d2{
    color:orange;
    text-align: center;
    height:100px;
    width:100px;
    background-color: cadetblue;
  }
</style>
```

```
<body>
  <div class="d2">
    这是一个盒子，就叫盒子二吧
  </div>
</body>
```

3.背景

背景颜色

background-color属性定义了元素的背景颜色。

三种常见定义方式：

十六进制- 如: "#ff0000"

RGB - 如: "rgb(255,0,0)"

注: 可以用rgba(255,0,0,0.5), 最后一个参数是透明度, 为0完全透明, 为1完全不透明。

颜色名称 - 如: "red"

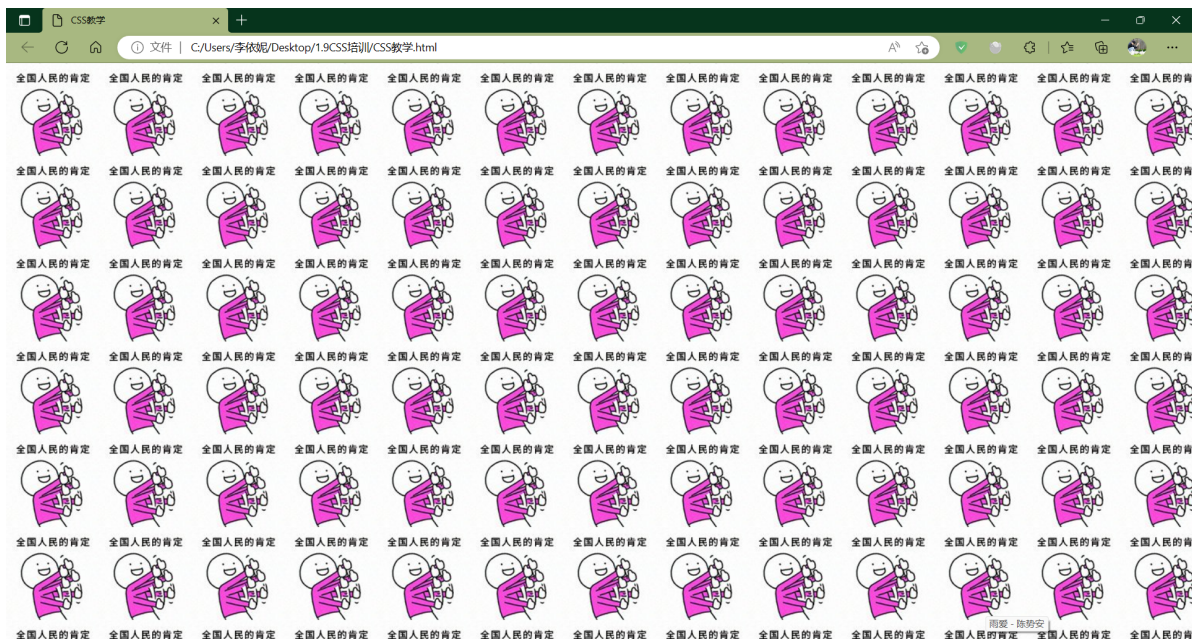
背景图像

background-image:url('图片链接');

图片链接或者本地图片地址

默认情况下, 背景图像进行平铺重复显示, 以覆盖整个元素实体。如图所示:

```
<style>
  body{
    background-image:url('./表情包.jpg');
  }
</style>
```



水平/垂直平铺

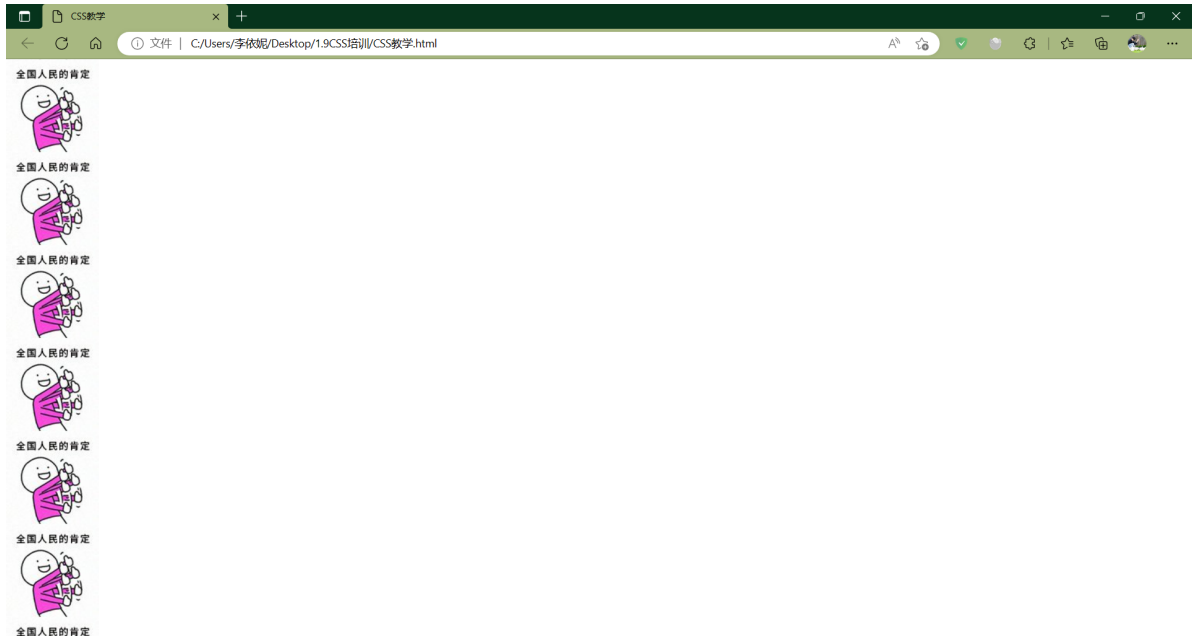
水平平铺

```
<style>
  body{
    background-image: url('./表情包.jpg');
    background-repeat: repeat-x;
    /* 水平平铺 */
  }
</style>
```



垂直平铺

```
<style>
  body{
    background-image:url('./表情包.jpg');
    background-repeat: repeat-y;
    /* 垂直平铺 */
  }
</style>
```



不平铺

```
<style>
  body{
    background-image:url('./图片.jpg');
    background-repeat:no-repeat;
    /* 不平铺 */
  }
</style>
```



设置定位

background-position:值一, 值二;

常见定义方式:

值一: 1.left/right/center; 2.x%;3.50px;

值二: 2.top/bottom/center;2.y%;3.100px;

注: 如果用第一种定义方式, 仅指定一个关键字, 其他值将会是"center"; 如果用第二、三种定义方式, 仅指定了一个值, 其他值将是50%;

```
<style>
  body{
    background-image:url('');
    background-repeat:no-repeat;
    background-position:right top;
  }
</style>
```

4.文本

文本颜色

color是文本颜色属性。

```
<style>
  .d1{
    color:green;
  }
  #d2{
    color:#00ff00;
  }
  p{
    color:rgb(255,0,0);
  }
</style>
```

文本对齐

text-align是文本水平对齐属性。

文本可居中对齐"center"、左对齐"left"、右对齐"right"、两端对齐"justify"。

```
<style>
  .d1{
    text-align:center;
  }
</style>
```

文本修饰

text-decoration是设置或删除文本的装饰的属性。

上划线“**overline**”、下划线“**underline**”、删除线“**line-through**”。

```
<style>
  .d1{
    text-decoration:underline;
  }
</style>
```

5.链接

可以设置不同状态的链接颜色“color”、背景颜色“background-color”和文本样式“text-decoration”不同。

注：CSS定义超链接是要有先后顺序的。否则，在有些浏览器里面有可能会出现某个样式不起作用的bug。不能正确显示想要的效果。

CSS属性的排列顺序：link-visited-hover-active

a:link - 正常，未访问过的链接

a:visited - 用户已访问过的链接

a:hover - 当用户鼠标放在链接上时

a:active - 链接被点击的那一刻

```
<style>
  a:visited {
    color:#00FF00;
    /* 已访问链接 */
  }
</style>
```

6.列表

在HTML中，有两种类型的列表：

无序列表 ul - 列表项标记用特殊图形（如小黑点、小方框等）

有序列表 ol - 列表项的标记有数字或字母

```
<style>
  ul.a {
    list-style-type:circle;
  }
  ul.b {
    list-style-type:square;
  }
  ol.c {
    list-style-type:upper-roman;
  }
  ol.d {
    list-style-type:lower-alpha;
  }
</style>
```

- 写代码
- 改代码
- 发现新bug
- 继续改bug

- 写代码
- 改代码
- 发现新bug
- 继续改bug

- I. 写代码
- II. 改代码
- III. 发现新bug
- IV. 继续改bug

- 1. 写代码
- 2. 改代码
- 3. 发现新bug
- 4. 继续改bug

- a. 写代码
- b. 改代码
- c. 发现新bug
- d. 继续改bug

7.表格

表格边框

```
<style>
  table,th,td{
    border:1px solid black;
  }
</style>
```

折叠边框

```
<style>
  table{
    border-collapse: collapse;
  }
  table,th,td {
    border: 1px solid black;
  }
</style>
```


表格宽度和高度

```
<style>
  table {
    width:100%;
  }
  th{
    height:50px;
  }
</style>
```

表格文字对齐

```
<style>
  td{
    text-align:right;
  }
</style>
```

表格填充

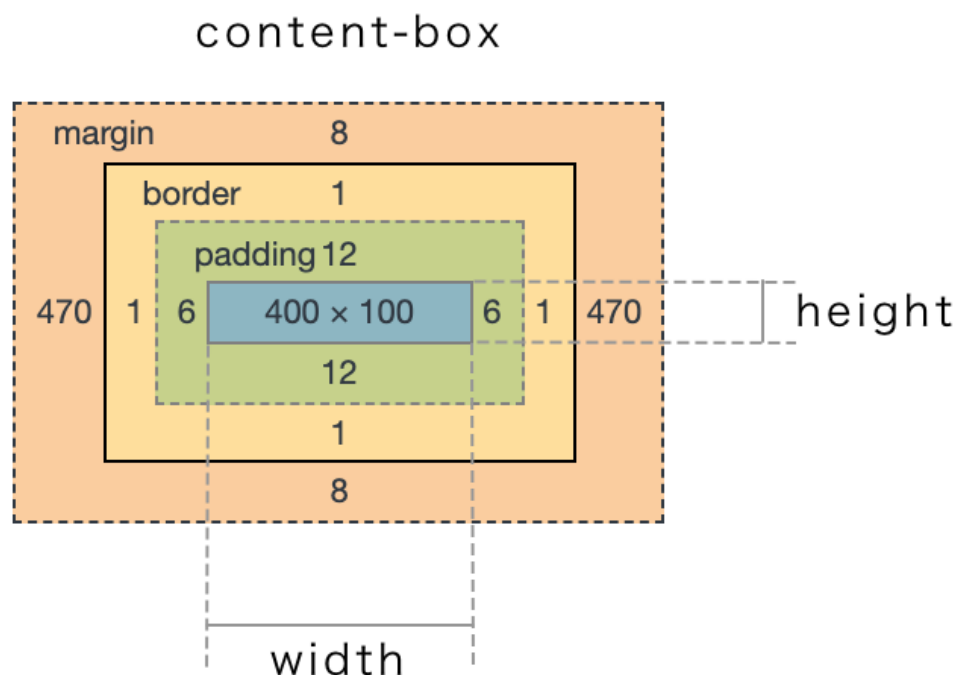
控制边框和表格内容之间的间距，使用td和th元素的填充属性。

```
<style>
  td{
    padding:15px;
  }
</style>
```

表格颜色

```
<style>
  table, td, th{
    border:1px solid green;
  }
  th{
    background-color:green;
    color:white;
  }
</style>
```

8.盒子模型



Margin(外边距) - 清除边框外的区域，外边距是透明的。

Border(边框) - 围绕在内边距和内容外的边框。

Padding(内边距) - 清除内容周围的区域，内边距是透明的。

Content(内容) - 盒子的内容，显示文本和图像。

当指定一个 CSS 元素的宽度和高度属性时，只是设置了内容区域的宽度和高度。如果要完整大小的元素，还必须添加内边距，边框和外边距。

比如下面这个例子总宽度为 450px:

```
<style>
.d3{
  width: 300px;
  border: 25px solid green;
  padding: 25px;
  margin: 25px;
}
</style>
```



300px (宽) + 50px (左 + 右填充) + 50px (左 + 右边框) + 50px (左 + 右边距) = 450px

如果只有 250 像素的空间。设置总宽度为 250 像素的元素:

```
<style>
.d3{
    width: 220px;
    border: 5px solid green;
    padding: 10px;
    margin: 0;
}
</style>
```

220px (宽)+ 20px (左 + 右填充)+ 10px (左 + 右边框)+0px (左 + 右边距)= 250px

9.边框

指定元素边框样式和颜色。

边框样式

none: 默认无边框

dotted: 定义一个点线边框

dashed: 定义一个虚线边框

solid: 定义实线边框

double: 定义两个边框。两个边框的宽度和 border-width 的值相同

groove: 定义3D沟槽边框。效果取决于边框的颜色值

ridge: 定义3D脊边框。效果取决于边框的颜色值

inset: 定义一个3D的嵌入边框。效果取决于边框的颜色值

outset: 定义一个3D突出边框。效果取决于边框的颜色值

边框宽度

通过 border-width 属性为边框指定宽度。

```
<style>
.d5{
    border-style:solid;
    border-width:5px;
}
</style>
```

边框颜色

border-color属性用于设置边框的颜色。可以设置的颜色：

name - 指定颜色的名称, 如 "red"

RGB - 指定 RGB 值, 如 "rgb(255,0,0)"

Hex - 指定16进制值, 如 "#ff0000"

还可以设置边框的颜色为"transparent"透明。

单独设置各边

可以单独指定不同的侧面不同的边框。

```
<style>
  p{
    border-top-style:dotted;
    border-right-style:solid;
    border-bottom-style:dotted;
    border-left-style:solid;
  }
</style>
```

简写

也可以利用border-style属性有1-4个值这个特点简写。

例：border-style:dotted solid double dashed;

上边框是 dotted，右边框是 solid，底边框是 double，左边框是 dashed

例：border-style:dotted solid double;

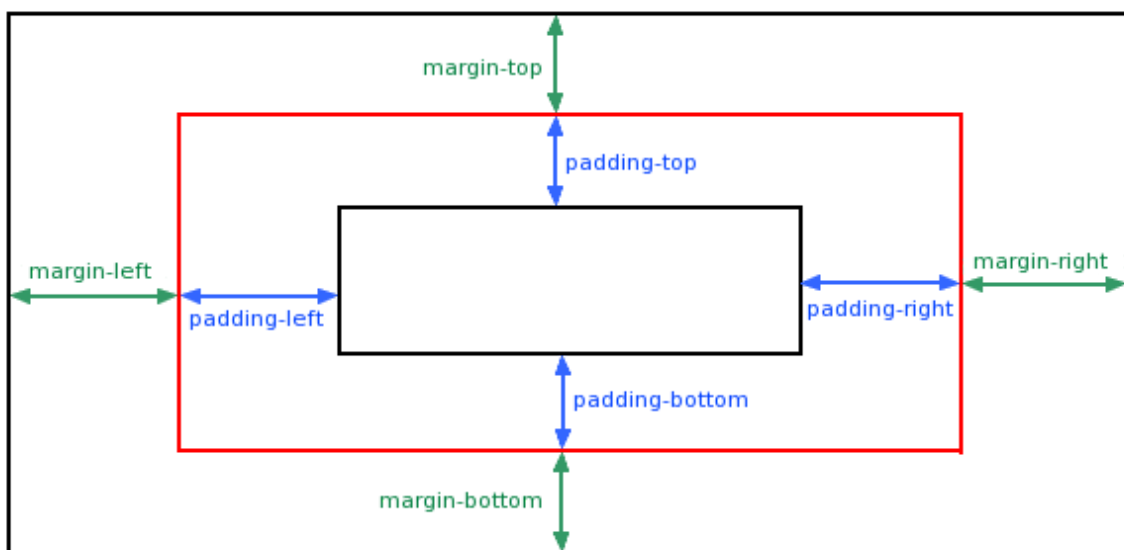
上边框是 dotted，左、右边框是 solid，底边框是 double

例：border-style:dotted solid;

上、底边框是 dotted，右、左边框是 solid

10.外边距margin

margin 可以单独改变元素的上，下，左，右边距，也可以一次改变所有的属性。



取值可以是：

auto：设置浏览器边距，依赖于浏览器

length：定义一个固定的margin（使用像素，pt，em等）

%：定义一个使用百分比的边距

注：margin可以使用负值，重叠的内容。

单边外边距属性

可以单独设置某一边的外边距。如下：

margin-top:100px;

margin-bottom:100px;

margin-right:50px;

margin-left:50px;

简写

margin属性可以有一到四个值。

例: `margin:25px 50px 75px 100px;`

上边距为25px, 右边距为50px, 下边距为75px, 左边距为100px

例: `margin:25px 50px 75px;`

上边距为25px, 左右边距为50px, 下边距为75px

例: `margin:25px 50px;`

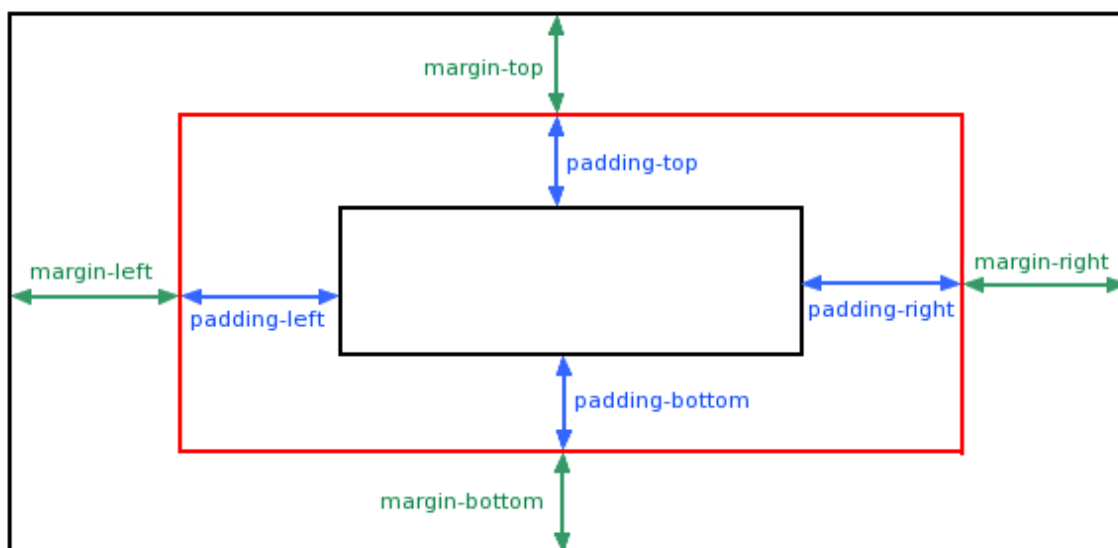
上下边距为25px, 左右边距为50px

例: `margin:25px;`

所有的4个边距都是25px

11.填充padding

padding (填充) 是一个简写属性, 定义元素边框与元素内容之间的空间, 即上下左右的内边距。



取值可以是:

length: 定义一个固定的填充(像素, pt, em,等)

%: 使用百分比值定义一个填充

简写

Padding属性可以有一到四个值。

例: `padding:25px 50px 75px 100px;`

上填充为25px, 右填充为50px, 下填充为75px, 左填充为100px

例: `padding:25px 50px 75px;`

上填充为25px, 左右填充为50px, 下填充为75px

例: `padding:25px 50px;`

上下填充为25px, 左右填充为50px

例: `padding:25px;`

所有的填充都是25px

12.分组和嵌套

分组选择器

由于在样式表中有很多具有相同样式的元素。比如：

```
<style>
  h1 {
    color:green;
  }
  h2 {
    color:green;
  }
  p {
    color:green;
  }
</style>
```

为了尽量减少代码，可以使用分组选择器。每个选择器用逗号分隔。如下：

```
<style>
  h1,h2,p{
    color:green;
  }
</style>
```

嵌套选择器

p{ }: 为所有 p 元素指定一个样式。

.marked{ }: 为所有 class="marked" 的元素指定一个样式。

.marked p{ }: 为所有 class="marked" 元素内的 p 元素指定一个样式。

p.marked{ }: 为所有 class="marked" 的 p 元素指定一个样式。

13.显示

display属性设置一个元素应如何显示，visibility属性指定一个元素应可见还是隐藏。

隐藏

隐藏一个元素可以通过把display属性设置为"none"，或把visibility属性设置为"hidden"。

display:none;

visibility:hidden;

区别

两种方法会产生不同的结果。

visibility:hidden可以隐藏某个元素，**但隐藏的元素仍需占用与未隐藏之前一样的空间**。也就是说，该元素虽然被隐藏了，**但仍然会影响布局**。

14.定位position

position 属性指定了元素的定位类型。

属性有5个值：

static、relative、fixed、absolute、sticky

static

HTML 元素的默认值，即没有定位，遵循正常的文档流对象。静态定位的元素不会受到 top, bottom, left, right影响。

fixed

元素的位置相对于浏览器窗口是固定位置。**即使窗口是滚动的它也不会移动。**

relative

相对定位元素的定位是相对其正常位置。移动相对定位元素，但它原本所占的空间不会改变。相对定位元素经常被用来作为绝对定位元素的容器块。**即选一个元素为参照物，其他元素以他为标准定位**

absolute

absolute 定位使元素的位置与文档流无关，因此不占据空间。absolute 定位的元素和其他元素会重叠。

sticky

sticky 英文字面意思是粘，粘贴，所以可以把它称之为粘性定位。

position: sticky;** 基于用户的滚动位置来定位。 **

粘性定位的元素是依赖于用户的滚动，在 position:relative 与 position:fixed 定位之间切换。

它的行为就像 position:relative; 而当页面滚动超出目标区域时，它的表现就像 position:fixed;, 它会固定在目标位置。

元素定位表现为在跨越特定阈值前为相对定位，之后为固定定位。

重叠的元素

元素的定位与文档流无关，所以它们可以覆盖页面上的其它元素

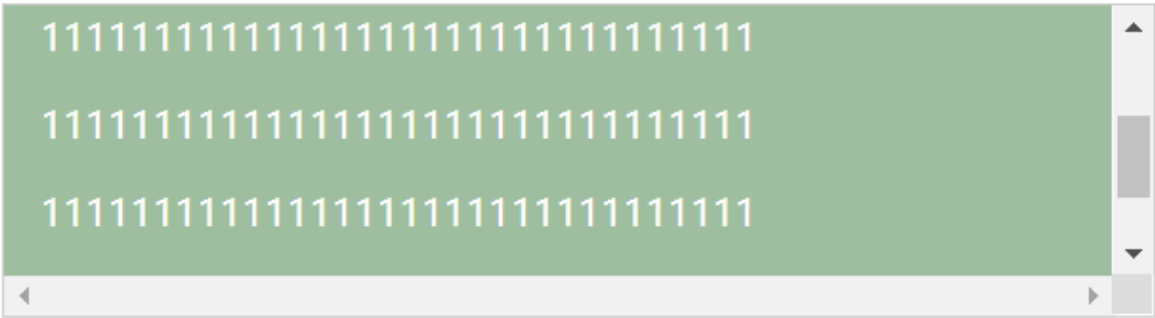
z-index属性指定了一个元素的堆叠顺序（哪个元素应该放在前面，或后面）

一个元素可以有正数或负数的堆叠顺序：

```
<style>
.d6{
    position:absolute;
    left:0px;
    top:0px;
    z-index:-1;/*这里元素重叠时，z-index值小的在下方*/
}
</style>
```

15.Overflow

overflow 属性用于控制内容溢出元素框时显示的方式。



取值可以是：

- visible**：默认值。内容不会被修剪，会呈现在元素框之外。
- hidden**：内容会被修剪，并且其余内容是不可见的。
- scroll**：内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。
- auto**：如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。
- inherit**：规定应该从父元素继承 overflow 属性的值。

16.Float浮动

Float（浮动），会使元素向左或向右移动，其周围的元素也会重新排列，往往是用于图像，但它在布局时一样非常有用。



量子力学（Quantum Mechanics），为物理学理论，是研究物质世界微观粒子运动规律的物理学分支，主要研究原子、分子、凝聚态物质，以及原子核和基本粒子的结构、性质的基础理论。它与相对论一起构成现代物理学的理论基础。量子力学不仅是现代物理学的基础理论之一，而且在化学等学科和许多近代技术中得到广泛应用。19世纪末，人们发现旧有的经典理论无法解释微观系统，于是经由物理学家的努力，在20世纪初创立量子力学，解释了这些现象。量子力学从根本上改变人类对物质结构及其相互作用的理

元素浮动

float:left;左浮动



量子力学（Quantum Mechanics），为物理学理论，是研究物质世界微观粒子运动规律的物理学分支，主要研究原子、分子、凝聚态物质，以及原子核和基本粒子的结构、性质的基础理论。它与相对论一起构成现代物理学的理论基础。量子力学不仅是现代物理学的基础理论之一，而且在化学等学科和许多近代技术中得到广泛应用。19世纪末，人们发现旧有的经典理论无法解释微观系统，于是经由物理学家的努力，在20世纪初创立量子力学，解释了这些现象。量子力学从根本上改变人类对物质结构及其相互作用的理

float:right;右浮动

量子力学（Quantum Mechanics），为物理学理论，是研究物质世界微观粒子运动规律的物理学分支，主要研究原子、分子、凝聚态物质，以及原子核和基本粒子的结构、性质的基础理论。它与相对论一起构成现代物理学的理论基础。量子力学不仅是现代物理学的基础理论之一，而且在化学等学科和许多近代技术中得到广泛应用。19世纪末，人们发现旧有的经典理论无法解释微观系统，于是经由物理学家的努力，在20世纪初创立量子力学，解释了这些现象。量子力学从根本上改变人类对物质结构及其相互作用的理



17.网页布局

网页布局对完善网站的外观非常重要，具有为网站所有者和用户构造网站上存在的信息的作用。将网站的最重要的元素置于网站的正面和中心，定义了内容层次结构，引导访问者浏览网站，尽可能地传达信息。

在写布局前注意：

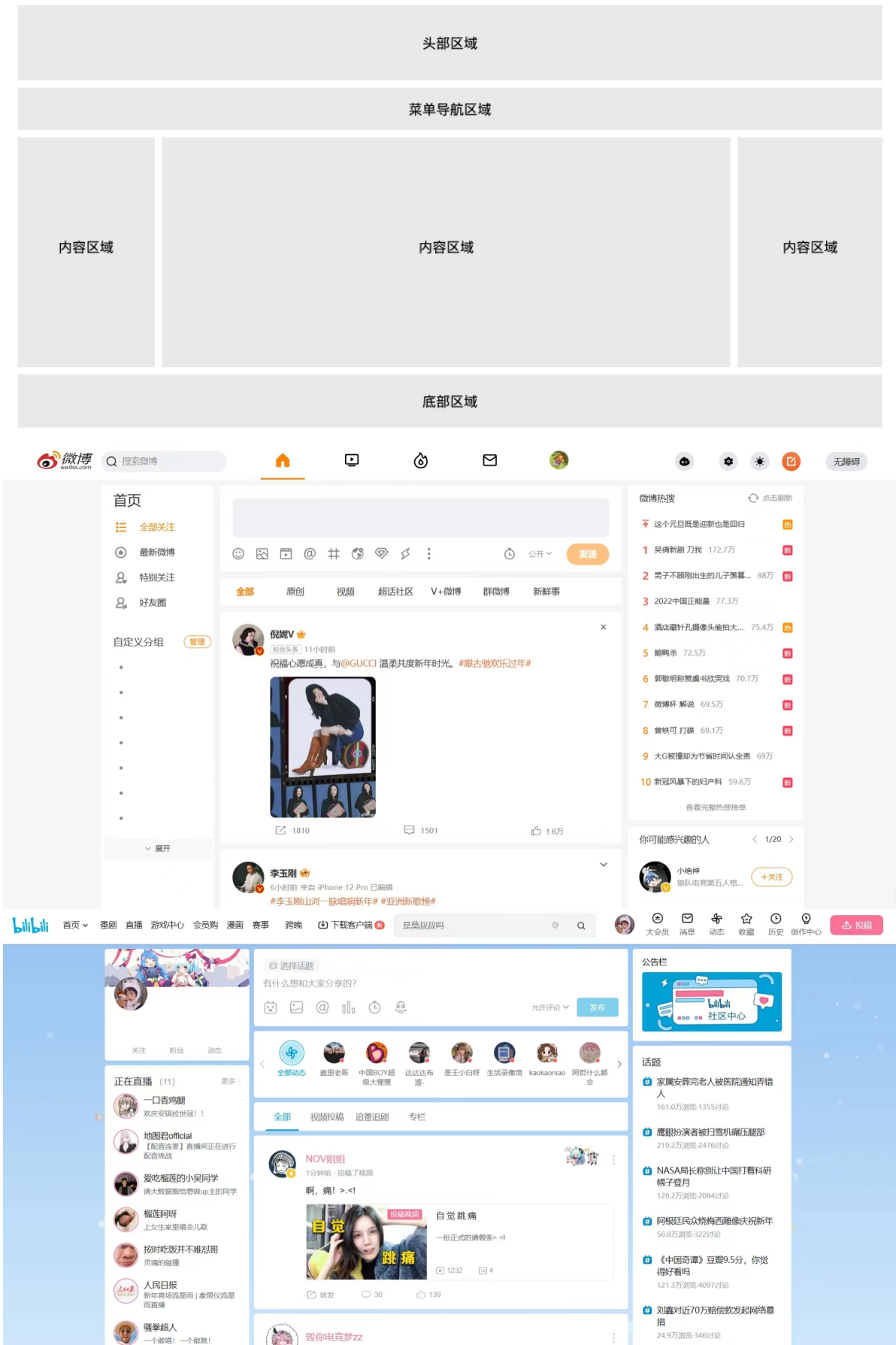
用全局样式通配符把各种边距归零，用于网页所有样式的边距都重置：

```
<style>
*{
    margin:0;
    padding:0;
}
</style>
```


布局类型

布局结构

一般分为以下几个部分：头部区域、菜单导航区域、内容区域、底部区域。



响应式布局

响应式布局是同一页面在不同的屏幕上有不同的布局，即只需要一套代码使页面适应不同的屏幕。

width=device-width: 自适应手机屏幕的尺寸宽度。

maximum-scale: 缩放比例的最大值。

minimum-scale: 缩放比例的最小值。

initial-scale: 缩放的初始化。

user-scalable: 用户的可以缩放的操作。

第一步设置meta标签

第二步使用媒体查询设置样式

flex布局（弹性布局）

`display:flex;`

设置后，子元素默认排成一行。

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”。容器默认存在两根轴，分别为水平的主轴（main axis）和垂直的交叉轴（cross axis）。主轴的开始位置叫做 main start，结束位置叫做 main end；交叉轴的开始位置叫做 cross start，结束位置叫做 cross end。项目默认沿主轴排列。单个项目占据的主轴空间叫做 main size，占据的交叉轴空间叫做 cross size。

注：可以通过将元素的 display 属性设置为 flex（生成块级 flex 容器）或 inline-flex（生成类似 inline-block 的行内块级 flex 容器）。当一个元素设置了 Flex 布局以后，其子元素的 float、clear 和 vertical-align 等属性将失效。

display 指定 HTML 元素的盒子类型

flex-direction 指定弹性盒子中子元素的排列方式

flex-wrap 设置当弹性盒子的子元素超出父容器时是否换行

flex-flow flex-direction 和 flex-wrap 两个属性的简写

justify-content 设置弹性盒子中元素在主轴（横轴）方向上的对齐方式

align-items 设置弹性盒子中元素在侧轴（纵轴）方向上的对齐方式

align-content 修改 flex-wrap 属性的行为，类似 align-items，但不是设置子元素对齐，而是设置行对齐

order 设置弹性盒子中子元素的排列顺序

align-self 在弹性盒子的子元素上使用，用来覆盖容器的 align-items 属性

flex 设置弹性盒子中子元素如何分配空间

flex-grow 设置弹性盒子的扩展比率

flex-shrink 设置弹性盒子的收缩比率

flex-basis 设置弹性盒子伸缩基准值

圣杯布局（经典web布局）

具有以下特点：

- 1.header和footer各自占领屏幕所有宽度，高度固定。
- 2.中间的container是一个三栏布局。
- 3.三栏布局两侧宽度固定不变，中间部分自动填充整个区域。

18.瀑布流

使用column-count和column-gap来实现瀑布流布局（适合简单的瀑布流排版）。