

CSS，主讲：汤小洋

一、CSS简介

1. 什么是CSS

CSS: Cascading Style Sheet 层叠样式表

是一组样式设置规则，用于控制页面的外观样式

2. 为什么使用CSS

- 实现内容与样式的分离，便于团队开发
- 样式复用，便于网站的后期维护
- 页面的精确控制，让页面更精美

3. CSS作用

- 页面外观美化
- 布局和定位

二、基本用法

1. CSS语法

```
<head>
  <style>
    选择器 {
      属性名:属性值;
      属性名:属性值;
    }
  </style>
</head>
```

- 选择器：要修饰的对象（东西）
- 属性名：修饰对象的哪个属性（样式）
- 属性值：样式的取值

2. CSS应用方式

也称为CSS引用方式，有三种方式：内部样式、行内样式、外部样式

2.1 内部样式

也称为内嵌样式，在页面头部通过head内style标签定义

对当前页面中所有符合样式选择器的标签都起作用

2.2 行内样式

也称为嵌入样式，使用HTML标签的style属性定义

只对设置style属性的标签起作用

2.3 外部样式

使用单独的.css文件定义，然后在页面中使用 link标签 或 @import指令 引入

- ★ 使用link标签链接外部样式表文件 (推荐)

```
<link rel="stylesheet" type="text/css" href="CSS样式文件的路径">
```

type属性可以省略

- @import指令导入外部样式表文件

```
<style>
  @import "CSS样式文件的路径";
  @import url(CSS样式文件的路径);
</style>
```

三、选择器

1. 基础选择器

1.1 标签选择器

也称为元素选择器，使用HTML标签作为选择器名称

以标签名作为样式应用依据

1.2 类选择器

使用自定义名称，以 `.` 号作为前缀，然后通过HTML标签的 `class` 属性调用类选择器

以标签的 `class` 属性作为样式应用依据

注意事项:

- 调用时不能添加 `.` 号
- 同时调用多个类选择器以 空格 隔开
- 类选择名称不能以数字开头

1.3 ID 选择器

使用自定义名称，以 `#` 号作为前缀，然后通过HTML标签的 `id` 属性进行名称匹配

以标签的 `id` 属性作为样式应用依据，一对一的关系

2. 复杂选择器

2.1 复合选择器

标签选择器和类选择器、标签选择器和ID选择器，一起使用

必须同时满足两个条件才能应用样式

```
/* 复合选择器 */  
h1.aaa{  
    color:red;  
}  
p#bbb{  
    color:blue;  
}
```

2.2 组合选择器

```
h1,p,div,.ccc{  
    font-size:30px;  
}
```

也称为集体声明

将多个具有相同样式的选择器放在一起声明，使用逗号隔开

2.3 嵌套选择器

```
div p{  
    color:green;  
    text-decoration:underline;  
}
```

在某个选择器内再设置选择器，通过空格隔开

只有满足层次关系最里层的选择器所对应的标签才会应用样式

注：使用 空格 时不区分父子还是后代，使用CSS3中新增的 `>` 时必须是父子关系才行

2.4 伪类选择器

根据不同的状态显示不同的样式，一般多用于 `<a>` 标签

四种状态:

- !link 未访问的链接
- 4visited 已访问的链接
- 2hover 鼠标移动到链接上，即悬浮在链接上
- 3active 选定的链接，被激活

注：默认超链接为：蓝色、下划线

2.5 伪元素选择器

:first-letter 为第一个字符添加样式

:first-line 为第一行添加样式

:before 在元素内容的最前面添加内容，需要配合content属性使用

:after 在元素内容的最后面添加内容，需要配合content属性使用

```
p:first-letter{
  color:red;
  font-size:30px;
}
p:first-line{
  background:pink;
}
p:before{
  content:"嘿嘿";
}
p:after{
  content:"哈哈";
}
```

3. 选择器优先级

3.1 优先级

行内样式>ID选择器>类选择器>标签选择器

原因：首先加载标签选择器，再加载类选择器，然后加载ID选择器，最后加载行内样式

后加载的会覆盖先加载的同名样式 加载顺序

3.2 内外部样式加载顺序

就近原则

原因：按照书写顺序依次加载，在同优先级的前提下，后加载的会覆盖先加载的同名样式，所以离的越近越优先

3.3 !important

可以使用!important使某个样式有最高的优先级

四、常用CSS属性

1. 字体属性

定义字体相关的样式

属性	含义	说明
font-size	大小、尺寸	可以使用多种单位
font-weight	粗细	
font-family	字体	
font-style	样式	
font	简写	

1.1 font-size

取值：

- inherit 继承，默认从父标签继承字体大小（默认值），所有CSS属性的默认值都inherit
- px 像素 pixel
- % 百分比，相对于父标签字体大小的百分比
- em 倍数，相对于父标签字体大小的倍数

HTML根元素默认字体大小为16px，也称为基础字体大小

1.2 font-weight

取值：

- normal 普通（默认）
- bold 粗体
- 自定义 400 normal、700 bold

1.3 font-family

要求系统中要安装指定的字体

一般建议写3种字体：首选、其次、备用，以及逗号隔开

1.4 font-style

取值：

- normal 普通
- italic 斜体

1.5 font

简写属性：font::font-style | font-weight | font-size | font-family

必须按此顺序书写

2. 文本属性

属性	含义	说明
color	颜色	
line-height	行高	行之间的高度
text-align	水平对齐方式	取值: left、center、right
vertical-align	垂直对齐方式	取值: <u>top、middle、bottom</u> 可用于图片和文字的对齐方式
text-indent	<u>首行缩进</u>	让文本向右缩进60px <div>div{ text-indent:35px; }</div>
text-decoration	文本修饰	取值: underline、overline、line-through、none 删除线
text-transform	字母大小写	取值: lowercase、uppercase、capitalize 首字母大写
letter-spacing	<u>字符间距</u>	px
word-spacing	<u>单词间距</u>	只对英文有效 不换行
white-space	空白的处理方式	文本超出后是否换行, 取值: nowrap

```
white-space:nowrap;  
overflow:hidden; /* 超出后隐藏不显示 */
```

也可以自动加省略号

2.1 color

取值, 四种写法:

- 颜色名称, 使用英文单词

✱ 16进制的RGB值: #RRGGBB `background:#9032e6;`

特定情况下可以缩写:

```
#FFFFFF ---> #FFF  
#000000 ---> #000  
#FF0000 ---> #F00  
#00FF00 ---> #0F0  
#0000FF ---> #00F  
#CCCCCC ---> #CCC  
#FF7300 ---> 无法缩写
```

注: 不区分大小写

- rgb函数: rgb(red,green,blue)

每个颜色的取值范围: [0,255]

```
rgb(255,0,0) ---> 红  
rgb(0,255,0) ---> 绿  
rgb(0,0,255) ---> 蓝
```

- rgba函数: rgba(red,green,blue,alpha)

可以设置透明度, alpha取值范围: [0,1], 0表示完全透明, 1表示完全不透明

```
rgba(255,0,0,1) ---> 纯红  
rgba(255,0,0,0.5) ---> 红色半透明
```

3. 背景属性

属性	含义	说明
background-color	背景颜色	
background-image	背景图片	
background-repeat	背景图片的重复方式	
background-position : right bottom	背景图片的显示位置	
background-attachment	背景图片是否跟随滚动	
background	简写	

3.1 background-color

取值: transparent 透明

3.2 background-image

- 必须使用url()方式指定图片的路径
- 如果是在.css样式文件中使用相对路径，此时是相对于css文件，不是相对于html文件

3.3 background-repeat

取值: repeat (默认)、repeat-x、repeat-y、no-repeat

3.4 background-position

默认背景图片显示在左上角

取值:

- 关键字: top、bottom、left、right、center
- 坐标: 左上角为(0,0)坐标，向右为x正方向，向下为y正方向

CSS雪碧图，即CSS Sprites，也称为CSS精灵，是一种CSS图像合并技术

含义: 将网页中许多非常小的图标/图片整合到一张大图中，当访问该页面时只需要下载一次，可以减少访问服务器的次数，提高性能

原理: 使用background-position进行背景定位，使用坐标精确的定位出背景图片的背景

3.5 background-attachment

取值: scroll (默认)、fixed固定不动

如始终显示在右上角

3.6 background

简写属性: background:background-color | background-image | background-repeat | background-position

以空格隔开, 书写顺序没有要求

4. 列表属性

属性	含义	说明
list-style-type	设置列表前的标记	
list-style-image	将图像作为列表前的标记	
list-style-position	设置标记的位置	取值: outside (默认)、inside
list-style	简写	

1. hello1



hello2

o hello3

hello4

4.1 list-style-type

取值: none、disc、circle、square、decimal

此时不再区分有序列表还是无序列表, 只要设置列表前的标记就可以了

4.2 list-style

简写属性: list-type:list-style-type | list-style-image | list-style-position

书写顺序没有要求

```
/*导航菜单*/
.nav li{
  list-style:none;
  float:left; /* 左浮动 */
  width:100px;
}
```

5. 表格属性

border-collapse 表格中相邻的边框是否合并 (折叠) 为单一边框

取值: separate (默认)、collapse

```
<!-- table>(tr>td{td$}*5)*4 -->
```

```
table{
  width:500px;
  /* border:1px solid red; */
  border-collapse:collapse;
}
td{
  border:1px solid red;
}
```

6. 使用LiveReload

可以实现当保存页面文件时实时刷新浏览器

步骤:

1. 在Chrome中安装LiveReload扩展程序

将 liveload_2_1_0.crx 文件拖放到Chrome的扩展程序中

勾选"允许访问文件地址"

2. 在Sublime中安装LiveReload插件

将 LiveReload.zip 解压后放到Sublime的插件目录Packages/中

3. 配置Sublime的LiveReload插件

Preferences——>Package Settings——>LiveReload——>Settings - Default

```
{
  "enabled_plugins": [
    "SimpleReloadPlugin",
    "SimpleRefresh"
  ]
}
```

4. 在浏览器中启用LiveReload

先打开要访问的页面，然后点击浏览器地址栏右边的黑色圆圈，当中心的小圆圈变成实心时表示已启用

5. 在Sublime中启用LiveReload

按Ctrl+Shift+P——>搜索livereload: 选择enable——>搜索simple reload: 选择enable

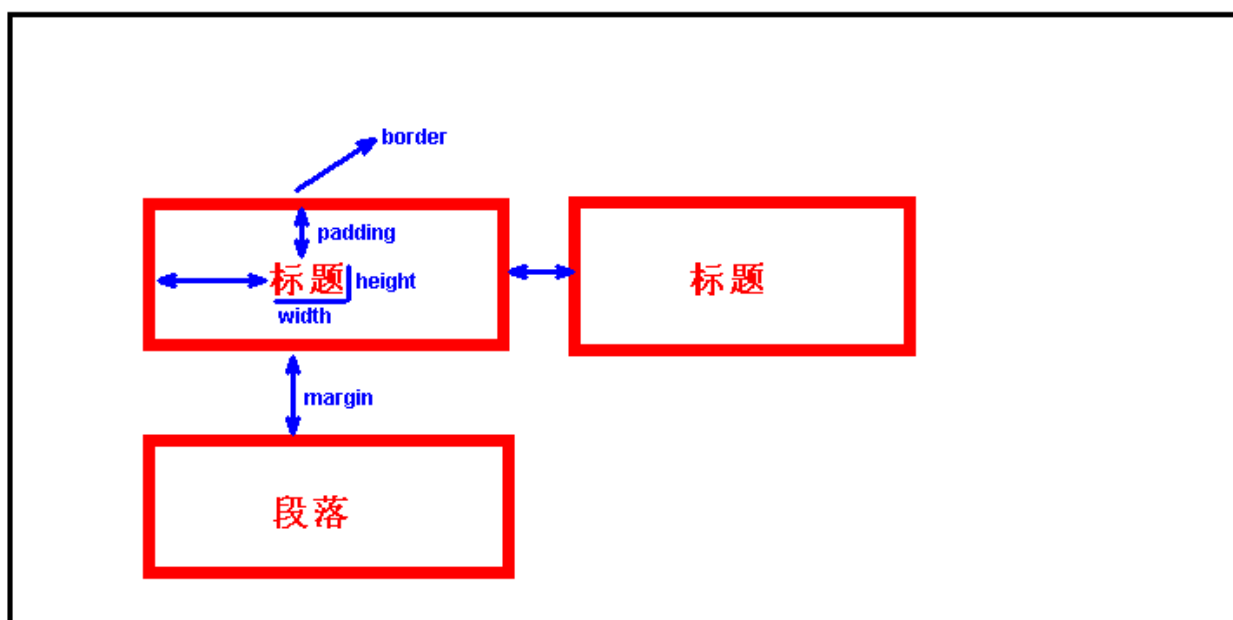
五、盒子模型

1. 简介

盒子模型是网页布局的基础，将页面中所有的元素都看作成一个盒子，盒子都包含以下几个属性：

- width 宽度
- height 高度
- border 边框
- padding 内边距
- margin 外边距

盒子模型



2. 盒子属性

2.1 border

表示盒子的边框

分为四个方向：

上top、右right、下bottom、左left

border-top、border-right、border-bottom、border-left

每个边都包含三种样式：

颜色color、粗细width、样式style

border-top-color、border-top-width、border-top-style

border-right-color、border-right-width、border-right-style

border-bottom-color、border-bottom-width、border-bottom-style

border-left-color、border-left-width、border-left-style

样式style的取值：

solid实线、dashed虚线、dotted点线、double双线、inset内嵌3D线、outset外嵌3D线

简写，三种方式：

- 按方向简写

border-top、border-right、border-bottom、border-left

书写顺序：

border-方向：width style color

- 按样式简写

border-color、border-width、border-style

书写顺序：

border-样式：top right bottom left

必须按顺序针方向书写，同时可以缩写

border-width: 2px; ——>四个边的宽度都为2px

border-width: 2px 4px;

border-width: 1px 2px 4px;

规则：如果省略，则认为上下一样，左右一样

- 终极简写

如果四个边样式完成相同，border: width style color;

2.2 padding

表示盒子的内边距，即内容与边框之间的距离

同样也分为四个方向，也可以简写（按顺序针方向，默认上下一样，左右一样）

注：如果上下冲突，则以上为准，如果左右冲突，则以左为准

2.3 margin

表示盒子的外边距，即盒子与盒子之间的距离

同样也分为四个方向，也可以简写（按顺序针方向，默认上下一样，左右一样）

居中对齐：

```
/*
元素的居中
*/
/* 1. 块级元素水平居中 */
margin:0 auto;
/* 2. 文本水平居中 */
text-align:center;
/* 3. 垂直居中：将height和line-height设置为相同 */
height:100px;
line-height:100px;
```

3. 其他

3.1 元素所占空间

页面中的元素实际所占空间

- 宽=width+左右padding+左右border+ 左右margin
- 高=height+上下padding+上下border+上下margin

3.2 盒子属性默认值

不同标签的盒子属性默认值可能不同，需要自己设置

```
body,ul,ol,dl,li,p,h1,h2,h3,h4,h5,h6,form{
  margin:0;
  padding:0
}
```

3.3 外边距合并

也称为外边距折叠，指的是两个块级元素垂直外边距相遇时，它们将合并为一个外边距，合并后的外边距值为其中的较大的那个外边距值

两种情况：

- 当一个元素出现在另一个元素上面时，第一个元素的下外边距与第二元素的上外边距会发生合并
- 当一个元素包含在另一个元素中时，并且没有内边距或边框把外边距分隔开时，两个元素的上外边距会发生合并

外边距合并的好处：让排版在视觉上显得更美观

六、定位方式

1. 简介

通过position属性实现对元素的定位，有四种定位方式

常用取值：

取值	含义	说明
static	默认值	按照常规文档流进行显示
relative	相对定位	相对于标签原来的位置进行的定位
absolute	绝对定位	相对于第一个非static定位的父标签的定位
fixed	固定定位	相对于浏览器窗口进行定位

设置定位方式后，还要设置定位属性（偏移量）：top、bottom、left、right

2. 相对定位

先设置元素的position属性为relative，然后再设置偏移量

3. 绝对定位

先设置父标签为非static定位，然后设置元素的position属性为absolute，最后再设置偏移量

注意事项：

- 一般来说都会把父标签设置为非static定位
- 如果父标签不是非static定位，则会相对于浏览器窗口进行定位
- 设置元素为绝对定位后，元素会浮在页面上方

4. 固定定位

先设置元素的position属性为fixed，然后再设置偏移量

设置元素为固定定位后，元素会浮在页面上方

5. z-index

设置元素定位方式后，元素在浮在页面上方，此时可以通过z-index属性设置优先级，控制元素的堆叠顺序

取值为数字，值越大优先级越高，默认为auto（大多数浏览器默认为0）

注：只能给非static定位的元素设置z-index属性

七、其他CSS属性

1. 浮动和清除

1.1 浮动属性

通过float属性实现元素有浮动，可以让块级元素脱离常规文档流，向左或向右移动，在同一行显示，如果一行显示不下，则会换行

常用取值：

- left 左浮动
- right 右浮动
- none 不浮动，默认值

设置float属性后，元素会浮在页面上层，此时父容器无法计算自己的尺寸，通常会在容器末尾添加一个清除了float的空div来解决

1.2 清除属性

通过clear属性实现清除，设置元素的哪一侧不允许有浮动元素，目的是为了和其他浮动元素换行隔开，只对块级元素有效

常用取值：

- left 左侧不允许有浮动元素
- right 右侧不允许有浮动元素
- both 两侧不允许有浮动元素
- none 允许两侧有浮动元素，默认值

结论：

- 对于非浮动元素，两边都可以设置清除（常用）
- 对于浮动元素，向哪边浮动，就只能设置哪边清除

```
.clr{
  clear:both;
}
</style>
</head>
<body>
  <div id="container">
    <div class="div1">div1</div>
    <div class="div2">div2</div>
    <div class="div3">div3</div>
    <div class="div4">div4</div>
    <div class="clr"></div>
  </div>
</body>
```

2. 元素的显示和隐藏

2.1 display

通过display属性设置元素是否显示，以及是否独占一行

常用取值：

取值	含义	说明
none	不显示	
inline	显示为内联元素，行级元素的默认值	将块级元素变为行级元素，不再独占一行
block	显示为块级元素，行级元素的默认值	将行级元素变为块级元素，独占一行
inline-block	显示为内联元素，但可以设置宽和高	在inline基础上允许设置宽和高

注：行级元素默认无法设置宽和高，可以为行级元素设置display:inline-block，然后就可以设置宽和高了

2.2 visibility

也可以通过visibility属性设置元素的显示和隐藏

常用属性：

取值	含义	说明
visible	显示	
hidden	隐藏	

2.3 区别

- display隐藏时不再占据页面中空间，后面的元素会占用其位置
- visibility隐藏时会占据页面中空间，位置还保留在页面中，只是不显示

3. 轮廓属性

3.1 简介

轮廓 outline，用于在元素周围绘制一个轮廓，位于border的外围，可以突出显示元素

3.2 基本用法

常用属性：

- outline-width 轮廓的宽度
- outline-color 轮廓的颜色

- `outline-style` 轮廓的样式
- `outline` 简写

在浏览器中，当鼠标点击或使用Tab键让一个表单或链接获取焦点时，该元素会有一个轮廓 `outline`

优点：可以提高使用表单的用户体验

缺点：有时会影响美观

3.3 `outline`与`border`的区别

- `border`可应用于所有的html元素，而`outline`主要用于表单元素、超链接等
- 当元素获取焦点时会自动出现`outline`轮廓效果，当失去焦点时会自动消失，这是浏览器的默认行为
- `outline`不影响元素的尺寸和位置，而`border`会影响

4. 其他属性

4.1 宽高相关

- `max-width` 设置元素的最大宽度
- `min-width` 设置元素的最小宽度
- `max-height` 设置元素的最大高度
- `min-height` 设置元素的最小高度

4.2 `overflow`属性

当元素内容溢出时该如何处理

常用取值：

- `visible` 溢出时可见，显示在元素外，默认值
- `hidden` 溢出的部分不可见（常用）
- `scroll` 无论是否出现溢出始终出现滚动条
- `auto` 溢出时自动会出现滚动条

4.3 `cursor`属性

用来设置光标的形状

常用属性：

- `default` 默认光标，一般为箭头
- `pointer` 手形，光标移动超链接上时一般显示为手形
- `move` 表示可移动
- `text` 表示文本
- `wait` 表示程序正忙，需要等待

- `help` 表示帮助

八、页面布局

1. 简介

常见页面布局：

- 表格布局
- `div`布局

2. 表格布局

2.1 简介

不适用于复杂布局，仅适用于简单、有规则的结构

定位相对准确，与浏览器基本无关，适用简单分割

2.2 用法

`table`常用样式属性：

- `border` 在表格外围设置边框
- `border-spacing` 设置单元格之间的距离（相当于`table`标签的`cellspacing`属性，即间距）
- `border-collapse` 表格中相邻边框是否合并，取值：`seprate`、`collapse`

`th`/`td`常用样式属性：

- `border` 为单元格设置边框
- `padding` 设置单元格的内边距（相当于`table`标签的`cellpadding`属性，即边距）

3. `div`布局

定位绝对精确，使用灵活，适合于复杂的布局方式

3.1 简单布局

两种形式：

- 1-1-1布局
- 1-2/3-1布局

3.2 圣杯布局

页面结构：两边的边栏宽度固定，中间主体在一定的范围内可自适应，并且主体优先加载

一般防止页面缩放太小影响浏览，都会为页面设置一个最小宽度min-width

3.3 双飞翼布局

源自淘宝的UED（用户体验设计）团队

双飞翼布局和圣杯布局要实现的效果是 相同的，只是思路不同

圣杯布局和双飞翼布局的区别：

- 双飞翼布局比圣杯布局多创建了一个div
- 双飞翼布局不用设置边框的相对定位，也不用设置偏移量
- 双飞翼布局使用的margin，圣杯布局使用的padding

实际开发中建议使用**CSS3**中的**flex**弹性盒子布局，更简单