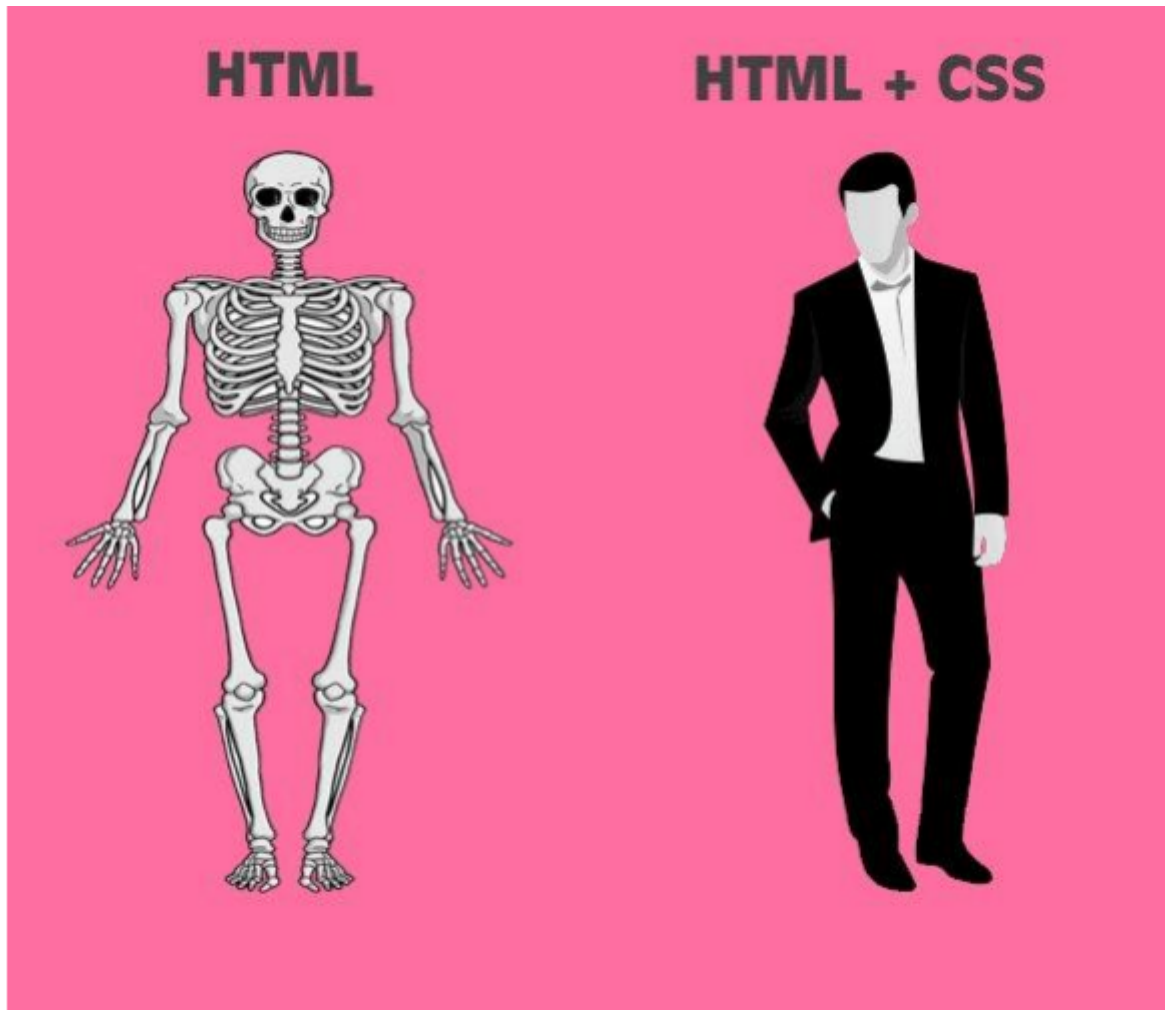


CSS

一、什么是css

是一组样式设置的规则，用于控制页面的外观样式

CSS 全称 Cascading Style Sheets 翻译过来就是层叠样式表 如果说HTML是网页的结构 那么CSS就是网页化妆师。



二、为什么使用CSS

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

三、CSS作用

1. 页面外观美化
2. 布局和定位

四、css写在哪里

CSS引用方式，有三种方式：

内部样式（直接写在head标签内）、行内样式（写在style 标签内）、外部样式（使用外部.css 文件）

1. 内部样式

在 `title` 标签下面建一个 `style` 标签 写 `css` 代码

```
<head>
  <!-- 网页采用utf-8的编码格式 -->
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- 网页标题 -->
  <title>Document</title>
  <style>
    /* 直接写css代码 */
  </style>
</head>
```

2. 行内样式

- 也称为嵌入样式，写在标签中，使用HTML标签的style属性定义
- 只对设置style属性的标签起作用

```
<div style="在这里写css代码"></div>
```

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 引入

```
<head>
  <!-- 网页采用utf-8的编码格式 -->
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- 网页标题 -->
  <title>Document</title>
  <!-- 引入css文件 -->
  <link rel="stylesheet" href="./index.css">

</head>
```

五、选择器

CSS 的选择器是 CSS 最基础也是最重要的一个知识点 很重要，用于准确的选中元素，并赋予样式。

选择器权重：

谁的权力大，就听谁的，同理，选择器权重也是一样，谁的权重值高，应用谁的

Id选择器 > class 选择器 > 元素选择器

1. 元素选择器

也称为标签选择器，使用HTML标签作为选择器的名称

```
<!-- 以内部样式为例 -->
<head>
  <style>
    div {
      /* 调整样式代码 */
    }
  </style>
</head>
<body>
  <div>我是div标签</div>
  <div>我是div标签</div>
</body>
```

2. 类选择器

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

```
<!-- 以内部样式为例 -->
<style>
  .name{

  }
  .name2{

  }
</style>

<body>
  <div class="name">我是div标签</div>
  <div class="name">我是div标签</div>
  <div class="name2">我是div标签</div>
</body>
```

3. id选择器

使用自定义名称，以 `#` 作为前缀，然后通过HTML标签的id属性进行名称匹配，id唯一。

```
<!-- 以内部样式为例 -->
<style>
  #name{

  }
</style>

<body>
  <div id="name">我是div标签</div>
</body>
```

六、css属性

1. 字体属性

属性	作用	说明
font-family	指定文本的字体系列	font-family: Arial, 宋体;
font-size	指定文本的字体大小	font-size: 16px;
font-weight	指定文本的字体粗细	<code>normal</code> ：默认值，表示普通字体粗细。 <code>bold</code> ：表示粗体字。 <code>bolder</code> ：相对于父元素更粗的字体。 <code>lighter</code> ：相对于父元素更细的字体。数值100-900,400普通字体，700粗体。
text-decoration	指定文本的装饰效果，如下划线、删除线等。可以使用关键字如 <code>none</code> 、 <code>underline</code> 、 <code>line-through</code> 来设置装饰效果。	text-decoration: line-through;

代码示例（以内部样式为例）：

```
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    #name{
      /* font-size 设置字体大小 */
      font-size: 20px;
      /* font-size 设置字体粗细 */
      font-weight:400;
      /* 设置字体 */
      font-family: '仿宋';
    }
  </style>
</head>

<body>
  <div id="name">我是div标签</div>
</body>
```

2. 文本属性

属性	含义	说明	代码
color	颜色	可以使用颜色名称、十六进制值或RGB值来指定颜色。	color: red;

属性	含义	说明	代码
line-height	行高	行之间的高度（当其值与div高度一致时，实现垂直对齐；可以结合text-align进行居中）	line-height: 1.5;
text-align	水平对齐	取值：left、right、center 或 justify	text-align: center;

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>学习CSS</title>
  <style>
    p{
      text-align:center;
    }
    .font1{
      font-family:楷体;
      font-size:80px;
    }
    .font2{
      font-family:LiSu;
      color:red;
      font-weight:bold;
      text-decoration:line-through;
    }
    #font3{
      color:greenyellow;
      text-decoration:underline;
      line-height:10;
    }
    #font4{
      text-decoration:line-through;
    }
  </style>
</head>
<body>

  <p class="font1">孙子兵法·计篇</p>
  <p style="color:gray;font-family:楷体;">【作者】孙武【朝代】春秋末年</p>
  <p class="font2">孙子曰:兵者，国之大事，死生之地，存亡之道，不可不察也。</p>
  <p class="font2">故经之以五事，校之以计，而索其情：一曰道，二曰天，三曰地，四曰将，
五曰法。</p>
  <!-- 当一个表情同时设置几个选择器时，如果选择器中的属性有冲突，优先使用权重高的-->
  <p class="font2" id="font3">道者，令民与上同意，可与之死，可与之生，而不危也；天
者，阴阳、寒暑、时制也；地者，远近、险易、广狭、死生也；将者，智、信、仁、勇、严也；法者，曲
制、官道、主用也。</p>
  <p id="font4">凡此五者，将莫不闻，知之者胜，不知之者不胜。</p>
</body>
```

```
</html>
```

3. 背景属性

属性	含义	说明	代码
background-color	背景颜色	可以使用颜色名称、十六进制值或RGB值来指定颜色。	background-color: #f1f1f1;
background-image	背景图	可以指定图像的URL	background-image: url('image.jpg');
background-size	背景图像的大小	<code>auto</code> : 图像保持原始大小。 <code>cover</code> : 图像等比例缩放，以填充背景区域。图像可能会被裁剪。 <code>contain</code> : 图像等比例缩放，以适应背景区域。图像完整显示，可能会留有空白区域。像素值: 指定具体的宽度和高度，例如 <code>200px 150px</code> 。	background-size: cover;
background-position	背景图像的位置	<code>left</code> 、 <code>right</code> 、 <code>center</code> 、 <code>top</code> 、 <code>bottom</code> ，用于指定图像在水平和垂直方向上的对齐方式。指定具体的水平和垂直位置，例如 <code>10px 20px</code>	background-position: center;

```
.bg{
  width:1194px;
  height:834px;
  margin:0 0 0 360px;
  background-image:url(./壁纸5.png)
}
```

4. 盒子属性

在CSS中，可以使用相关的盒子属性（如width、height、padding、border、margin）来控制每个部分的大小、样式和间距。通过调整这些属性的值，可以对元素的布局、外观和空间分配进行精确的控制。

属性	含义	说明
width 和 height	宽度和高度	可以使用像素（px）、百分比（%）或其他单位来指定尺寸。

属性	含义	说明
<code>margin</code>	外边距 (即元素周围的空白区域)	可以指定单个值、水平和垂直值组合，或分别指定上、右、下、左的值。
<code>padding</code>	内边距 (即元素内容与边框之间的空白区域。)	可以指定单个值、水平和垂直值组合，或分别指定上、右、下、左的值。
<code>border</code>	边框	可以指定边框的宽度、样式和颜色。
<code>border-radius</code>	边框圆角	指定圆角的半径值，使元素的边框呈现出圆角效果。
<code>border-style</code>	边框的样式	<code>none</code> ：没有边框样式。 <code>solid</code> ：实线边框。 <code>dashed</code> ：虚线边框。 <code>dotted</code> ：点线边框。 <code>double</code> ：双线边框。 <code>groove</code> ：凹槽边框。 <code>ridge</code> ：脊状边框。 <code>inset</code> ：内嵌边框，显示为向内的效果。 <code>outset</code> ：外嵌边框，显示为向外的效果。
<code>border-color</code>	边框颜色	

```
#box{
  width:200px;
  height:300px;
  background-color:red;
  border:20px;
  border-radius:60px;
  border-style: solid;
  border-color: black;
  margin:200px 800px;
}
```

什么是盒子模型？

盒子模型（Box Model）是CSS中用于描述元素布局和渲染的基本概念。它将每个HTML元素视为一个矩形的盒子，这个盒子包含了元素的内容、内边距、边框和外边距。

盒子模型由以下四个部分组成：

- 内容区（Content）**：盒子的实际内容，例如文本、图像或其他子元素。
- 内边距区（Padding）**：内边距是位于内容区与边框之间的空白区域。它提供了元素内容与边框之间的间距。
- 边框区（Border）**：边框是位于内边距区周围的线条或边界。它用于定义元素的边界和外观。

4. **外边距区 (Margin)** : 外边距是位于边框与相邻元素之间的空白区域。它提供了元素与其他元素之间的间距。

这四个部分形成了一个完整的盒子，它们相互影响和叠加，决定了元素在页面中的布局和尺寸。

盒子模型在网页设计和布局中起着重要的作用，它使我们能够灵活地控制元素的外观和相互之间的关系

盒子模型应用？

`<div>` 元素是一个块级元素，它通常用作容器来组织和布局其他元素。它的默认显示方式是块级显示，它会占据一行并撑满其父元素的宽度。因此，可以使用盒子模型的所有属性来设置 `<div>` 元素的宽度、高度、边框、内边距和外边距等。

`` 元素是一个内联元素，它通常用于包裹文本或内联元素的小片段。它的默认显示方式是内联显示，它不会独占一行，而是根据内容的大小进行自适应。同样，可以使用盒子模型的一些属性来设置 `` 元素的边框、内边距和外边距，但宽度和高度属性对内联元素没有直接效果。

无论是 `<div>` 还是 ``，它们都是盒子模型的一部分，并且可以使用盒子属性来控制它们的布局、边框、内边距和外边距等。然而，由于 `<div>` 是块级元素，而 `` 是内联元素，它们在默认行为和布局上有所不同，因此在应用盒子模型属性时可能会有一些差异。