

Web 信息处理课程学习资料 (第二部分)

模块六：HTML 颜色 (Colors)

在 HTML 和 CSS 中，有多种方式可以定义颜色。

1. 按名称 (Color Names)

- 你可以直接使用标准的颜色名称来设置颜色。
- CSS 支持 140 种标准颜色名称。
- 示例: Red, Orange, Yellow, Cyan, Blue

2. 按 RGB 值 (Red, Green, Blue)

Example

Color	RGB
Red	rgb(255,0,0)
Yellow	rgb(255,255,0)
Green	rgb(0,255,0)
Cyan	rgb(0,255,255)
Blue	rgb(0,0,255)

- 你可以使用 `rgb(red, green, blue)` 的格式来指定颜色。
- 每个参数 (red, green, blue) 定义了颜色的强度，取值范围是 0 到 255。
- 例如，`rgb(255, 0, 0)` 表示红色，因为红色被设为最高值 (255)，而其他颜色为 0。
- 灰色阴影通常通过为三个参数设置相等的值来定义。例如，`rgb(128, 128, 128)` 是中灰色。

3. 按十六进制值 (Hexadecimal Colors)

Example

Color	HEX
Red	#FF0000
Yellow	#FFFF00
Green	#00FF00
Cyan	#00FFFF
Blue	#0000FF

- 你可以使用 `#RRGGBB` 的十六进制格式来指定 RGB 颜色值。
- `RR` (红), `GG` (绿), 和 `BB` (蓝) 是介于 00 和 FF 之间的十六进制值，等同于十进制的 0-255。
- 例如，`#FF0000` 表示红色，因为红色被设为最高值 (FF)，而其他颜色为最低值 (00)。
- 同样，灰色阴影也可以通过设置相等的值来定义。例如，`#808080` 是中灰色。

模块七：CSS (层叠样式表) 入门

1. 什么是 CSS?

- CSS 是 **层叠样式表 (Cascading Style Sheets)** 的缩写。
- 它描述了 HTML 元素在屏幕、纸张或其他媒介上的显示方式。
- CSS 可以一次性控制多个网页的布局，从而节省大量工作。
- 外部样式表存储在 CSS 文件中。

2. 如何使用 CSS

有三种方式可以将样式添加到 HTML 元素中：

1. 内联 (Inline)

在 HTML 元素中使用 `style` 属性。用于对单个元素应用独特样式。

```
<h1 style="color:blue;">This is a Blue Heading</h1>
```

This is a Blue Heading

2. 内部 (Internal)

在 HTML 的 `<head>` 部分中使用 `<style>` 元素。用于为单个 HTML 页面定义样式。

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color:lightgrey;}
h1 {color:blue;}
p {color:green;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

This is a heading

This is a paragraph.

3. 外部 (External)

使用一个或多个外部 CSS 文件。这是最常见的方式，可以将样式保存在单独的 CSS 文件中，从而通过修改一个文件来改变整个网站的外观。

4. Styles.css

```
body {  
    background-color: lightgrey;  
}  
  
h1 {  
    color: blue;  
}  
  
p {  
    color: green;  
}
```

5. page.html

```
<!DOCTYPE html>  
<html>  
<head>  
    <link rel="stylesheet" href="styles.css">  
</head>  
<body>  
  
<h1>This is a heading</h1>  
<p>This is a paragraph.</p>  
  
</body>  
</html>
```

6. Result



外部 CSS 引用方法:

- 在 HTML 文档的 `<head>` 部分添加一个链接指向样式表。
- 使用

```
<link rel="stylesheet" href="styles.css">
```

- `rel` 属性指定了当前文档与链接文档之间的关系。
- `href` 属性指定了样式表文件的位置。

3. CSS 字体与盒模型

字体 (Fonts):

- `color`: 定义文本颜色
- `font-family`: 定义使用的字体
- `font-size`: 定义文本大小

```
<style>
```

```
h1 {  
    color: blue;  
    font-family: verdana;  
    font-size: 300%;  
}  
p {  
    color: red;  
    font-family: courier;  
    font-size: 160%;  
}  
</style>
```

This is a heading

This is a paragraph.

盒模型 (Box Model):

- 每个 HTML 元素周围都有一个盒子。
- `margin`: 定义边框外部的空间 (外边距)
- `border`: 定义一个可见的边框

- `padding`: 定义边框内部的空间 (内边距)

This is a heading

This is a paragraph.

This is a paragraph.

This is a paragraph.

Result

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    border: 1px solid grey;
    padding: 10px;
    margin: 30px;
}
</style>

</head>
<body>

<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>

</body>
</html>
```

4. CSS 选择器 (`id` 和 `class`)

`id` 属性:

```
<p id="p01">I am different</p>
```

```
#p01 {
    color: blue;
}
```

- 用于为一个特殊的、唯一的元素定义样式。
- 在 HTML 中添加 `id` 属性: `<p id="p01">I am different</p>`
- 在 CSS 中使用 `#` 号来选择该 `id`: `#p01 { color: blue; }`

`class` 属性:

Code

```
<!DOCTYPE html>
<html>
<head>
<style>
p.error {
    color: red;
}
</style>

</head>
<body>

<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p class="error">I am different.</p>
<p>This is a paragraph.</p>
<p class="error">I am different too.</p>

</body>
</html>
|
```

Result

This is a paragraph.

This is a paragraph.

I am different.

This is a paragraph.

I am different too.

- 用于为一类（或一组）具有相同特征的元素定义样式。
- 在 HTML 中添加 `class` 属性: `<p class="error">I am different</p>`
- 在 CSS 中使用 `.` 号来选择该 class: `p.error { color: red; }`

模块八：HTML 链接 (Links)

链接几乎存在于所有网页中，允许用户在页面间跳转。

1. `<a href` 链接语法

Example:

```
<!DOCTYPE html>
<html>
<body>

<p><a href="default.html">Visit our HTML tutorial</a></p>

</body>
</html>
```

Result:

Visit our HTML tutorial

- HTML 链接是超链接，使用 `<a>` 标签定义。
- 语法：

```
<a href="url">link text</a>
```

- `href` 属性是 `<a>` 元素最重要的属性，它指明了链接的目标地址。
- **链接文本**是用户可见并可以点击的部分。

2. `target` 属性

- `target` 属性指定在何处打开链接的文档。
- **默认行为**: 在当前浏览器窗口中显示链接的页面。
- 常见值
 - :
 - `_self`: (默认值) 在当前窗口/标签页中打开
 - `_blank`: 在新窗口或新标签页中打开
 - `_parent`: 在父框架中打开
 - `_top`: 在整个窗口中打开

3. 链接样式与颜色

默认颜色:

- 未访问的链接：蓝色带下划线
- 已访问的链接：紫色带下划线
- 激活的链接（点击时）：红色带下划线

自定义样式: 你可以使用 CSS 伪类来更改链接的默认颜色和样式：

- `a:link`: 未访问的链接
- `a:visited`: 已访问的链接

- `a:hover`: 鼠标悬停时的链接
- `a:active`: 激活的链接
- `text-decoration`: 是否下划线

Code

```
<style>
a:link {color:green; background-color:transparent; text-decoration:none}
a:visited {color:purple; background-color:transparent; text-decoration:none}
a:hover {color:red; background-color:transparent; text-decoration:underline}
a:active {color:yellow; background-color:transparent; text-decoration:underline}
</style>
```

Result

You can change the default colors of links

HTML Images

4. 绝对与相对 URL

- **绝对 URL (Absolute URL)**: 一个完整的网址, 例如 `https://www.w3.org/`
- 相对 URL (Relative URL)
 - : 一个本地链接, 不包含 "`https://www.`" 部分, 用于链接到同一网站内的页面
 - `href="default.asp"`: 链接到与当前页面在同一文件夹中的页面
 - `href="/html/default.asp"`: 带斜杠开头, 表示 URL 相对于网站的根目录

5. 特殊链接

- **图像链接**: 将 `` 标签放在 `<a>` 标签内部, 就可以创建一个图像链接。
 - **邮件链接**: 在 `href` 属性中使用 `mailto:` 协议, 可以创建一个能打开用户邮件程序的链接。例如:
`Send email`
-

模块九：HTML 图像 (Images)

1. 图像语法

- 图像使用 `` 标签定义。
- `` 标签是空标签, 只包含属性, 没有闭合标签。
- `src` 属性指定图像的 URL (网址)。
- `alt` 属性提供图像的替代文本, 当图像无法显示时会用到。

2. 图像尺寸

- 应该始终为图像指定宽度和高度，否则页面在加载图像时可能会闪烁。

方法一 (推荐):

使用 `style` 属性：

```

```

方法二:

使用 `width` 和 `height` 属性：

```

```

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
    width:100%;
}
</style>
</head>
<body>




</body>
</html>
```

Result:

It is better to use the `style` attribute (instead of the `width` and `height` attributes), because it prevents internal or external stylesheets to change the original size of an image:



3. 图像位置

- 在其他文件夹: 如果图像存储在子文件夹中，必须在 `src` 属性中包含文件夹名称，例如

```
src="/images/html5.gif"
```

- 在其他服务器: 你可以链接到世界上任何一个网址的图像。例如

```
src="http://www.baidu.com/images/green.jpg"
```

4. 图像映射(图像中可点击区域)(不是很熟)

你可以在一张图片上设定多个“可点击区域”，每个区域都可以跳转到不同的链接

属性	说明
<code>shape</code>	区域形状: <code>rect</code> (矩形)、 <code>circle</code> (圆形)、 <code>poly</code> (多边形)
<code>coords</code>	区域坐标 (根据 <code>shape</code> 类型不同而不同)
<code>alt</code>	区域的替代文本 (无图时显示)
<code>href</code>	点击区域后跳转的 URL

- 图像映射是一个带有可点击区域的图像。
- 使用 `<map>` 标签来定义图像映射。
- `<map>` 标签的 `name` 属性与 `` 标签的 `usemap` 属性相关联，建立图像和映射之间的关系。
- `<map>` 标签内部包含一个或多个 `<area>` 标签，用于定义可点击区域。

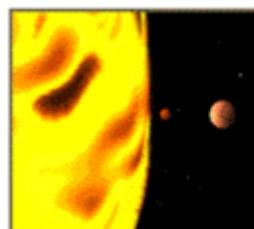
Code

```


<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm">
  <area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.htm">
  <area shape="circle" coords="124,58,8" alt="Venus" href="venus.htm">
</map>
```

Result

Click on the sun or on one of the planets to watch it closer:



模块十：HTML 表格 (Tables)

1. 基本结构

- `<table>`: 定义一个表格
- `<tr>`: 定义表格中的行
- `<td>`: 定义表格中的**数据单元格**。单元格内可以包含文本、图像、列表等各种 HTML 元素。
- `<th>`: 定义**表头单元格**。浏览器默认将表头显示为粗体并居中。
- `<caption>`: 定义表格的标题。必须紧跟在 `<table>` 标签之后插入。

Code

```
<table style="width:100%">
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

Result

Jill	Smith	50
Eve	Jackson	94

2. 表格样式 (使用 CSS)

- **边框 (Border)**: 使用 `border` 属性为表格和单元格添加边框。建议为 `table, th, td` 同时设置边框。

code

```

<!DOCTYPE html>
<html>
<body>

<table border="1" style="width:100%">
    <tr>
        <td>Jill</td>
        <td>Smith</td>
        <td>50</td>
    </tr>
    <tr>
        <td>Eve</td>
        <td>Jackson</td>
        <td>94</td>
    </tr>
    |
    </table>

</body>
</html>

```

Result

Jill	Smith	50
Eve	Jackson	94
John	Doe	80

- **折叠边框**: 使用 `border-collapse: collapse;` 属性可以将单元格边框合并为单一边框。

```

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

```

Jill	Smith	50
Eve	Jackson	94

- **单元格内边距 (Padding)**: 使用 `padding` 属性指定单元格内容与其边框之间的空间。

```

table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
}

th, td {
    padding: 15px;
}

```

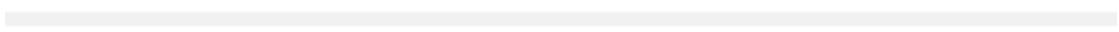
Jill	Smith	50
Eve	Jackson	94

- **单元格外边距 (Spacing):** 使用 `border-spacing` 属性指定单元格之间的空间。此属性在边框折叠时无效。

```

table {
    border-spacing: 5px;
}

```



Jill	Smith	50
Eve	Jackson	94

- **文本对齐:** 使用 `text-align` 属性对齐单元格文本。

```

th {
    text-align: left;
}

```

Firstname	Lastname	Points
Jill	Smith	50
Eve	Jackson	94

3. 单元格合并

- `colspan`: 使用 `colspan` 属性可以使一个单元格横跨多个列。

```

<table style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>555 77 854</td>
    <td>555 77 855</td>
  </tr>
</table>

```

Name	Telephone	
Bill Gates	555 77 854	555 77 855

- `rowspan`: 使用 `rowspan` 属性可以使一个单元格横跨多个行。

```

<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>555 77 854</td>
  </tr>
  <tr>
    <td>555 77 855</td>
  </tr>
</table>

```

Name:	Bill Gates
Telephone:	555 77 854 555 77 855

模块十一：HTML 列表 (Lists)

HTML 提供了有序、无序和描述性列表。

1. 无序列表 (Unordered Lists)

- 以 `` 标签开始。
- 每个列表项以 `` 标签开始。
- 可以使用

style

属性和

list-style-type

来定义项目符号的样式：

- `disc`: 实心圆点 (默认)

Disc:

```
<ul style="list-style-type: disc">
    <li>Coffee</li>
    <li>Tea</li>
    <li>Milk</li>
</ul>
```

- Coffee
- Tea
- Milk

- `circle`: 空心圆

Circle:

```
<ul style="list-style-type:circle">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

- Coffee
- Tea
- Milk

◦ square: 实心方块

Square:

```
<ul style="list-style-type:square">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

- Coffee
- Tea
- Milk

- none: 无项目符号

None:

```
<ul style="list-style-type:none">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Coffee
Tea
Milk

2. 有序列表 (Ordered Lists)

- 以 `` 标签开始, 列表项以 `` 标签开始, 并用数字标记。

```
<ol>
    <li>bananas</li>
    <li>mangoes</li>
    <li>cherries</li>
    <li>plums</li>
    <li>peaches</li>
    <li>grapes</li>
</ol>
```

- 
1. bananas
 2. mangoes
 3. cherries
 4. plums
 5. peaches
 6. grapes

- 可以使用

type

属性来定义编号的类型:

- 1: 数字 (默认)

Numbers:

```
<ol type="1">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

1. Coffee
2. Tea
3. Milk

◦ a: 小写字母

Lowercase Letters:

```
<ol type="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

- a. Coffee
- b. Tea
- c. Milk

- A: 大写字母

Uppercase Letters:

```
<ol type="A">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

- A. Coffee
- B. Tea
- C. Milk

- i: 小写罗马数字

Lowercase Roman Numbers:

```
<ol type="i">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

- i. Coffee
- ii. Tea
- iii. Milk

◦ I : 大写罗马数字

Uppercase Roman Numbers:

```
<ol type="I">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

- I. Coffee
- II. Tea
- III. Milk

3. 列表嵌套与水平列表

- **嵌套列表:** 列表可以嵌套在其他列表内部。

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

- Coffee
- Tea
 - Black tea
 - Green tea
- Milk

- **水平列表:** 使用 CSS 属性 `display:inline` 可以将列表水平显示，常用于制作导航菜单。

```
<!DOCTYPE html>
<html>

<head>
<style>
ul#menu li {
    display:inline;
}
</style>
</head>

<body>

<h2>Horizontal List</h2>

<ul id="menu">
    <li>HTML</li>
    <li>CSS</li>
    <li>JavaScript</li>
    <li>PHP</li>
</ul>

</body>
</html>
```

HTML CSS JavaScript PHP