

《网页基础》

《网页基础》

任务一.初探网站设计

一.网站和网页的概念

- 1.网站
- 2.网页
- 3.网站的分类
- 4.网页的分类

二.HTML语言

三.常见的网页编辑工具

四.网站建设流程

- 1.网站的规划
- 2.网站的建设

任务二.制作简单网页

一.HTML文档基本结构

二.HTML标签

- 1.HTML标签的概念
- 2.HTML标签的类型
- 3.HTML元素

三.知识链接

- 1.HTML注释
- 2.图像标签 ``
- 3.绝对路径和相对路径
- 4.超链接标签 `<a>`
- 5.空链接
- 6.特殊符号实体
- 7.视频标签 `<video>`

任务三.修饰页面元素

一.CSS规则

二.常见的选择器

三.CSS的基本属性

- 1.文本属性
- 2.文本颜色 (`color`)
- 3.文本装饰 (`text-decoration`)
- 4.文本对齐 (`text-align`)
- 5.行高和间距
- 6.文本缩进 (`text-indent`)
- 7.边框样式 (`border-style`)
- 8.边框常用属性
- 9.背景样式

四.CSS样式表的使用和优先级

五.通配符选择器 (*)

六.margin居中

任务四.DIV+CSS布局

- 一.盒子模型 (Box Model)
- 二.盒子模型尺寸
- 三. `<div>` 元素
- 四.文档流
- 五.浮动属性 (`float`)
- 六.响应式布局

任务一.初探网站设计

一.网站和网页的概念

1.网站

网站是指在互联网上，根据一定的规则，使用HTML语言等工具制作的用于展示特定内容的相关网页的集合。

2.网页

网页是构成网站的基本元素，是承载各种网站应用的平台。网页实际上是一系列文件，通过网址来识别与存取。

3.网站的分类

- **按照提供的内容：**咨询类网站，交易类网站，体验类网站，展示类网站，综合类网站
- **按照主题性质：**政府网站，企业网站，商业网站，教育科研机构网站，个人网站，非盈利机构网站以及其他类型网站

4.网页的分类

一般分为静态网页和动态网页。

• 静态网页

不能简单地理解成静止不动的网页，它主要指的是网页中**没有程序代码，只有HTML**（即：超文本标记语言），一般后缀为 `.html`, `.htm` 等。

静态网页的内容一旦做成，不会自动改变。用户直接双击打开，任何人任何时间看到的内容都是不变的。

• 动态网页

是指跟静态网页相对的一种网页编程技术。网页文件中除了HTML标记以外，还包括一些**特定功能的程序代码**。

这些代码可以使得浏览器和服务器可以交互，服务器端根据客户的不同请求**动态的生成网页内容**。

无论网页是否具有视觉上的动态效果，只要是采用了动态网站技术（如 `PHP`, `ASP`, `ASPX`, `JSP` 等）生成的网页都可以称为动态网页。

二.HTML语言

中文名：**超文本标记语言**

英文全称：`HyperText Markup Language`

主流版本：`HTML5`（新增新标签，交互更强，支持音频和视频）

作用：是构成 Web 世界的一砖一瓦，**定义了网页内容的含义和结构**。

三.常见的网页编辑工具

1. **代码编辑器** (`Visual Studio Code` 和 `Sublime Text`)：适合有一定代码基础。
2. **可视化网页编辑器** (`Adobe Dreamweaver` 和 `Wix`)：适合没有编程基础的用户。
3. **在线代码编辑器** (`CodePen` 和 `JSFiddle`)：适合学习和测试。
4. **集成开发环境 IDE** (`Android Studio`, `WebStorm` 和 `IntelliJ IDEA`)：适合全栈开发和复杂项目。

四.网站建设流程

1.网站的规划

- 步骤一：网站建设前的**市场调研**
- 步骤二：建设网站的**目标定位**
- 步骤三：网站的**技术解决方案**
- 步骤四：网站的**内容规划**
- 步骤五：网站的**资源组建**

2.网站的建设

- 第一阶段：网页设计与制作。
- 第二阶段：网站测试与维护。

任务二.制作简单网页

一.HTML文档基本结构

以下是一段 HTML5 代码的基本结构：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">      
  ->
    <title>网页标题</title>
  </head>
  <body>

  </body>
</html>
```

- `<!DOCTYPE html>`：声明了文档类型，是 **HTML5** 文档的标准声明，不区分大小写。
- `<html>`：定义了**整个HTML文档的范围**，是所有其他元素的父元素。
- `<head>`：包含了文档的**元数据**，如字符编码声明、页面标题、引入 `css` 样式和 `JavaScript` 脚本等。

head中常用的标签：

标签	作用
<code><meta></code>	定义字符编码，视口（ <code>viewport</code> ）设置，页面描述等
<code><title></code>	定义 页面的标题 ，显示在浏览器标签页和搜索引擎结果中
<code><link></code>	用于链接外部资源，如 <code>css</code> 样式表
<code><script></code>	用于嵌入或引入 <code>JavaScript</code> 脚本

- `<body>`：包含了用户能在浏览器中看到的**所有内容**，如文本、图片、视频、表格等。

二. HTML标签

1. HTML标签的概念

HTML标签是HTML文档中的基本组成单位，负责定义文档的结构和内容，标签以尖括号 `<` 和 `>` 包裹，分为开始标签和结束标签。

2. HTML标签的类型

- **成对标签** (Paired Tags) : 包括一个开始标签和一个结束标签。

标签	描述
<code><p>...</p></code>	定义段落
<code><div>...</div></code>	定义一个通用容器
<code><h1>...</h1></code>	定义一级标题

- **自闭合标签** (Void Tags) : 不需要结束标签。

标签	描述
<code></code>	插入图片
<code>
</code>	换行
<code><hr></code>	水平分割线
<code><meta></code>	定义元数据

3. HTML元素

HTML元素是由HTML标签及其内容组成的整体，**是构成HTML文档的基本单元**。

常见格式： `<标签名称 属性1="属性值" ...>元素内容</标签名称>`

例如： `<p id="main" class="text">这是一个段落。</p>`

常用的HTML元素：

结构元素	描述
<code><html></code>	定义HTML文档的 根元素
<code><head></code>	包含文档的 元数据
<code><body></code>	包含网页的 可见内容

文本内容元素	描述
<code><p></code>	定义 段落
<code><h1>~<h6></code>	定义 标题 。1号最大，6号最小
<code> / </code>	分别用于 加粗 和 斜体
<code></code>	用于设置 文本内联样式

多媒体元素	描述
<code></code>	嵌入图片 (GIF, JPG, PNG)
<code><audio></code>	嵌入音频
<code><video></code>	嵌入视频

链接元素	描述
<code><a></code>	定义超链接
<code><nav></code>	定义导航链接的集合

三. 知识链接

1. HTML注释

`<!-- 这是一个注释 -->`

注释文字不会显示在网页上，用于代码说明。

2. 图像标签 ``

格式：``

属性	描述
<code>src</code>	图像地址 (绝对路径或相对路径)
<code>alt</code>	图像无法显示时的替代文本
<code>width / height</code>	规定图像的宽度/高度
<code>title</code>	鼠标悬停时的提示文字

3. 绝对路径和相对路径

1. **绝对路径**: 文件在硬盘中的真实完整路径。

2. **相对路径**: 以当前工作目录为基准的路径。

- 同目录: `./file.jpg` 或 `file.jpg`
- 子目录: `data/file.jpg`
- 上级目录: `../data/file.jpg`

4. 超链接标签 `<a>`

格式：``

属性	描述
<code>href</code>	规定链接指向的页面资源
<code>target</code>	打开方式: <code>_blank</code> (新窗口), <code>_self</code> (当前窗口, 默认)

5.空链接

`空链接`：未指定具体目标地址，常用作占位符。

6.特殊符号实体

- `©`：版权符号 ©
- `&nbsp`：不换行空格

锚记链接：

1. 目标位置添加 `id` 属性。
2. 链接 `href` 属性设置为 `#id名`。

7.视频标签 `<video>`

格式：`<video src="路径地址"></video>`

属性	描述
<code>src</code>	视频路径
<code>autoplay</code>	自动播放
<code>controls</code>	显示播放控件
<code>loop</code>	循环播放

任务三.修饰页面元素

一.CSS规则

CSS (层叠样式表) 由选择器和声明模块组成。

```
选择器 {  
    属性: 属性值;  
}
```

注释：`/* 注释内容 */`

二.常见的选择器

1. 标签选择器

直接匹配HTML标签。范围广，适合定义通用样式。

```
p { color: red; }
```

2. 类选择器 (.)

通过 `class` 属性选择。可作用于多个元素，一个元素也可有多个类。

```
.classname { color: red; }
```

3. ID选择器 (#)

通过 `id` 属性选择。仅匹配页面中唯一的元素，优先级高。

```
#idname { color: red; }
```

4. 后代选择器 ()

选择父元素中的所有后代元素，用空格分隔。

```
div p { color: red; } /* 选择div内的所有p */
```

5. 伪类选择器 (:)

用于选择元素的特定状态。

- `a:link` : 未访问链接
- `a:visited` : 已访问链接
- `a:hover` : 鼠标悬停
- `a:active` : 被激活 (按下)

三.CSS的基本属性

1. 文本属性

- `font-family` : 字体名称。
- `font-size` : 字体大小。
- `font-weight` : 字体粗细 (`bold` 加粗)。
- `font-style` : 字体样式 (`italic` 斜体)。

2. 文本颜色 (color)

- 颜色名称: `red`
- 十六进制: `#ff0000`
- RGB: `rgb(255,0,0)`
- RGBA: `rgba(255,0,0,0.5)` (带透明度)

3. 文本装饰 (text-decoration)

- `underline` : 下划线
- `overline` : 上划线
- `line-through` : 删除线
- `none` : 无装饰

4. 文本对齐 (text-align)

`left`, `right`, `center` (居中), `justify` (两端对齐)。

5. 行高和间距

- `line-height` : 行高。
- `letter-spacing` : 字符间距。

6.文本缩进 (`text-indent`)

首行缩进，常用 `2em`。

7.边框样式 (`border-style`)

`solid` (实线), `dashed` (虚线), `dotted` (点线), `double` (双线), `none` (无)。

8.边框常用属性

- `border-width`: 宽度。
- `border-color`: 颜色。
- `border-radius`: 圆角。

9.背景样式

- `background-color`: 背景色。
- `background-image`: 背景图 (`url("img.jpg")`)。
- `background-size`: 大小 (`cover` 覆盖, `contain` 包含)。
- `background-repeat`: 平铺方式 (`no-repeat` 不平铺)。
- `background-position`: 位置。

四.CSS样式表的使用和优先级

优先级：行内样式 > 内部样式 > 外部样式 (就近原则)。

1. **外部样式** (`<link>`): 独立 `.css` 文件，**结构清晰，易于维护，可重用**。
2. **内部样式** (`<style>`): 写在 `<head>` 中，仅对当前页面有效。
3. **行内样式** (`style="..."`): 写在标签内，**优先级最高，但可读性差**。

CSS特性：

- **层叠性**: 多种样式叠加。
- **继承性**: 子元素继承父元素属性 (如文本属性)。
- **优先级权重**:
 - `!important`: 无穷大
 - 行内样式: 1000
 - ID选择器: 100
 - 类/伪类/属性: 10
 - 标签/伪元素: 1
 - 通配符/继承: 0

五.通配符选择器 (*)

选中文档中所有元素。常用于重置样式: `margin: 0; padding: 0;`。

六.margin居中

`margin: 0 auto;` : 实现块级元素在父容器中水平居中。

任务四.DIV+CSS布局

一.盒子模型 (Box Model)

HTML元素可视为一个矩形盒子。

从内到外组成：

1. **内容 (content)**: 实际显示内容。
2. **内边距 (padding)**: 内容与边框之间的空间。
3. **边框 (border)**: 围绕内容和内边距的线条。
4. **外边距 (margin)**: 元素外部的空间，用于控制元素间距。



属性简写 (Padding/Margin) :

- 1个值：上下左右。
- 2个值：**上下、左右**。
- 3个值：上、左右、下。
- 4个值：**上、右、下、左** (顺时针)。

二.盒子模型尺寸

`width` 和 `height` 仅设置 **内容区域 (content)** 的尺寸。

- **总宽度** = `width` + `padding` + `border` + `margin` (左右)
- **总高度** = `height` + `padding` + `border` + `margin` (上下)

三.<div>元素

- **块级元素**：默认独占一行。
- **容器角色**：用于包裹其他元素进行布局。
- **无特定语义**：纯粹的布局工具。

四.文档流

元素默认的排列方式：

1. **块级元素**：从上到下排列。
2. **行内元素**：从左到右排列。
3. **自然排列**：不重叠。

五.浮动属性 (float)

使元素脱离文档流，向左或向右移动。

值	描述
<code>left</code>	向左浮动
<code>right</code>	向右浮动
<code>none</code>	不浮动 (默认)

清除浮动 (`clear`):

防止高度塌陷 (父元素无法被浮动子元素撑开高度)。

值	描述
<code>left</code>	清除左侧浮动影响
<code>right</code>	清除右侧浮动影响
<code>both</code>	清除两侧浮动影响

六.响应式布局

自动适应不同屏幕尺寸。

1. 视口设置 (`viewport`)

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- `width=device-width`: 宽度等于设备宽度。
- `initial-scale=1.0`: 初始不缩放。

2. 媒体查询 (`@media`)

根据屏幕特性应用不同样式。

```
@media (max-width: 768px) {
    /* 手机端样式 */
    body { font-size: 14px; }
}
```

- `max-width`: 最大宽度 (小于等于)。
- `min-width`: 最小宽度 (大于等于)。

出处：23数媒2班 陆云清