

=====

更多干货技术访问小D课堂官网 <https://xdclass.net>

小D课堂QQ客服Vicky: 561509994

公众号搜索: 小D课堂

=====



小D课堂 愿景: "让编程不在难学, 让技术与生活更加有趣"

第一章 欢迎来到网页世界

第一节 趣谈网页成员及学后水平分析

第二节 详细介绍网页三大组成元素和 <!DOCTYPE>标签

- 是拿来声明文档的类型, 在html文档的第一行编写。作用是告知浏览器的解析器用什么文档标准去解析当前文档。**DOCTYPE**不存在或格式不正确会导致文档以怪异模式呈现

- 组成网页的三大标签元素、、

标签是HTML页面的根元素, 该标签的结束标志为

标签包含了文档的元数据(meta), 当meta标签属性设置为utf-8的时候即可定义网页编码格式为utf-8, title标签即定义文档标题 具体设置如下

```
<head>
  <meta charset="utf-8">
  <title>标签定义文档的标题</title>
</head>
```

标签定义文档主体，即网页可见的内容，该标签的结束标志为

第三节 详细讲解结构层、表现层和行为层

- 结构层HTML(HyperTextMarkup Language)
 - 是一个超文本标记语言
 - 负责描绘出网页内容的架构
- 表现层CSS(Cascading Style Sheets)层叠样式表
 - 负责如何显示结构层的有关内容
- 行为层Javascript
 - JavaScript是目前在Web上使用的最主要的客户端脚本语言
 - 是web脚本语言的一个标准
 - 可以对结构层和表现层的内容随意进行更改



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣"

第二章 页面结构层HTML

第一节 介绍HTML文本结构和head、body的作用

- **head作用**：在里面可编写文档元数据、定义文档标题、引入层叠样式表文件、引入js文件、编写css样式
- **body作用**：文档的主体部分，编写显示在网页里的内容，编写javascript脚本，引入js文件

```
<!DOCTYPE html>
<!--html 文档的根元素 -->
<html>
<!-- head标签作用：在里面可编写文档元数据、定义文档标题、引入层叠
样式表文件、引入js文件、编写css样式 -->
<head>
    <meta charset="utf-8">
    <title>我的第一个网页</title>
</head>
<!-- body 文档的主体部分，编写显示在网页里的内容，编写
javascript脚本，引入js文件 -->
<body>
    <p>hello world!</p>
</body>
</html>
```

第二节 介绍良好习惯之代码注释

- 不同地方的注释是不一样的例如：
 - 在编写css的时候注释是要这样写的 `/* 注释内容写在这里 */`
 - 在编写html的时候注释是要这样写的

- 在编写javascript脚本的时候注释是要这样写的 `//`注释内容写在这里 或者 `/*` 注释内容写在这里 `*/`
- 总结：我们写注释是有快捷键的 选中注释内容 然后按Ctrl+? (window系统) command+? (mac系统)

第三节 非构造页面标签元素介绍

- head标签 放style标签 script标签 title标签 meta标签
- meta 页面元信息标签
- script标签 写js的标签或者引入js文件的标签 利用src可引入外部js脚本
- link标签 引入外部样式表
- title 页面标题标签
- style标签 写嵌入样式表的标签

第四节 构造页面必备之HTML标签元素介绍 (一)

常见HTML标签元素介绍一：

- div标签 在网页制作过程中可以把你认为捆绑在一起的东西装在一个
里面,div充当的是一个容器的角色
- p标签 如果想在网页上显示文章或段落，这时就需要
标签了，把文章或段落放到

标签即可。

- **br**标签 需要加回车换行的地方加入 `
` 标签作用相当于word文档中的回车
- **strong**标签 用粗体表示
- **em**标签 用斜体表示 **strong**和**em**都是表示强调
- **span**标签 `` 标签是没有语义的，它的作用就是为了设置单独的样式用的

第五节 构造页面必备之HTML标签元素介绍 (二)

常见HTML标签元素介绍二：

- **ul li** 标签 当你网页需要做一个展示信息列表的时候可以使用 无序列表
- **ol li** 标签 当你网页需要做一个展示信息列表的时候可以使用 有序列表
- **hx(x可以是1-6)** 标签 此标签为标题标签
- **hr** 标签 信息展示时需要分隔的横线时可使用

第六节 构造页面必备之HTML标签元素介绍 (三)

常见HTML标签元素介绍三：

- **a**标签 实现超链接跳转，只要有链接的地方就可以用此标签 此外a标签还可以设置title和target等属性
 - `click here!`
- **img**标签 美丽的网页图片是一个不可划缺的部分，img可以插入图片 如下为使用方式

```

```

第七节 构造页面必备之HTML标签元素介绍 (四)

常见HTML标签元素介绍四：

- **input**标签 文本输入框：常见的type有text、radio、checkbox、button、submit、file、password
- **textarea**标签 文本输入域：用户需要在表单中输入大段文字时，需要用到
- **select option**标签 下拉选择 option里面设置selected="selected"即为默认选中
- **label**标签 label标签不会向用户呈现任何特殊效果，它的作用是为鼠标用户改进了可用性。如果你在label 标签内点击文本，就会触发此控件。就是说，当用户单击选中该label标签时，浏览器就会自动将焦点转到和标签相关的表单控件上（就自动选中与该label标签相关连的表单控件上） 使用方式有如下两种：

第一种: `<label for="控件id名称">`

例子: `<label for="male">男</label>`

`<input type="radio" name="sex" id="male" />`

第二种: `<label>男 <input type="radio" name="sex" id="male" /></label>`

第八节 页面常见元素之HTML表格

- **table** 标签 用来定义 HTML 表格, 可以用作展示数据 **border** 可设置
- **tr** 标签 定义HTML表格中的一行单元格
- **th** 标签 表示HTML表格的表头部分, 该标签中的内容会以粗体显示
- **td** 标签 表示table标签中的单元格
- **thead** 标签 定义了一组HTML表格的头
- **tfoot** 标签 定义了一组HTML表格的尾(表格的页脚)

第九节 重量级元素之HTML表单

表单是可以把浏览者输入的数据传送到服务器端, 这样服务器端程序就可以处理表单传过来的数据, 使用方式如下例子:

```
<form    method="传送方式"    action="服务器文件">
```

- **form** 标签是成对出现的双闭合标签
- **action** 里面传入浏览者输入的数据被传送到地方, 可以是 **json** 文件、**php** 文件等等
- **method** 里面传入的是数据传送的方式 (**get/post**)
- 注意:

1、所有表单控件（文本框、文本域、按钮、单选框、复选框等）都必须放在 `undefined` 标签之间（否则用户输入的信息可提交不到服务器上的）。

2、`method: post/get` 的区别这一部分内容属于后端程序员考虑的问题。完全取决于后端人员，他会告诉你是以什么方式传输的。

第十节 给标签元素一个命名, 让程序更容易找到

- 给标签元素一个 `id` 命名, `id` 是全局唯一的, 就像身份证号码一样
- 给标签元素一个 `class` 命名, `class` 可以多个标签元素共用
- 使用范例如下 (以 `div` 为例) :

```
<!-- id命名, 全局唯一 -->
```

```
<div id="tim"></div>
```

```
<!-- class命名, 可以多个标签元素共用 -->
```

```
<div class="xiaod"></div>
```

```
<p class="xiaod"></p>
```

```
<div class="xiaod"></div>
```


第十一节 行内元素、块级元素和行内块级元素的区别

- 块级元素

- 总是另起一行
- 可以设置其宽度、高度，内外边距
- 在不手动设置宽度的情况下，宽度默认为所在容器的100%（即容器宽度）
- 可以容纳行内元素和其他块元素。
- 常见的块级元素有： `<div>/<h1>~<h6>/<p>//<table>` 等（注意这里只是常见的）

- 行内元素

- 总是和相邻的行内元素在同一行上
- 设置宽高无效，水平方向的padding和margin属性可以设置，但是垂直方向上的无效。
- 默认宽度是他自身内容的宽度。
- 行内元素只能容纳其他行内元素或者文本。
- 常见的行内元素有： `/<a>//
/<input>/<textarea>/<select>//` 等(注意这里只是常见的)

- 行内块元素

- 在行内元素中就有那么几个特殊标签，比如 `/<input>/<td>`，可以给他们设置宽高、对齐属性
- 行内块元素综合了块元素和行内元素的不同特点
 - 和相邻行内元素在同一行，但是之间会有空白缝隙。
 - 默认宽度是他本身内容的宽度。
 - 宽度、高度、行高、外边距以及内边距都可以手动设置。



第三章 房子装修工程师CSS

第一节 详细讲解盒子模型

- CSS盒模型本质上是一个盒子，封装周围的HTML元素，它包括：
margin、*border*、*padding*、*content*(*width*、*height*)
- 盒子模型分两种
 - 标准盒子模型
 - 模型成员：*margin*、*border*、*padding*、*content*
 - *content*不包含其他成员
 - 怪异盒子模型(IE盒子模型)
 - 模型成员：*margin*、*border*、*padding*、*content*
 - *content*包含*border*和*padding*
- 给元素设置*box-sizing*可改变盒子模型类型
 - 标准盒子模型：*box-sizing:content-box;*
 - 怪异盒子模型：*box-sizing:border-box;*

第二节 关于css选择器的权重计算艺术

- 谁优先级高用谁的样式
 - 第一优先级：无条件优先的属性只需要在属性后面使用！

important。它会覆盖页面内任何位置定义的元素样式。 第二优先级：在html中给元素标签加**style**，即内联样式。该方法会造成css难以管理，尽量少用。 第三优先级：由一个或多个**id**选择器来定义。例如，**#id{margin:0;}**会覆

盖**classname{margin:3px}** 第四优先级：由一个或多个类选择器、属性选择器、伪类选择器定义。

如**classname{margin:3px}**会覆盖**div{margin:6px;}** 第五优先级：由一个或多个类型选择器定义。如**div{margin:6px;}**覆盖***{margin:10px; }**

- 第六优先级：通配选择器，如***{margin:6px;}**
- 权重计算值分级
 - 第一等级：代表内联样式，如**style=""**，权值为 1000
 - 第二等级：代表id选择器，如**#content**，权值为100
 - 第三等级：代表类，伪类和属性选择器，如**.content**，权值为 10
 - 第四等级：代表标签选择器和伪元素选择器，如**div p**，权值为1
- 如果遇到样式冲突了，你想设置的样式不起作用时可以利用上述知识提高优先级，增加其权重

第三节 讲解css常见样式属性

- **height** 设置高度
- **width** 设置宽度
- **padding** 设置内边距可以设置一到四个值，例子如下
 - **padding:10px;**
 - 所有四个填充都是 10px
 - **padding:10px 5px;**

- 上填充和下填充是 10px
- 右填充和左填充是 5px
- **padding:10px 5px 15px;**
 - 上填充是 10px
 - 右填充和左填充是 5px
 - 下填充是 15px
- **padding:10px 5px 15px 20px;**
 - 上填充是 10px
 - 右填充是 5px
 - 下填充是 15px
 - 左填充是 20px
- **margin** 设置外边距
- **border** 设置边框
- **color** 设置字体颜色
- **background** 设置背景颜色
- **font-size** 设置字体大小

第四节 详细讲解浮动float跟清除浮动

- 设置浮动布局, **float**的属性有:
 - **left** 元素向左浮动
 - **right** 元素向右浮动
 - **none** 默认值。元素不浮动, 并会显示在其在文本中出现的位置
 - **inherit** 规定应该从父元素继承 **float** 属性的值(这种情况一般可忽略)
- 清除浮动的方法

1. 在父元素上设置`overflow:hidden;`
2. 在父元素上设置伪类, 属性设置为`content: "";``display: block;``clear: both;`
3. 使用空标签设置 `clear:both`

第五节 深度讲解css定位position

- **absolute** 生成绝对定位的元素, 相对于 **static** 定位以外的第一个父元素进行定位。(祖上元素都没定位就相对第一个即最外层的祖先元素进行定位, 如果有祖上元素设置了`position: relative`、**absolute**、**fixed**就相对此祖上元素定位)

元素的位置通过 "`left`", "`top`", "`right`" 以及 "`bottom`" 属性进行规定。

- **fixed** 生成固定定位的元素, 相对于浏览器窗口进行定位。
- **relative** 生成相对定位的元素, 相对于自身正常位置进行定位
- **static** 默认值。没有定位, 元素出现在正常的流中
- **inherit** 规定应该从父元素继承 **position** 属性的值(这种情况一般可忽略)

第六节 常见两栏自适应布局讲解

- 利用浮动进行布局 代码如下:

```
.parent{  
    overflow: hidden; /*清楚浮动*/
```

```

        color: #fff;
    }
    .leftchild{
        float: left; /*设置左浮动*/
        width: 100px;
        height: 100px;
        background: red;
    }
    .rightchild{
        height: 100px;
        margin-left: 100px;
        background: green;
    }

```

```

<div class="parent">
  <div class="leftchild">222</div>
  <div class="rightchild">我是自适应布局我是自适应布局
我是自适应布局我是自适应布局我是自适应布局我是自适应布局我是自
适应布局我是自适应布局我是自适应布局我是自适应布局我是自适应布
局我是自适应布局我是自适应布局我是自适应布局我是自适应布局我是
自适应布局我是自适应布局我是自适应布局我是自适应布局</div>
</div>

```

- 利用定位position进行布局 代码如下:

```

.positionparent{
    position: relative;
}
.posileft{
    position: absolute;
    left: 0;
    width: 100px;
    height: 100px;
    background: grey;
}

```

```
.posiright{
    height: 100px;
    margin-left: 100px;
    background: red;
}

<div class="positionparent">
    <div class="posileft"></div>
    <div class="posiright"></div>
</div>
```

- **css3 flex布局**(这个会在接下来的课程详细讲解flex, 本节课只引出, 想了解可以先百度看一下这个flex是什么)

第七节 常见水平居中垂直居中实现方式

1. 第一种用**text-align: center; height与line-height设置同高**
2. 第二种图片的水平垂直居中 用**text-align: center; height与line-height设置同高**, 然后图片设置**vertical-align: middle;**
3. 第三种利用**position**定位例子如下:

```
.imgbox{
    width: 300px;
    height: 300px;
    position: relative;
    border: 1px solid #465468;
}

.imgbox img{
    position: absolute;
```

```
    left: 0;  
    top:0;  
    right: 0;  
    bottom: 0;  
    height:20px;  
    width:20px;  
    margin: auto;  
}
```

- **css3 flex布局**(这个会在接下来的课程详细讲解flex, 本节课只引出, 想了解可以先百度看一下这个flex是什么)
- <https://www.jianshu.com/p/7baa3d4a1e9c>

第八节 经典三角形实现方式详解

- 等边三角形实现

```
width: 0;  
height: 0;  
border-top:0 solid transparent;  
border-right:50px solid transparent;  
border-bottom:50px solid red;  
border-left:50px solid transparent;
```

- 等腰直角三角形实现


```
width: 0;
height: 0;
border-top:0 solid transparent;
border-right:0 solid transparent;
border-bottom:50px solid red;
border-left:50px solid transparent;
```

第九节 经典圆形实现方式详解(css3属性 border-radius提前了解)

```
.circle{
    width: 300px;
    height: 300px;
    border:1px solid red;
    border-radius: 50%;
}

<div class="circle"></div>
```

第十节 样式写入的常见方式讲解

- 内联样式表
 - `<p style="background:red;font-size:15px"></p>`
- 嵌入样式表(需要写在head标签里面 用style标签包着)

```
<head>

    <title></title>
    <style type="text/css">
        p{
            background-color:yellow;
            font-size:xx-small;
        }
    </style>
</head>
```

- 外部样式表
 - 在head标签里面写link标签引入



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣"

第四章 页面实战之搭建自己的第一个静态页面

第一节 分析页面的构造及讲解该如何写一个完整的页面

第二节 实战编写静态页面(一)

- 编写时移动端页面时加入这行信息配置
-
- 以上这行信息配置作用如下
 - **width=device-width**--宽度为设备宽度，如果你的页面宽度小于设备宽度，就需要修改一下这个属性，不然的话会出现可以页面左右滑动，如ipad ios7中客户端页面。
 - **initial-scale** - 初始的缩放比例
 - **minimum-scale** - 允许用户缩放到的最小比例
 - **maximum-scale** - 允许用户缩放到的最大比例
 - **user-scalable** - 用户是否可以手动缩放

第三节 实战编写静态页面(二)

第四节 回顾过去展望未来和搭建自己的学习路线