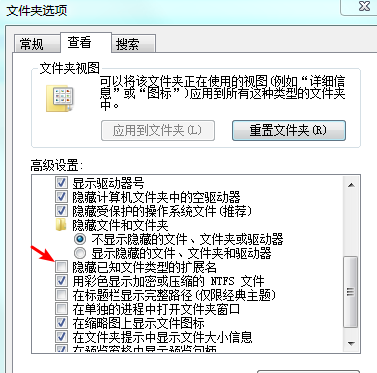
赵蒙蒙

[g-zhaomm@tedu.cn](mailto:g-zhaomm@tedu.cn)

授课阶段:Web前段

授课内容:HTML CSS JavaScript

课前准备:

1. 设置默认浏览器为Chrome
2. 显示文件后缀
   1. 
3. 编辑器(Editplus,sublime3,VSCode)

Web开发

1. Web即为网页,是有浏览器运行的程序.由浏览器,服务器和协议组成.
   1. 浏览器:代替用户发请求,解析并呈现数据
   2. 服务器:接收请求并相应,存储数据
   3. 协议：规范数据的传输方式
2. 主流浏览器
   1. 根据浏览器的引擎（内核）划分为五大浏览器：

chrome safari firefox IE/Edge opera

* 1. 浏览器引擎:渲染引擎、JS引擎

1. HTMl介绍
   1. HyperText Markup Language 超文本标记语言
   2. 标记:也叫标签或元素,以<>为标志，用于书写网页结构
      1. 分类：
         1. 双标签：成对出现，有开始有结束
         2. 单标签：只有开始标签
   3. 语法规范
      1. 标签名不区分大小写，推荐使用小写字母
      2. 双标签必须写完整，如果缺少结束标签或开始标签，浏览器会启动补全，可能影响整体布局；
      3. 所有不使用标签表示的内容都会自动添加到body中原样显示到窗口
      4. 合理的注释和缩进增加代码的可读性

注释语法：<! --注释内容-->

常用标签介绍

1. Head中常用标签
   1. <meta charset=”utf-8”>通过标签属性设置字符编码
   2. <title></title>设置网页标题
   3. <link rel=”shortcut icon” href=”” type=’image/x-icon”>
2. Body中常用标签
   1. 文本标签:标题 段落 加粗 span label
   2. 字符实体:用来处理特殊符号 &copy;&yen;
   3. 容器标签<div></div>,常用于页面结构的划分
   4. 列表标签：
      1. 有序列表
      2. 无需列表
      3. 列表嵌套
   5. 图片与超链接：
      1. URL（统一资源定位符）
         1. 组成：协议 域名 文件目录 文件名
         2. 分类
            1. 绝对路径：从根目录开始逐级查找文件，以盘符或/开头的路径都为绝对路径
            2. 相对路径：从当前所在的文件夹开始查找
      2. 图片标签
         1. <img src=””>在网页中插入图片
            1. 属性：

src 必填，设置图片URL

width/height 设置图片尺寸，取像素值，如果某个方向上缺省，浏览器会根据图片宽高比自适应

title设置图片标题，鼠标悬停在图片上时显示

alt设置图片加载失败时的提示文本

* + 1. 超链接标签
       1. 超链接：借助文本或图片链接到其他资源文件

<a>文本/图片</a>

属性：

href 必填项，指定链接地址

target 设置目标文件的打开方式，默认值\_self，表示在当前窗口打开目标文件

1. 标签之间的关系
   1. 标签嵌套时，产生祖先与后代的关系，父子关系，兄弟关系
      1. 父子关系：外层元素成为父元素，内层元素称为子元素
      2. 祖先和后代：在多级嵌套的结构下，外层所有元素统称为祖先元素，内层元素称为后代元素（直接子元素，间接后代）
      3. 兄弟关系：平级元素之间为兄弟关系
2. 超链接使用
   1. 特殊：描点链接，跳转至指定文件的指定位置
      1. 实现：
         1. 在指定位置添加描点<a name=”7”></a>
         2. 完善超链接的链接地址<a href=”#”>7.人物争议</a>

作业：

1新闻网局部

2实现图片超链接

1. 表格标签
   1. 表格：通过行和单元格实现辅助排版或数据展示
      1. 语法

<table>

<tr>

<td> table data </td>

</tr>

</table>

* + 1. 属性
       1. Table的标签属性

border:设置边框，单元格与会自动添加1px边框

width/height:设置表格尺寸。默认尺寸由内容决定，如果给表格固定尺寸，会自动为单元格分配空间

cellpadding:设置单元格内容与边框之间的距离

cellspacing:设置边框与边框之间的距离

bgcolor：设置背景颜色，取颜色的英文单词

* + - 1. tr的标签属性

bgcolor：设置行的背景颜色

align：设置行中内容的水平对其方式，默认局左，可取left/center/right

valign：设置内容的垂直对其方式，默认居中，可取top/middle/bottom

* + - 1. th/td的标签属性

bgcolor

width/height

align

valign

* + 1. 单元格合并（重点理解）：
       1. 借助单元格的属性、

colspan（collum span）实现跨列合并

rowspan实现跨行合

* + - 1. 取无单位的数值，表示包含自身在内，合并几个单元格
      2. 注意：一旦发生单元格合并，需要删除多余的单元格，保证表格结构的统一
         1. 跨列合并：删除当前行中多余的单元格
         2. 跨行合并：删除其后行中多余的单元格
    1. 表格行分组(了解)

<table>

<thead>

<tr>

<td></td>

</tr>

</thead>

<tfoot></tfoot>

<tbody></tbody>

</table>

使用：

结构化的分组标签根据需要添加，不是必填项

默认所有的行都会自动加入tbody中

推荐书写顺序 thead tfoot tbody，不影响表格结构，优化加载速度

1. 表单标签
   1. 表单：结合表单控件（重点）收集用户信息并提交给服务器
      1. 语法：

<form>

表单控件

</form>

* + 1. 使用：
       1. 表单元素<form></form>负责最终发送数据
          1. 属性：

action 设置提交地址

method 设置数据提交方式，默认为get可设置post

get请求：数据以参数的形式拼接在url后面铭文传输

post请求：数据会被打包传输，可以传输二进制数据

(图片，文件，音视频...)

enctype 设置数据编码类型，对应提交方式，默认为

application/x-www-form-urlencoded,如果涉及二进制数据提交， 需要设置提交方式为post，编码类型为multipart/form-data

* + - * 1. 表单控件（重点）

文本框和密码框

单选钮和复选框

文本与按钮控件的绑定：label for id

隐藏域和文件选择框

隐藏域：用户不需要了解，服务器端必须的数据可以放在隐藏域中传输，页面中对用户不可见

下拉菜单（了解）

多行文本域（了解）

功能性按钮

提交按钮：点击提交，会将表单中的数据按照指定的方式和指定的地址发送给后台

重置按钮：将表单还原至初始化状态

普通按钮：需要结合JS自定义处理

按钮标签

<button>提交</button>

form外部的<button>为普通按钮，需要自定义JS处理事件（用户行为）

form内部的<button>相当于提交

CSS

1. Cascading Style Sheet 层叠样式表，HTML负责书写网页结构和内容，CSS调整页面布局和外观
2. CSS使用方式
   1. 行内样式（内联样式）
      1. 借助于style标签属性书写样式代码

例<h1 style=”color:red;font-size:20px;”></h1>

* 1. 内嵌样式表
     1. 实现结构与样式的分离
     2. 借助于<style type=”text/css”></style>书写CSS样式声明
        1. 例：

<style>

选择器{

属性:值;

属性:值;

}

</style>

* + 1. 选择器：根据标签名，属性值，层级结构等方式匹配文档中的元素，为其应用样式

例：标签选择器，根据标签名匹配文档中所有的该元素

p{

color:red:

}

* 1. 外部样式表

创建.css文件，书写CSS选择器

在HTML中借助<link>引入外部样式表

<link rel=”stylesheet” href=”” type=”text/css”>

1. 样式表特征
   1. 层叠性：同一个元素应用多组样式，共同起作用
   2. 继承性：子元素可以继承父元素中的某些样式

例如：大部分的文本属性都可以被继承，超链接的文本色不能由继承得到，需要单独修改

注意：块元素默认宽度与父元素保持一致

* 1. 优先级
     1. 如果为同一个元素多次设置同一个样式，就产生样式冲突，此时考虑样式表的优先级
     2. 行内样式优先级最高-----内嵌与外链优先级一致，依据代码的书写顺序，后来者居上-----浏览器默认样式和继承样式优先级较低

1. 选择器
   1. 作用：匹配文档中的元素为其应用样式
   2. 分类：
      1. 标签选择器：
         1. 根据标签名匹配文档中的所有该元素
         2. 语法：

标签名{

属性：值

}

* + 1. id选择器：
       1. 根据元素的id属性值匹配文档中唯一的元素
       2. 语法：

#id属性值{

}

* + - 1. 注意：

id属性值自定义，可以由数字，字母，下划线，-，等组成，不能使用数字开头

id具有唯一性

* + 1. class选择器：
       1. 根据元素的class属性值匹配，能够实现样式的服用
       2. 语法：

.class属性值{

}

* + - 1. 特殊用法：
         1. 标签选择器与类选择其结合使用

例：匹配类名为c1的span的元素

span.c1{//标签名在前

}

无效写法： .c1span{}//匹配类名为c1span的元素

* + - * 1. class属性可以取多个属性值，使用空格隔开，同时应用多个类选择器的样式
    1. 群组选择器：
       1. 同时匹配各个选择器的元素，统一设置样式
       2. 语法：

选择器1,选择器2,选择器3{

}

* + 1. 后代选择器
       1. 依托标签的层级结构，匹配所有的后代元素
       2. 语法：

//选择器1表示外层元素，选择器2表示内层元素

选择器1 选择器2{

}

* + 1. 子选择器
       1. 依托于层级关系匹配直接子元素
       2. 语法：

选择器1>选择器2{

}

* + 1. 伪类选择器
       1. 为元素的某种状态设置样式，需要与基础选择器结合使用，表示元素的某种状态
       2. 分类：

:link 超链接访问之前

:visited 超链接访问后

:hove 鼠标悬停

:active 鼠标点按不抬起(激活状态)

:focus 焦点状态(输入框处于编辑状态)

常用设置：

a{//统一设置超链接四种状态下的默认样式

Color:gray;

Text-decoration:none;

}

a:hover{

Color:

Background-color:

}

练习：

1创建超链接，设置默认文本色为白色，背景色为橘色，取消下划线；鼠标悬停时修改背景色

2创建div，设置尺寸为200\*200（width/height），设置背景色；实现鼠标悬停时修改尺寸为300\*300，修改背景色

3下拉菜单的隐藏与显示

display:none;//设置隐藏

display:block;//设置显示

1. 选择器的优先级
   1. 使用不同的选择器为统一元素设置样式，发生冲突时考虑选择器的优先级，有权重决定：
      1. 标签选择器1
      2. 类/伪类 10
      3. id选择器 100
      4. 行内样式 1000+
      5. 注意：
         1. 群组选择器中，各个选择器由自身的权重决定：后代，子代选择器的优先级由各组成部分的权重相加得到
         2. 例：

body,#d1,.c1{}

body div{}

body #d1{}

标签分类及嵌套

标签分类：

1. 块元素
   1. 特点：独占一行，不与其他元素共行；可以手动调整宽高，默认宽度与父元素保持一致

例：h1~h6 p div ul ol li form table (默认宽度由内容决定)

1. 行内元素
   1. 特点：可与其他元素共行显示：不能手动调整宽高，尺寸由内容决定
   2. 例：a b strong span label i s u sub sup
2. 行内块元素
   1. 特点：既可以共行显示，又能手动调整宽高
3. 转换元素类型
   1. 属性：display
   2. 取值：inline(行内元素)/block(块元素)/inline-block(行内块元素)/none(隐藏)

标签嵌套：

1. 块元素中可以嵌套任意类型的元素
   1. 特殊：段落标签中不能嵌套块元素，只能嵌套行内/行块内元素
2. 行内元素中尽量只嵌套行内（行内块）元素

尺寸与颜色

1. 尺寸单位
   1. px默认使用像素
   2. % 默认参照父元素相应的属性值进行百分比计算
   3. em参照父元素的字体大小计算 默认1em=16px
   4. rem参照根元素的字体大小计算默认1rem=16px
2. 颜色取值
   1. 取颜色的英文单词：red blue 等
   2. rgb(r,g,b)取三原色设置，每种颜色取值范围0~255

black rgb(0,0,0)

white rgb(255,255,255)

red rgb(255,0,0)

* 1. Rgba(r,g,b,alpha)设置三原色及透明度，alpha取值0(透明)~1(不透明)
  2. 通过16进制表示：将三原色的范围0~255转换成十六进制表示(00~ff)，十六进制的取值范围：0~9 a~f
     1. 使用：
        1. 长十六进制，每两位为一组代表一种三原色

例：#ff0000 红色

* + - 1. 短十六进制，采用三维字符代表三原色，浏览器会自动对每一位进行重复，最终完善为长十六进制

例#0f0 等于#00ff00

内容尺寸与溢出处理

1. 尺寸属性
   1. wideth/height:取像素值或者百分比，用来设置内容区域的大小
2. 溢出处理
   1. 溢出：内容超出内容尺寸范围，称为溢出，默认溢出部分仍然可见，可借助overflow属性调整溢出内容的显示
   2. 属性：overflow
   3. 取值：
      1. visible（默认可见）
      2. hidden（溢出部分隐藏）
      3. scroll（强制在水平和垂直方向添加滚动条）
      4. auto（自动在溢出方向添加可用的滚动条）
      5. 自动

盒模型

1. 盒模型包含：内容区域、padding（内边距）、border边框、margin（外边距），关系到文档中的实际占位，影响排版布局
2. 内容区域：使用width/height属性设置（按钮元素除外）
3. 边框设置
   1. 边框实现：
      1. border:width style color;
      2. 边框样式可取 solid（实线）/dashed（虚线）/dotted（点线）/double（双线）
   2. 单边框设置
      1. border属性用于统一设置四个方向的边框，可以借助以下属性实线单边框设置：
         1. border-top 上边框
         2. border-right 右边框
         3. border-bottom 下边框
         4. border-left 左边框
      2. 取值：width style color;
   3. 网页三角标设置
      1. 设置元素尺寸为0
      2. 分别添加四个方向的边框，保留一个可见色，其他为透明色
      3. 三角标的大小可通过边框宽度调整
   4. 轮廓线
      1. 边框在文档中实际占位，轮廓线不占位
      2. 属性：outline
      3. 取值：color style width;
      4. 特殊：输入框在编辑状态自带轮廓线，可以设置outline:none,取消轮廓线
   5. 圆角边框
      1. 属性；border-radius
      2. 取值：
         1. 取像素值px，设置圆角的半径
         2. 取百分比%，参照元素自身宽度属性计算圆角半径
      3. 取值范围：
         1. 统一设置四个角的圆角半径，取值最大取50%，改变元素形状（超过50%仍然显示50%的效果
         2. 单独设置每个角的圆角半径0%~100%

联系：

创建div，设置尺寸和背景颜色

在鼠标悬停时，修改尺寸，背景色，修改元素形状