1. CSS入门知识
   1. 关于WEB标准

WEB标准不是某一个标准，而是一系列标准的集合。

网页主要由四部分组成：内容（Content）、结构（Structure）、表现（Presentation）和行为（Behavior）。

1） 内容：就是制作者放在页面内真正想要让访问者浏览的东西，如：图片、文本、多媒体等（内容为王）

2） 结构：使内容更加更加具有逻辑性与易用性，更清晰易懂

3） 表现：用于修饰内容的外观的样式，称为表现，使网页更加美观

4） 行为：网页中的交互及操作效果，主要通过脚本语言完成事件+动作

对应的标准有三方面：

结构化标准语言主要包括HTML、XHTML和XML，

表现标准语言主要包括CSS，

行为标准主要包括对象模型 W3C DOM、JavaScript、ActionScript等。

WEB标准的目的：

创建一个统一的用于Web表现层的技术标准，以便通过不同浏览器或终端设备向最终用户展示统一的信息内容。

|  |  |
| --- | --- |
| 结构 | HTML / XHTML / XML |
| 样式 | CSS |
| 行为 | JavaScript |

* 1. 什么是CSS

CSS是Cascading Style Sheets的简写，即层叠样式表，简称样式表。

* 1. CSS的主要作用
     + 1. 遵循W3C标准，符合WEB2.0标准。
       2. 减少重复格式化，减少网页体积，加快下载和访问速度。
       3. 符合内容与表现形式分离的原则，方便于搜索引擎抓取到有用的内容。
       4. 便于更新和维护，成千上万的网页只需要修改CSS便可以更改显示外观。
       5. 浏览器干扰相对较小，实现一些HTML格式化不能实现的高级功能。

1. CSS的基本语法
   1. 第一个CSS样式

需求：网页中显示“源码时代教育机构”，并且对文本设置为红色、加粗、改变字体大小。

方案1：不使用使用CSS的写法：

<font size="6" color="red"><b>源码时代教育机构</b></font>

方案2：使用CSS的写法：

<!--

所有的标签都有style属性, 这个属性代表的就是修改当前这个元素的样式

颜色 color: red;

加粗font-weight: bold;

字体大小font-size: 24px;

-->

<span style="font-size:30px; font-weight=bold; color=red;">源码时代教育机构</span>

* 1. CSS的语法结构

**CSS的定义是由三个部分构成：选择器（selector），属性（properties）和属性的取值（value）**

选择器名称

{

属性名称1：属性的值1；

属性名称1：属性的值1；

……

}

**注意事项：**

1.在CSS中插入注释来说明你代码的意思，注释有利于你或别人以后编辑和更改代码时理解代码的含义。在浏览器中，注释是不显示的，CSS注释以"/\*" 开头，以"\*/" 结尾。

2.定义多个值时，浏览器按照从前向后顺序选择属性值。如果第1个值有效，则尝试使用，如果无效，则使用第2个，依次类推。 如：font-family: 'arial black', '微软雅黑';

CSS样式一般放到HTML的head标签中，写在成对的<style>标签中，标签可以设置一个type属性并设置值为text/css，目的是告诉浏览器<style>标签中的内容是css样式。比如：以下样式定义了一个新闻标题的样式，分别对字体、字号、粗体字进行了定义。

<html>

<head>

…..

<style type="text/css">

.NewsTitle

{

font-family: "微软雅黑";

font-weight: bold;

font-size: 24px;

}

</style>

</head>

……

</html>

* 1. CSS的三种引用方式

使用CSS有三种方式： 内联式CSS、 内部式CSS 、外链式CSS

* + 1. 内联式CSS

内联定义即是在对象的标记内使用对象的style属性定义适用其的样式表属性。

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>新闻网页</title>

</head>

<body>

<p style="font-family: '微软雅黑'; font-weight: bold; font-size: 24px; ">新闻标题</p>新闻正文

</body>

</html>

* + 1. 内部式CSS

在HTML文档的<HEAD>标记中插入一对<STYLE>...</STYLE>标签。

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

<style type="text/css">

.NewsTitle

{

font-family: "微软雅黑";

font-weight: bold;

font-size: 24px;

}

</style>

</head>

<body>

<p class="NewsTitle">新闻标题</p>新闻正文

</body>

</html>

* + 1. 外链式CSS
* 先建立外部样式表文件（\*.css），然后在HTML中使用的link标签将其引入到网页中。

外部news.css文件

.NewsTitle

{

font-family: "微软雅黑";

font-weight: bold;

font-size: 24px;

}

* 新闻网页文件news.html文件

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>新闻网页</title>

<link rel="stylesheet" type="text/css" href="news.css"/>

</head>

<body>

<p class="NewsTitle">新闻标题</p>新闻正文

</body>

</html>

1. CSS选择器的类型
   1. 什么是选择器？

在CSS 中，选择器是一种匹配模式，用于**选择需要添加样式的目标对象**。

注： 行内样式不需要用选择器， 内部或外部CSS需要用选择器。

* 1. 选择器的语法

基本格式如下：

**selector {property: value}**

**选择器 {属性：值}**

选择器名称

{

属性名称1：属性的值1；

属性名称2：属性的值2；

属性名称3：属性的值3；

……

}



* 1. 基本选择器（元素选择符）
     1. 类选择器

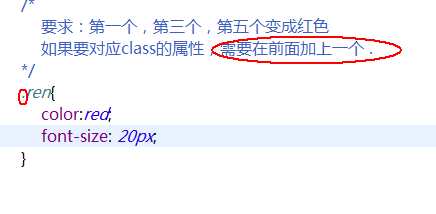
所有的HTML标签都有class属性，为这个class设定一个值，然后就可以使用类选择器来匹配这一个值。并对其应用样式。

作用：选择指定了class属性值的标签

语法：

**.类的名称**{ 样式表 }

示例：



注意：使用类选择器必须在值的前面加一个点符号（.）

* + 1. ID选择器

所有的HTML标签都有一个id属性

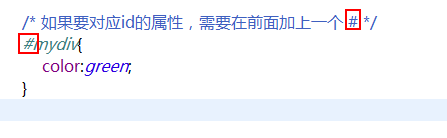
我们可以为这个id设定一个值，然后再写选择器来匹配这一个值

作用：选择指定的ID属性值的标签

语法：

**#ID值**{ 样式表 }

示例：



*id使用的 # ，class 使用的是 .*

***通常id是唯一的（只能够有一个元素来匹配）***

* + 1. 标签(元素)选择器

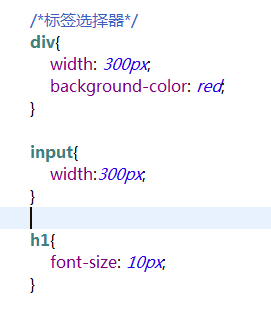
**必须是HTML语法中固有的标签，而不能使自己定义的名称。**

作用：选择HTML网页中所有指定的标签，并对其应用样式

语法：

**标签**{ 样式表 }

示例：



* + 1. 通用选择器

\* 星号选择器，也称为通用选择器

作用：选择HTML页面上所有的元素

语法：

**\***{ 样式表 }

示例：



通用选择器一般我们不会使用，因为它会对页面上所有的元素的样式进行渲染，这样就会使页面的渲染速度变慢。

* 1. 复合选择器（关系选择符）
     1. 包含选择器E1 E2（后代关系）

作用：选择所有被E1包含的E2。

语法：

E1 E2

示例：

|  |
| --- |
| table td { font-size:14px; }  div .sub a { font-size:14px; } |

* + 1. 子选择器 E1>E2（父子关系）

作用：选择所有作为E1子对象的E2。

语法：

E1 > E2

示例：

|  |
| --- |
| body > p { font-size:14px; } |

**注意： 子选择器与包含选择器有区别，子选择器只选中下一代，包含选择器可以选中所有的后代（不限层级）。**

* + 1. 选择器分组 E1,E2,E3（并列关系）

作用：将同样的样式应用于多个选择器，可以将选择符以逗号分隔的方式并为组。

语法：

E1,E2,E3

示例：

|  |
| --- |
| .td1,div a,body { font-size:14px; } |

* + 1. 相邻选择器E1+E2（邻居关系）

选择紧贴在E1元素之后E2元素。相邻选择符只会命中符合条件的相邻的兄弟元素。

示例：

|  |
| --- |
| <style>  p + p {  color: #f00;  }  </style>  <div class="test">  <h3>这是一个标题</h3>  <p>这是一个文字段落</p>  <p>这是一个文字段落</p>  <h3>这是一个标题</h3>  <p>这是一个文字段落</p>  <h3>这是一个标题</h3>  <p>这是一个文字段落</p>  <p>这是一个文字段落</p>  </div> |

说明：如果使用p + p{color:#f00;}，那么p2, p5将会变成红色；如果使用p ~ p{color:#f00;}，那么p2,p3,p4,p5将会变成红色；

* + 1. 兄弟选择器E1~E2（兄弟关系）

选择E元素后面的所有兄弟元素F。

与相邻选择符不同的是，兄弟选择符会命中所有符合条件的兄弟元素，而不强制是紧邻的元素。

* 1. 属性选择器（了解）
     1. E1[attr]

选择具有attr属性的E1

示例：

|  |
| --- |
| /\* 所有具有title属性的div标签 \*/  div[title] { color: blue; } |

* + 1. E1[attr=value]

选择具有attr属性且属性值等于value的E1

示例：

|  |
| --- |
| span[class=demo] { color: red; } |

* 1. 伪类选择器
     1. 鼠标的四种状态（重点）

作用：<a>标记，超链接有4种不同的状态：未访问链接(link)、鼠标悬停状态(hover)、已访问链接(visited)、当前点击的活动链接(active)。CSS允许对于元素的不同状态，定义不同的格式化信息。

**E:link 未访问的链接**

**E:visited 已访问的链接**

**E:hover 鼠标悬停状态**

**E:active 选定的激活状态**

示例：

|  |
| --- |
| /\* 按规则的写法 \*/  a:link{color:black;text-decoration: none;}  a:visited{color:black;text-decoration: none;}  a:hover{color:red;text-decoration: underline;font-size:30px;}  a:active{color:black;text-decoration: none;}  /\* 通常我们是这样写的：\*/  a:link,a:visited{color:black;text-decoration: none;}  a:hover{color:red;text-decoration: underline;font-size:30px;}  /\* 或者：将所有的a标签的状态都修改成下面这样 \*/  a{color:black;text-decoration: none;}  /\* 当鼠标称上来的状态我们进行单独的修改 \*/  a:hover{color:red;text-decoration: underline;font-size:30px;} |

注意：

1. :active 这个伪类现在已经不常用了。
2. hover状态必须写在最后
3. Link visited hover active

**a.red:link class等于red的a元素**

**a.red:visited class等于red的a元素**

**a.red:active class等于red的a元素**

**a.red:hover class等于red的a元素**

**简化：**

**a.red:link, a.red:visited**

**a.red:hover**

**最精简：**

**.red**

**.red:hover**

实例如下：

<style type="text/css">

.red

{

color:#666666;

text-decoration: none;

}

.red:hover

{

color:#ff3300;

text-decoration: underline;

}

</style>

<a href="# " class="red">我是超链接的文本</a>

* + 1. E:focus 焦点

设置元素在成为输入焦点（该元素的onfocus事件发生）时的样式。

实例如下：

input:focus {

background: #f6f6f6;

color: #f60;

border: 1px solid #f60;

outline: none;

}

* + 1. E:after和E:before

E:after 选择器在被选元素的内容后面插入内容。

E:before 选择器在被选元素的内容前面插入内容。

通常使用 content 属性配合，来指定要插入的内容。

示例：

|  |
| --- |
| <style type="text/css">  p:before  {  font-size: 18px;  color: red;  font-weight: bold;  content: "以下是歌词：";  }  p:after  {  font-weight: bold;  content: '----歌词结束了。';  }  </style>  <p>歌词的内容，歌词的内容，歌词的内容，歌词的内容，歌词的内容，歌词的内容，歌词的内容</p> |



1. CSS的继承与优先级

### 1. CSS的继承特性

HTML文档以树形结构进行组织，各元素之间是一种层次关系，这种层次关系同样反映在样式表的应用中。**具有层次关系的元素之间，内层元素将继承外层元素的样式，多个外层元素中定义的样式将叠加到内层元素**。

HTML中，<body>是其他元素的容器，是其他元素的最外层元素，所以<body>元素中定义的样式将影响其他所有元素的显示格式。

能被继承的CSS属性：

**文本相关的属性是继承的**

text-align、color、text-indent、font-family、font-size

font-style、font-weight、 letter-spacing、word-spacing

text-transform、line-height等

**列表相关的属性是继承的**

list-style、 list-style-image、list-style-position、list-style-type

演示：

|  |
| --- |
| **ul{**  **list-style-type: none;**  **}**  **/\* 没有定义时内外层的列表都不显示符号 \*/**  **#inner li{**  **list-style-type: circle;**  **}**  **<ul id="out">**  **<li>列表内容1</li>**  **<li>列表内容2</li>**  **<li>列表内容3</li>**  **<li>列表内容4</li>**  **<li>列表内容5</li>**  **<ul id="inner">**  **<li>内层列表1</li>**  **<li>内层列表2</li>**  **<li>内层列表3</li>**  **<li>内层列表4</li>**  **</ul>**  **</ul>** |

### 2. CSS的优先级

|  |
| --- |
| **CSS样式是有优先级的，遵循顺序：**  **行内样式 > id选择器 > 类选择器 > 标签选择器**  **在选择器优先级相同的情况下，我们参照就近原则** |