# DAY04 HTML

# 1 表单

### 特点

|  |
| --- |
| 1. 提供了一些可视化的输入控件 2. 会自动收集整理用户输入的内容，并提交给服务器 |

### 语法

|  |
| --- |
| <form></form>  可以添加的属性:  action=“url” 表示会将表单收集的数据提交到哪个服务器  目前我们可以先写一个#,表示提交到当前页面 |

### input元素

|  |
| --- |
| **公共属性：**  type 设置input元素的类型，默认值是text  value 用来保存用户输入的值，用于后期提交给服务器  如果控件是按钮，value用来设置按钮上显示的文本  name 为控件起名字,注意：form表单必须起name否则提交不了数据 |
| 1. **文本框与密码框**   type=“text” 普通文本输入框  type=“password” 密码框  属性：  maxlength="5" 设置输入数据的最大长度  placeholder="请输入用户名" 提示文字  value="jack" 表示给当前输入框设置了一个默认值jack  这个属性不写默认也有，默认值为空，到时用户输入什么数据，value就保存什么值 |
| 1. **按钮**   type=“submit”提交按钮，会自动手机整理用户输入的数据，提交  type=“reset 重置按钮，会将表单的控件恢复到初始状态，注意不是清空  type="button" 普通按钮，后期会绑定JS代码  注意: 按钮的value属性是用来设置按钮上显示的文字  H5新增了一个表示按钮的标签 <button></button>等价于上面的type=”button” |
| 1. **单选与多选框**   type="radio" 单选框  type="checkbox" 多选框  **属性：**  name【必须加】 为控件起名并分组，一组单选框/多选框的名称必须相同才能实现效果  value【必须加】 不加提交的值是on  checked 表示当前项默认被选中 |
| 1. **文件上传**   请上传您的文件<input type="file" multiple>  multiple可以实现文件上传的多选效果 |

1. **h5新增的input控件**

|  |
| --- |
| **数字控件**  <input type="number" max="30" min="10" step="3"/>  min 最小值 max最大值  step步长，一次递增/递减的大小，默认步长为1 注意：这个输入框无法阻止用户自行输入数据 |
| 颜色控件 <input type="color"/>  提供了一个取色器，可以选择不同的颜色，默认是黑色 |
| 日期控件 <input type="date"/>  可以年月日，后期如果想要其他更加精美的日期效果，可以使用其他的插件 |
| 月份控件 <input type="month"/> 选择年-月 |
| 星期控件 <input type="week"/> 选择年-周 一年有52周 |
| 搜索控件 <input type="search"/> 提供了一个快速删除所有内容的小叉叉 |
| 范围控件 <input type="range" min="10" max="30" step="7"/>  min最小值 max最大值 step步长，表示数字滑动的跨度  作用:可以控制区间，比如音量 地图缩放 进度  注意：设置步长时应该选取可以除得尽的值，否则区间中会有一部分值无法被选中 |

### select下拉选择框

|  |
| --- |
| <select name="food" multiple>  <option value="1" >四川火锅</option>  <option value="2">东北乱炖</option>  </select>   1. 提交时如果没有给option设置value值，提交的就是option标签之间的文本 2. selected 设置当前option默认被选中 3. multiple 设置当前下拉选择框可以多选 |

### textarea多行文本域

|  |
| --- |
| <textarea rows="10" cols="50"></textarea>  rows="10" 文本域的行数，改变的是高度  cols="50" 文本域的列数，改变的是宽度  注意：默认的文本域大小可以被用户随意拖拽改变，想要禁用拖拽：  style="resize:none;" |

### label关联控件

|  |
| --- |
| 用于将form表单中的文本与控件进行关联  点击文本，效果如同点击控件  <label for="要关联控件的id值"></label>  <input type="checkbox" name="yes" id="y" value="1">  <label for="y">请同意此协议</label> |

# 2 浮动框架

|  |
| --- |
| iframe 用于在一个html页面中，引入其它的html页面  <iframe src="https://www.codeboy.com/xuezi/admin/login.html"></iframe>  **属性：**  src="" 被引入资源的路径  width="500px" 被引入资源在本页面显示的宽度  height="450px" 被引入资源在本页面显示的高度  frameborder="0" 去掉被引入资源的边框线  scrolling="no" 是否需要拖拽条，no表示不要滚动条  **优势与劣势：**   1. 我们可以直接引入已经写好的页面，比较方便 2. 被引入的页面使用了多次，复用性较高 3. 样式不好控制 4. 有额外的链接，请求的次数会增加，速度会变慢 |

# 3 结构化标签

|  |
| --- |
| h5新增了带有结构语义的标签，来取代div  虽然在用户看来之前使用div没有区别，但是带语义的标签可以增加代码的可读性  很方便的实现页面各个部分的划分，让网络爬虫更快找到  但注意: 低版本的浏览器会不兼容 |
| <header> </header> 定义网页的头部或者某个区域的顶部  <nav></nav> 定义网站的导航栏  <section></section> 定义网页的主体内容  <article></article> 定义与文字相关的内容，比如文章 论坛 回帖  <aside></aside> 定义网页的侧边栏比如分类 广告  <footer></footer> 定义网页的底部或者某个区域的底部 15:06 |

# 4 CSS

## CSS概述

|  |
| --- |
| CSS:Cascading Style Sheet 级联样式表  1995诞生，2010年左右开始陆续推出最新版本V3.0  样式：我们可以控制元素的颜色 背景 大小 位置 边框 阴影...  HTML—内容 CSS—样式层  我们要是有CSS来修饰HTML元素 |

## CSS编写的不同位置

### 行内样式 inline style

|  |
| --- |
| <标签 style="color:red;"></标签>  注意：   1. 属性名与属性值之间用冒号连接，结束后记得写分号！！！ 2. 行内样式只属于当前元素，不能给其他元素分享 3. 由于只渲染当前元素，优先级非常高，但不利于修改，复用性很差 |

### 内部样式 inner style

|  |
| --- |
| 在html文件中head标签里写style标签，style标签中的就是内部样式  <style>  选择器 {  CSS样式代码  }  </style>  内部样式可以供页面中多个元素共享，不能共享给其他页面的元素 |

### 外部样式 outer style

|  |
| --- |
| 需要把样式写在一个独立的.css文件中  需要此样式的html页面在自己的head标签中使用link标签引入该外部样式即可  <link rel="stylesheet" href="01.css">  rel="stylesheet" 表示当前引入的文件是一个样式表【必写】  href="01.css" 被引入的CSS文件的url  外部样式的范围比较大，谁引入，谁生效，可以被多个页面共享 |

### 三种样式的比较

|  |
| --- |
| **生效的范围：**  外部样式>内部样式>行内样式  **样式的优先级：**  行内样式>（内部样式=外部样式）>浏览器自带样式 |

## 3 选择器

CSS选择器用在内部或外部样式中，告诉浏览器，将选中哪些元素应用样式

|  |
| --- |
| 选择器 {  CSS样式  } |

### 1. 通用选择器

|  |
| --- |
| \* 表示所有 \*{ }表示当前页面上所有的标签都应用该样式  \* {  color: red;  }  因为选中的是所有元素，所以优先级非常低，性能低 |

### 2. 标签选择器

|  |
| --- |
| 选中某类特定的HTML元素，选择器的名字就是标签的名字，注意标签名不加尖括号  div {  background-color: yellow;  }  优点：快速为当前页面中同种类型的标签统一样式  缺点：会把当前页面所有的这类标签选中，选择太广泛，不能设计差异化样式 |

### 3. ID选择器（为元素设定样式时推荐少用）

|  |
| --- |
| 选中唯一一个指定的元素，注意：ID值与#号之间不允许有空格！  #d2 {  color: blue;  }  注意：要先在标签中加入id属性，再在CSS中选中  优点：能够非常直观的找到该元素  但是ID选择器推荐少用，因为ID值经常用于为元素绑定JS脚本，较少用于样式 |

### 4. 类选择器

|  |
| --- |
| 选中某一类指定的元素，这个类名是可以自定义的  一个class属性的值，可以写多个，值与值之间使用空格隔开  注意：使用选择器时点(**.**)与class值之间不能有空格！  .text-danger {background-color: red; color: white;}  .text-success {background-color: green; color: white;}  <span class="text-danger">用户名不能为空</span>  <span class="text-danger">密码不能为空</span>  <span class="text-danger">两次密码输入不一致</span>  <span class="text-success">用户名可以使用</span>  <span class="text-success">密码可以使用</span>  <span class="text-success">两次密码输入一致</span>  注意：要先在标签中加入class属性，再在CSS中选中  优点：复用性很高，在项目中经常使用 |

### 群组选择器

|  |
| --- |
| 使用本选择器可以一次选中多个不同的元素，使用逗号分隔，可以有空格  h4, #p, .r {  color: red;  } |
| 注意: 群组选择器不限制选择器必须是同一种类型，可以使用不同的类型  只要满足其中任意一个选择器就可以应用该样式 |

### 伪类选择器

“假的类”—伪类的名字不可以自定义，是固定的，表示选中某个时刻/状态

|  |
| --- |
| :hover{ } 选中”鼠标悬停在元素上方时”的状态 |
| :active{ } 选中”元素被激活时”的状态(鼠标按住元素不松开) |
| :link{ } 选中”超链接未被访问过”的状态 |
| :visited{ } 选中”超链接已经被访问过”的状态  注意：超链接的4个伪类有顺序： :link -> :visited -> :hover -> :active  如果没有按照这个顺序，有的伪类会不生效 |
| :focus{ } 选中”输入框获得焦点时”的状态 |

# 拓展

|  |
| --- |
| CSS会在控制台报错，黄色叹号+删除线才是报错，常见错误：  Unknown property name 未知的属性名—属性名写错了  Invalid property value 无效的属性值—属性值写错了  注意：只有删除线的不是报错，是样式冲突了未生效 |

# 作业

1. **思考并实验:**

相同选择器，不同的样式，会如何渲染？

相同选择器，相同的样式，会如何渲染？

id选择器 标签名选择器 类选择器 通用选择器 优先级如何排列?

1. 写一个用户名输入框，提示文字“请验证自己的用户名是否正确”

样式要求：提示文字的默认颜色为透明色transparent

当输入框获得焦点时，将输入框的背景颜色改为黄色，提示文字字体颜色改为红色

1. 仿写学子商城首页导航条的效果，实现效果：

默认颜色#ccc 悬停颜色#B8B8B8 访问过后颜色#ccc

激活颜色 #0aa1ed

