

一.HTML表单

表单是用于采集用户输入的数据，并和服务器进行交互的。在html中，使用 `<form>...</form>` 标签来定义一个表单，在form标签体中，通过定义各种 "表单项标签" 来对用户的数据进行采集、并提交到服务器。form标签定义的范围，就代表了采集用户数据的范围，即：在form标签体之外定义的表单项标签所采集的数据是无法被提交的。表单标签的入门程序、基本使用如下：

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>表单标签入门程序</title>
  </head>
  <body>
    <!-- 定义一个表单标签form-->
    <form action="#" method="get">

      <!-- 定义三个表单项标签input-->
      用户名: <input name="username"><br>
      密码:   <input name="password"><br>
      <input type="submit" value="登录">

    </form>
  </body>
</html>
<!-- 在浏览器开发者工具中，可以看到每次点击登录按钮所提交的数据信息-->
```

- (1).action属性：当提交表单时，指定向何处发送表单数据（action=""、action="#"、没有action属性 的作用相同，都是将数据发送到当前页面）
- (2).method属性：指定提交数据的方式（一共有七种提交方式，其中两种比较常用）

get提交方式	post提交方式
a. 请求参数会在地址栏中显示、会封装到请求行中(HTTP协议中讲解) b. 请求参数的大小、长度是有限制的 c. 不太安全	a. 请求参数不会在地址栏中显示、会封装在请求体中(HTTP协议中讲解) b. 请求参数的大小、长度没有限制 c. 较为安全

1.表单项标签

所有的表单项标签都具有id、name属性。id属性是每个表单项标签的唯一标识，不可重复。name属性指定了各个表单项标签的名称，用于对提交到服务器后的表单数据进行标识。并且表单项中的数据想要被提交，必须指定其name属性。

- (1).<input>标签：可以通过type属性，来指定标签的展示样式、功能
 - ①.type="text"：文本输入框，为type属性的默认值。value属性可设置文本输入框的初始内容。
 - ②.type="password"：密码输入框，与文本输入框不同的是，数据输入时会被隐藏显示。value属性可设置密码输入框的初始内容。
 - ③.type="radio"：单选框
 - 要想让多个单选框实现单选的效果，则多个单选框的name属性值必须一样
 - 一般会给每一个单选框提供value属性，指定其被选中后提交的值
 - checked属性：为true代表被选中，为false代表未被选中，可以指定单选框的初始默认值(checked="true"、checked="checked"、checked三种写法均可)
 - ④.type="checkbox"：复选框
 - 一般会给每一个复选框提供value属性，指定其被选中后提交的值
 - checked属性：为true代表被选中，为false代表未被选中，可以指定复选框的初始默认值(checked="true"、checked="checked"、checked三种写法均可)
 - ⑤.type="file"：文件选择框
 - ⑥.type="hidden"：隐藏域，用于提交一些隐藏的信息。value属性可以指定隐藏域的提交值。
 - ⑦.其他常用的type属性：取色器、日期输入框、邮箱输入框、数字输入框
 - ⑧.常用的按钮：按钮不需要name属性
 - type="submit"：提交按钮，可以提交表单。value属性可以指定按钮的文本信息。
 - type="image"：图片提交按钮，也可以用来提交表单。src属性可以指定按钮图片的路径。
 - type="reset"：重置按钮，可以重置表单内容。value属性可以指定按钮的文本信息。
 - type="button"：普通按钮，无实质性功能，后期和JavaScript结合使用。value属性可以指定按钮的文本信息。

- #.注意：
 - ①.placeholder属性：指定输入框的提示信息，当输入框的内容发生变化，会自动清空提示信息。

②.<label>...</label>标签: 指定输入项的文字描述信息。被label标签包裹的文本无任何特殊效果, 但是label标签的for属性一般会与input标签的id属性值对应, 如果对应了, 则点击label区域, 会让input输入框获取焦点。

(2).<select>...</select>标签: 下拉列表, 使用子标签<option>...</option>来指定列表项

- value属性: 指定每个列表项提交上去的值
- selected属性: 设置其初始默认值(selected="selected"、selected两种写法均可)

(3).<textarea>...</textarea>标签: 文本域

- cols属性: 指定列数, 每一行有多少个字符
- rows属性: 默认多少行(文本域的大小不够会自动扩充, rows属性只是默认其初始有多少行)

2. 表单标签的案例: 编写一个用户注册页面, 需要实现的效果见 ["参考\01_用户注册页面的案例效果\(html\).png"](#)

二.CSS

CSS(Cascading Style Sheets): 层叠样式表, 用于页面美化和布局控制。层叠: 多个样式可以作用在同一个html元素上, 并且同时生效。CSS可以将内容展示、样式控制分离开来, 降低程序的耦合度, 让分工协作更加容易, 提高开发效率。CSS的基本语法格式如下:

```
选择器 {
    属性名1: 属性值1;
    属性名2: 属性值2;
    ...
}

div {
    color: red;
    font-size: 10px;
}
```

- 选择器: 用来筛选具有相似特征的元素, 即选择在哪些标签、元素上使用该CSS样式
- 属性: 用来控制标签、元素的样式, 每一对属性需要使用分号隔开, 最后一对属性可以不加分号

1.CSS与HTML的结合方式

- (1).内联样式: 在标签内使用style属性指定css代码, 来控制此标签的样式。即: style属性的内容就是css代码。
- (2).内部样式: 在head标签体内, 定义style标签, 标签体内容就是css代码, 来控制指定标签的样式。
- (3).外部样式: 在head标签体内, 定义link标签, 引入外部的css资源文件。

#.注意:

- ①.外部样式还可以使用以下写法: 在head标签体内, 定义style标签, 导入外部的css资源文件(这种方式不常用)

```
<style>
    @import "../css/test.css";
</style>
```

②.内联样式只能作用于一个标签、内部样式可以作用于整个html文件、外部样式可以作用于所有html文件(在引入资源的情况下)。所以这三种方式, css的作用范围越来越大。其中内部样式、外部样式最常用。

- ③.css的注释方式为: /* 注释内容 */ (在内联样式的style属性中、内部样式的style标签体中, 也必须使用css的注释方式, 因为这些都是css的代码)

2.CSS选择器

(1).基础选择器

- id选择器: 选择具体的id属性值的元素。语法: #id属性值 { }
- 元素选择器: 选择具有相同标签名称的元素。语法: 标签名称 { }
- 类选择器: 选择具有相同的class属性值的元素。语法: .class属性值 { }

#.注意:

- ①.html中的所有标签都具有class属性:

class属性定义了元素的类名。同id属性一样, 可以将其理解为一个标识, 用来标识特定的标签。但是在同一个html页面中, id属性不允许重复, class属性允许重复。并且class属性值可以定义多个, 多个属性值之间使用空格分隔。如: <div class="div1 div2 div3">

- ②.选择器的优先级问题: id选择器 > 类选择器 > 元素选择器

"迪丽热巴"先被id选择器变成红色, 再被元素选择器变成绿色。由于: id选择器优先级高于元素选择器, 所以其显示为红色。

"古力娜扎"先被元素选择器变成绿色, 再被类选择器变成蓝色。由于: 类选择器优先级高于元素选择器, 所以其显示为蓝色。

(2).扩展选择器

- 选择所有元素：语法：* { }
- 交集选择器：选择同时符合这些选择器的元素。语法：选择器1, 选择器2 { }（可推广至多个交集选择器）
- 并集选择器：选择符合其中任一选择器的元素。语法：选择器1, 选择器2 { }（可推广至多个并集选择器）
- 子选择器：筛选选择器1元素下的选择器2元素。语法：选择器1 选择器2 { }（可推广至多个子选择器）
- 父选择器：筛选选择器2的父元素选择器1。语法：选择器1 > 选择器2 { }（可推广至多个父选择器）
- 属性选择器：选择元素名称，属性名=属性值的元素。语法：元素名称[属性名="属性值"] { }
- 伪类选择器：选择一些元素具有的状态。语法：元素:状态 { }

例如：链接元素<a>...在被访问时具有以下状态，我们可以对元素处于不同的访问状态进行不同的样式控制

- link：初始化的状态
- hover：鼠标悬浮在链接上还没点击的状态
- active：鼠标正在点击还没松手的状态
- visited：链接被点击后的状态

#.注意：上述 选择器1、选择器2 为基础选择器(即：id选择器、元素选择器、类选择器)，所以扩展选择器就是多个基础选择器按照一定规则的组合。

3.CSS属性

(1).基本属性

①.文本属性：

- font-size：文本字体大小
- color：文本颜色
- text-align：文本对其方式
- line-height：文本行高
- ...

②.边框属性border：

- border-width：设置边框宽度
- border-style：设置边框样式
- border-color：设置边框颜色
- ...

在实际应用中，可以单独使用上述边框属性，也可以用border属性将上面所有的设置项放在一起。放在一起使用时：各个设置项的顺序没有要求、各个设置项使用空格隔开、有的设置项不写也可以，会使用默认值。

③.背景属性background：

- background-color：设置背景颜色
- background-image：设置背景图片地址
- background-repeat：设置背景图片是否重复平铺
- background-position：设置背景图片的位置
- background-attachment：设置背景图片是固定还是随着页面滚动条滚动
- ...

在实际应用中，可以单独使用上述背景属性，也可以用background属性将上面所有的设置项放在一起。放在一起使用时：各个设置项的顺序没有要求、各个设置项使用空格隔开、有的设置项不写也可以，会使用默认值。

④.尺寸属性：

- width：宽度
- height：高度
- ...

(2).盒子模型

使用以下属性，来控制各个标签在页面中的布局，像在页面中摆放各个盒子(标签)的位置一样。(关于盒子模型，笔记无法说清楚，把视频再看一遍就理解了！)

①.外边距margin属性

margin: 10px 5px 15px 20px;	上外边距是 10px、右边边距是 5px、下外边距是 15px、左外边距是 20px
margin: 10px 5px 15px;	上外边距是 10px、右边边距和左外边距是 5px、下外边距是 15px
margin: 10px 5px;	上外边距和下外边距是 10px、右边边距和左外边距是 5px

margin: 10px;	所有4个外边距都是 10px
---------------	----------------

②.内边距padding属性：默认情况下设置内边距会影响整个盒子的大小，可通过 box-sizing: border-box; 来设置盒子大小不发生变化

padding: 10px 5px 15px 20px;	上内边距是 10px、右内边距是 5px、下内边距是 15px、左内边距是 20px
padding: 10px 5px 15px;	上内边距是 10px、右内边距和左内边距是 5px、下内边距是 15px
padding: 10px 5px;	上内边距和下内边距是 10px、右内边距和左内边距是 5px
padding: 10px;	所有4个内边距都是 10px

③.浮动float属性：left为左浮动，right为右浮动

4.综合案例：编写一个用户注册页面，需要实现的效果见 ["参考\02 用户注册页面的案例效果\(css\).png"](#)