

HTML基础

1、前言

- **(1)、软件产品分类**

当前我们学习的软件开发的课程，所以我们有必要软件产品做一些基础的了解；

首先关于软件产品，我们可以简单分为如下几类：

- **系统软件类：** Windows、Linux、macOS等
- **应用软件类：** QQ、微信、淘宝、饿了么、爱奇艺、B站、学校的教务系统 等
- **游戏软件类：** LOL、王者荣耀、绝地求生、刀塔等

平时我们使用或者接触最多的当属应用软件，当前我们学习的技术也多是应用在这类型的产品的开发上。

- **(2)、客户端与服务器端**

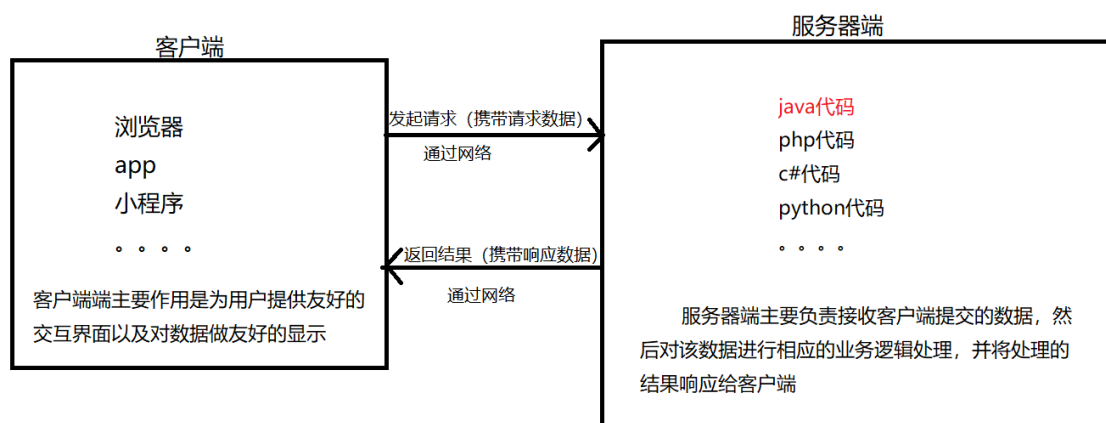
刚才有讲到我们所学的技术多半进行的是应用软件产品领域的研发，那么这里就不得不对应用软件类的产品的结构做一个了解：

通常情况下，我们的应用软件都由两部分组成：

客户端 + 服务器端

客户端： 主要给用户使用， 用户通过客户端来操作软件， 给用户友好的操作界面， 漂亮的数据回显， 在客户端一般不会涉及到复杂的业务逻辑处理。

服务器端： 主要负责业务逻辑的处理， 在服务器端有部署处理业务逻辑的程序（比如java程序）。以登录为例， 当用户在客户端做登录操作时， 客户端会通过网络请求， 将用户输入的数据提交给服务器端， 服务器端拿到数据进行业务逻辑判断， 然后将处理的结果再返回给客户端， 客户端此时再给用户不同的效果显示（登录成功 / 失败）。



- **(3)、B/S与C/S结构**

在上文中我们已经了解了 客户端 + 服务器端这种结构， 在实际的使用过程， 我们又会根据客户端的不同分为 **B/S (Browser/Server) 结构** 和 **C/S (Client/Server) 结构**：

- ①、**C/S (Client/Server) 结构:**

C/S 结构顾名思义又叫做 "**客户端—服务器**" 架构，当然说到这里大家可能会有些疑惑，上面我们不是已经说了 **客户端—服务器** 这种结构吗？

要注意的是在上文中我们说客户端的时候，它是指任何可以给用户操作界面的地方，可以是一个浏览器器，可以是一个APP，也可以是一个小程序；

而C/S 中的 客户端，再用户本地有一个客户端程序，再远程有一个服务端程序。比如qq；当qq点击登录的时候，会把登录信息发到腾讯服务器上。

- ②、**B/S (Browser/Server) 结构**

BS结构顾名思义就是 "**浏览器 — 服务器**" 架构，即用户通过浏览器界面实现对一些应用的操作 。只需要一个浏览器，用户通过不同的网址 (url) ， 客户访问不同的服务器程序。

- ③、**B/S 与 C/S对比**

不管是BS结构 还是CS结构，他们要达成的目的基本一致， 即通过给用户提供一个 友好的操作界面，来让用户使用这些软件产品。

当然他们也有一些区别：

①、BS架构开发时跨平台方便，一个浏览器搞定所有，只要平台支持浏览器，就能实现对该应用的使用，但是对于app来说，安卓的app是不能在苹果系统上使用的，反之亦然；

②、BS架构无需安装程序，只要有浏览器，打开某个网址，就实现了对某个应用软件的使用，但是像桌面程序、app等都是需要安装的；

③、BS架构是单向通信的，消息及时性差，一般情况下我们是点击时，浏览器才会发起请求拿到新数据，服务器一般不会主动给浏览器发送数据，像CS结构，是双向通信，比如qq，你不发消息，也是可以收到信息的。

④、BS不能后台运行，关了就是关了，CS程序可以后台运行

⑤、CS可以将数据保存在本地，当没有网络时，可以支持一些离线操作，但是浏览器没有网基本什么也做不了。

○ ④、总结

其实不管CS还是BS，都是属于 "客户端 + 服务器" 架构，客户端的目的在于为用户提供更加友好的界面，服务器端的目的在于数据业务逻辑的处理。

我们之前的学习的java，就是属于主流的服务端语言， 接下来我们要掌握的就是 客户端 的开发， 当然本阶段课程，我们主要以BS架构开发为主，所以接下来我们需要对基于浏览器相关的技术体系做详细的学习。

- (4) B/S的资源分类：

客户端(浏览器)-----解析引擎 -----
>网址----->服务器(有静态资源 和动态资源)

我现在请求的是动态资源，动态资源会转换成静态资源（**注意**：浏览器只能解析静态资源）；服务器向浏览器转发的是静态资源。

- 静态资源：

- 使用静态页面开发技术发布的资源
- 特点：
 - 所有用户访问，得到的结果是一样的
 - 如文本，图片，音频，视频，html，css，javaScript
 - 如果用户请求的是静态资源，那么服务器会直接将静态资源发送给浏览器。浏览器中内置了静态资源的解析引擎，可以展示静态资源

- 动态资源： ----不同用户登录用户名显示不同

- 使用动态网页技术发布的资源
- 特点：

- 所有用户访问，得到的结果可能是不一样的
- 如：jsp/servlet, php。。。。。
- 如果用户请求的是动态资源，那么服务器会执行动态资源，转换为静态资源，再发送给浏览器

2、HTML基础

2.1、"网页三剑客" 简单了解

想进行网页界面的开发，首先我们需要了解网页的三个重要组成部分：**结构、表现和行为**。他们分别也对应着网页开发中的三种技术。

- **结构：** 说的通俗一些，就是给用户展示的数据摆放在浏览器界面的哪个位置，需依靠**HTML**技术实现；
 - html:用于搭建基础网页，展示页面的内容
- **表现：** 这里我们更习惯叫它**样式**，一个界面如果只是纯黑的文字 和 几张图片，想必它的体验比看报纸的好不到哪儿去，所以不难看出**样式**的效果就是让我们的界面变的更**酷**， 需要依靠**CSS**技术实现；
 - css：用于美化页面，布局页面
- **行为：** 行为指的是你在这个界面做出某个动作后，浏览器会给出相应的 "反应"，当你点击某个按钮时，会弹出来一个界面，当你鼠标悬停在导航栏的时候，会弹出二级菜单，当你回车的时候，会触发表格

数据的提交，等等等等，这些都是网页的行为，需要**JavaScript** 技术实现。

- JavaScript：控制页面的元素，让页面有一些动态的效果

不难看出，我们想编写出一个合格的网页，需要对HTML、CSS、JavaScript这三个技术都要有较好的掌握。

tips: Java 和 JavaScript 的关系就像 周杰 和 周杰伦 的关系一样。

2.2、HTML简介

- 最基础的网页开发的语言
- HTML，全称“Hyper Text Markup Language（超文本标记语言）”，是用来描述网页的一种语言；
- HTML不是一种编程语言，而是一种标记语言（由标签构成的语言 符号：<>）
- 标记语言是一套标记标签，使用标记标签来描述网页
- HTML文档包含了HTML 标签及文本内容

3、HTML语法基础

3.1、标签（元素/容器）的简单理解：

- 一般来说在html中，标签、元素、容器指的都是同一个意思
- **容器**这个词可以帮助我们更容易理解html标签

假设现在有两句话： 我显示在页面左边 ， 我显示在页面右边

想象一下， 你现在在用**text**编写他们， 怎么做，让他们能实现在浏览器上分别居左，居右显示？

空格实现？按多少次空格？

能不能有这样的操作： 我们简单的描述一下，第一句话居左，第二句居右，然后在浏览器中显示的时候，就自动摆好了位置，不用敲那么多空格？

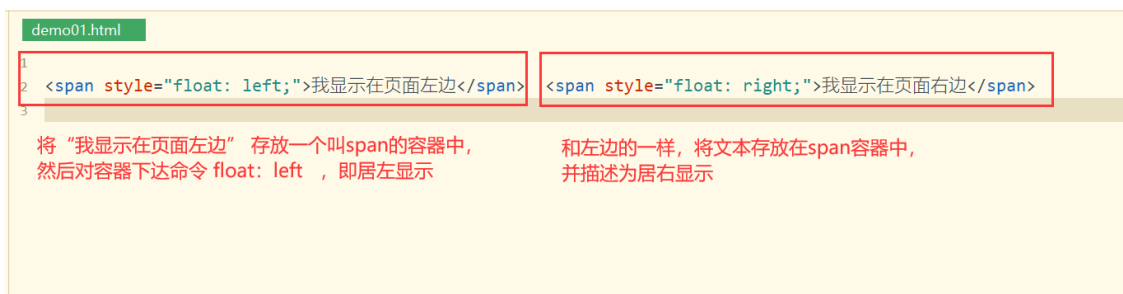
但是显然文字死的，它并不能按照我们的意愿行事，就好比你对河边的泥沙下命令说，你给我到工地上去，它肯定不会动。

但是如果我们将泥沙装车（容器，用来存放泥沙）呢？

我们通过控制车（容器），是不是就实现对泥沙的移动操作？

同样的，文字虽然是死的，那么能不能将文字也装到一个可以被浏览器自由操作的容器中，浏览器操作容器，就等价于操作文字。

也即是说，在浏览器中，想要操作一段数据，我们是它放在指定的容器中的，我们来看一段代码：



接着我们再看浏览器的效果：



基本上看到这里，各位对html中标签的作用应该有一个直观的了解了，说的通俗一些，就是我们将要展示给用户的数据存放到一对标签中（容器中），便于浏览器对这些数据做出操作（排版、调整样式，修改内容等）

3.2、标签（元素/ 容器）的基本语法规则：

- HTML 标签是由尖括号包围的关键词，比如上面的
 - 文本信息在一行展示，行内标签 内联标签
- HTML标签通常是成对出现（少数单个出现）
- 标签对中的第一个标签是开始标签，第二个标签是结束标签， 在他们之间存放数据内容
- 标签与标签之间可以相互嵌套，比如
 - 每一个div占满一整行。块级标签

- 一个html网页中的所有内容（标签 + 数据内容），都存放在一对 标签。这个标签我们称为是包围标签，有开始标签和结束标签。
- 在开始标签中可以定义属性。属性是由键值对构成，值需要用引号（单双都可以）引起来
- 自闭和标签:开始标签和结束标签在一起。如
- html标签不区分大小写，但是建议使用小写

3.3、Html的基本结构：

- 文件标签：构成html最基本的标签
 - html：html文档的根标签
 - head：头标签，用于指定html文档的一些属性，引入外部的资源
 - title：标题标签
 - body：体标签

`<!DOCTYPE html>` `<!-- !DOCTYPE: 文档声明,有助于浏览器中正确显示网页. 在h5中定义该文档是html文档-->`

`<html>` `<!-- 整个网页是从<html>这里开始的, 然后到</html>结束。 -->`

`<head>` `<!-- head标签代表页面的“头”，定义一些特殊内容，这些内容往往都是“不可见内容”（在浏览器不可见），比如标题、编码格式等-->`

`<meta charset="utf-8" />` `<!-- 指定当前页面编码格式-->`

```
<title>第一个网页</title>    <!-- 指定
当前页面标题-->
```

```
</head>
```

```
<body> <!-- body标签代表页面的“身”，定义网页
展示内容，这些内容往往都是“可见内容”（在浏览器可
见），后续课程讲解的标签都是在<body>标签内部的各种
标签。-->
```

```
    hello world!
</body>
</html>
```

3.4、Html中的注释：

```
<!--    -->
```

4、常用标签

4.1、段落相关标签

- 标题标签 h1~ h6
- 段落标签 p
- 换行标签：br
- 区域标签：span
- 水平线标签：hr

4.2、文字格式化标签

- 字体加粗: strong
- 字体倾斜: em

4.3一般标签和自闭合标签

HTML标签分为2种，一种是“一般标签”，另外一种“自闭合标签”。一般标签有开始符号和结束符号，自闭合标签只有开始符号没有结束符号。

一般标签可以在开始符号和结束符号之间插入其他标签或文字。

自闭合标签由于没有结束符号，不会插入其他标签或文字，只能定义自身的属性。

比如: h1~h6、p、span 等大多数标签是一般标签

img、br、hr、input 等少数标签是自闭合标签

4.4、图片标签

- ①、图片标签: img

是空标签，意思是说，它只包含属性，并且没有闭合标签。

<!--

src属性： 要显示图片，我们需要在标签的src属性中指定图片的路径

alt属性： 用来为图像定义一串预备的可替换的文本， 在浏览器无法载入图像时，浏览器将显示这个替代性的文本而不是图像。

-->

```

```

- ②、设置图像的高度与宽度

height（高度） 与 width（宽度） 属性用于设置图像的高度与宽度。

<!-- 按照像素大小指定 -->

```

```

<!-- 按比例缩放图片，通过宽度属性width实现 -->

```

```

注： 指定图像的高度和宽度是一个很好的习惯。如果图像指定了高度宽度，页面加载时就会保留指定的尺寸。如果没有指定图片的大小，加载页面时有可能会破坏HTML页面的整体布局。

- ③、相对路径、绝对路径

在学习 Java 的 IO 流的知识中，加载文件的时候我们会用到相对路径和绝对路径，那么同样的，使用 img 标签去加载图片的时候，也会用到相对和绝对路径。

绝对路径： 完整路径，资源如果在本地，就是从某个盘符根目录开始的路径

资源如果在网络中，则是完整的地址路径

```

```

相对路径： 以某个为路径位置作为参照基准，通过该位置定位其他资源的路径 在 java 普通项目中，是以项目目录所在位置为参照

- 以.开头的路径
 - ./：代表当前目录 ./image/1.jpg 正常 ./是省略的
 - ../：代表上一级目录

```
<!--
```

./ 表示当html前文件所在目录，可以省略不写
如下两个效果等价

```
-->
```

```
<img src = "logo.png"/>
```

```
<img src = "../logo.png"/>
```

```
<!-- ../ 表示当前文件所在目录的上级目录 -->
```

```
<img src = "../logo.png"/>
```

4.5、超链接标签 a

- ①、超链接标签简介

在超链接a标签，我们可以实现从一个页面到另一个页面的跳转

默认情况下，链接将以以下形式出现在浏览器中：

- 一个未访问过的链接显示为蓝色字体并带有下划线；
- 访问过的链接显示为紫色并带有下划线；
- 点击链接时，链接显示为红色并带有下划线。

- ②、a标签中的 href 属性

href 属性描述了链接的目标

```
<a href="www.baidu.com">百度</a>
```

```
<a href="test.html">打开测试页</a>
```

- ③、a标签中的target 属性

target属性主要是用来指定目标窗口的打开方式

target属性值	说明
_self	默认方式，即在当前窗口打开链接
_blank	在一个全新的空白窗口中打开链接
_top	在顶层框架中打开链接
_parent	在当前框架的上一层里打开链接

一般来说我们只需要掌握“_self”和“_blank”这两个属性值就可以了，其他两个基本上用不到。

- ④、锚点

在浏览一些页面的时候，我们经常可以看到这样的按钮： 回到顶部， 到达底部，

点击一下就能跳转到当前页面指定的位置处， 这个就是将 a 标签的锚点功能

锚点的使用： 使用锚点，我们需要借助html标签的id属性， id属性就是给当前标签设置一个唯一标识，然后我们在 href 属性中，指定值为： #id值， 表示定位到这个标签的位置

```
<h3 id="top">我是顶部位置</h3>
```

```
<img src = "img1.png"/>
```

```
</br>
```

```
<img src = "img2.png"/>
```

```
</br>
```

```
<img src = "img3.png"/>
```

```
</br>
```

```
<img src = "img4.png"/>
```

```
</br>
```

```
<a href="#top"> 回到顶部</a>
```

4.6、网页特殊符号

- **网页中的空格（掌握）**

网页空格跟你平常排版的空格是不一样的，你不能在网页代码中按入space键来输入空格，这是无效的做法。在HTML中，空格也是要用代码来实现。

空格： " "

```
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>网页空格</title>
</head>
<body>
    <h3>爱莲说</h3>
    <p>&nbsp; &nbsp;水陆草木之花，可爱者甚
蕃。晋陶渊明独爱菊。自李唐来，世人甚爱牡丹。予独爱
莲之出淤泥而不染，濯清涟而不妖，中通外直，不蔓不
枝，香远益清，亭亭净植，可远观而不可亵玩焉。</p>
    <p>&nbsp; &nbsp;予谓菊，花之隐逸者也；牡
丹，花之富贵者也；莲，花之君子者也。噫！菊之爱，陶
后鲜有闻；莲之爱，同予者何人？牡丹之爱，宜乎众
矣。</p>
</body>
</html>
```

- 其他特殊符号： http://www.lyyestudy.com/les_hj/hj_4.7.aspx

5、标签中的属性

5.1、属性基础简介

属性是 HTML 标签提供的附加信息，比如我们可通过id属性为标签设置唯一标识。

属性总是以名称/值对的形式出现，**比如**：`id="test"`。

属性一般添加在**开始标签**中

5.2、属性的值

属性值应该始终被包括在引号内，如：`href=" www.baidu.com "`；

双引号是最常用的，不过使用单引号也没有问题，如：`href=' www.baidu.com '`；

在个别的情况下，如果属性值本身就含有双引号，那你必须使用单引号，

例如：`name= 'John`

`"ShotGun" Nelson'`

5.3、标签中的常见属性

- **class属性**

指定标签的类名，可以重复

如果你要为两个或者两个以上元素定义相同的样式，建议使用class属性。因为这样可以减少大量的重复代码。

- **id属性**

指定标签的唯一id，不能重复

- **style属性**

指定元素的行内样式

其他通用属性见：<https://www.runoob.com/tags/refresh-standardattributes.html>

6、列表标签

我需要对一些短的文本列点显示的时候，就会使用列表标签

比如大家常见的菜单栏，鼠标悬停时，就会显示二级菜单列表

6.1、有序列表 (ol)

- ①、描述

有序列表中的项目采用数字或英文字母开头，通常各项目间有先后的顺序性默认情况下，有序列表的序号是数字的。

- ②、基本语法

```
<!--ol:有序列表 ol的type可以修改序列展示的图案-
->
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

6.2、无序列表 (ul)

- ①、描述

无序列表中的每个项目使用粗体圆点（典型的小黑圆圈）进行标记

- ②、基本语法

```
<!--ul:无序列表 ul的type可以修改序列展示的图案-
->
<ul>
  <li>Coffee</li>
  <li>Milk</li>
</ul>
```

6.3、定义列表 (dl)

- ①、描述

定义列表，我们可以自定义列表项目，然后对其作出解释

dl 创建一个定义列表

dt 创建定义列表中的项目

dd 对该项目作出解释

- ②、简单语法

```
<dl>
  <dt> html </dt>
  <dd> 在"网络三剑客中"位于结构层</dd>

  <dt> css </dt>
  <dd> 在"网络三剑客中"位于表现层</dd>

  <dt> javascript </dt>
  <dd> 在"网络三剑客中"位于行为层</dd>
</dl>
```

7、表格标签

我们在查询一些数据时（比如本学期的成绩列表），会以表格的形式逐行显示出来，就需要依靠表格标签

7.1、表格的基本结构

- ①、表格基本标签1:

标签	描述
table (表格标签)	table标签用来创建一个表格，表格相关的其他元素都必须在table中
tr (行标签)	表示表中的一行
td (单元格)	表示一行中的每个单元格
th(表头单元格)	表示表头

```

<table border="1" width="300" height="50"
cellpadding="0" cellspacing="0"
bgcolor="azure" align="center">
    <!-- <tr>
        <td>编码</td>
        <td>姓名</td>
        <td>年龄</td>
    </tr> -->
    <tr>
        <th>编码</th>
        <th>姓名</th>
        <th>年龄</th>
    </tr>

    <tr>
        <td>1</td>
        <td>王笑</td>

```

```
        <td>20</td>
    </tr>

    <tr>
        <td>2</td>
        <td>李晓</td>
        <td>31</td>
    </tr>

</table>
```

- ②、表格基本标签2:

标签	描述
table（表格标签）	table标签用来创建一个表格，表格相关的其他元素都必须在table中
thead	表头
tbody	表身
tfoot	表尾
tr	表行
th	表头中的每个单元格
td	身体和尾部单元格

```
<table>
    <caption>表格标题</caption>
```



```
<thead>    <!--表头-->
    <tr>
        <th>编号</th>
        <th>名字</th>
        <th>价格</th>
    </tr>
</thead>
```

```
<tbody>    <!--表身-->
    <tr>
        <td>1</td>
        <td>手机</td>
        <td>2000</td>
    </tr>
    <tr>
        <td>2</td>
        <td>电脑</td>
        <td>3000</td>
    </tr>
</tbody>
```

```
<tfoot><!--表脚-->
    <tr>
        <td>合计</td>
        <td>5000</td>
    </tr>
</tfoot>
</table>
```

7.2、表格边框调整

可以看到，我们现在定义的表格是没有明显分割线来把数据隔开的，可能看起来不那么友好，我们这里简单的设置一下它的样式。

border属性： 设置表格的边框

cellpadding属性： 定义内容和单元格的距离

cellspacing属性： 定义单元格之间的距离，如果指定为0，则单元格的线会合并为一条

```
<!--  
    设置当前表格有边框，为1px粗细， 每个单元格之间  
    没空白  
    整个表格宽400px，高100px  
-->  
<table border="1" cellspacing="0"  
width="400px" height="100px">  
    . . . . .  
</table>
```

7.3、合并行

合并行使用td标签的 **rowspan属性**

语法:

```
<td rowspan="跨度的行数">
```

```
<table>
```

```
    <tr>    <!--第1行-->
        <td>姓名:</td>
        <td>小明</td>
    </tr>
```

```
    <tr> <!--第2行-->
        <!--表示这列算起，向下合并两行 -->
        <td rowspan="2">喜欢水果:</td>
        <td>苹果</td>
    </tr>
```

```
    <tr> <!--第3行-->
        <td>香蕉</td>
    </tr>
```

```
</table>
```

7.4、合并列

合并列则用到td标签的 **colspan**属性

语法:

```
<td **colspan="跨度的列数">
```

```
<table>
  <!--第1行-->
  <tr>
    <td>编号</td>
    <td>名字</td>
    <td>价格</td>
  </tr>

  <!--第2行-->
  <tr>
    <td>1</td>
    <td>苹果</td>
    <td>10</td>
  </tr>

  <!--第3行-->
  <tr>
    <td>2</td>
    <td>梨</td>
    <td>11</td>
  </tr>

  <!--第4行-->
  <tr>
    <!--合并两列-->
    <td colspan="2">总价</td>
    <td>21</td>
  </tr>
</table>
```

8、表单标签

我们在使用一些网站时，经常会看到注册、登录这样的界面，需要用户在相应的文本框中填报数据，然后提交到远端的服务器，这就需要用到表单标签。

8.1、form标签

用于收集用户输入的数据。用于和服务器进行交互

- ①、作用

form叫做表单标签，其他所有的表单元素都需要放在该标签内

用于定义表达，可以定义一个范围，范围代表采集用户数据范围

- ②、form标签常见属性

- **action:** 指定表单数据提交的url地址
- **method:** 提交数据的方式，常见有get 和 post 两种方式
 - get:
 - 请求参数会在的地址栏中显示。会封装在请求行中
 - 请求参数大小是有限制
 - 不太安全
 - post:

- 请求参数不会在的地址栏中显示，会封装在请求体中
- 请求参数大小是没有限制
- 较安全
- **enctype**： 一般不进行设置，当我们提交数据有图片、视频等多媒体文件时，需要将它的值设置为 multipart/form-data
- 表单项中的数据要想被提交，必须指定其name属性

• ③、示例

```
<form action="" method="">
```

表单元素

```
</form>
```

表单项：

8.2、input标签

• ①、作用

input标签规定了用户可以在其中输入数据的输入字段；

比如我们常见的 用户名、密码、电话号码等文本框就是input标签

- ②、value属性

为用户输入的具体数据， 后端服务器需要的就是该数据值， 我们也可通过它提前设置一些默认值

- ③、type 属性

input标签只有一个， 但是通过改变type属性的值， 可以让它呈现不同的控件效果

- **text值：** 创建单行文本输入框， 如用户名文本框

//placeholder: 指定输入框得提示信息；当输入框内容发生变化，会自动清空信息

//label: 指定输入项的文字描述信息

注意：label的for属性一般会 and input的id属性值对应。如果对应了，则点击label区域，会让input的输入框获取焦点

```
<label for="name">用户名:</label>
```

```
<input type="text" id =name placeholder="请输入用户名"/>
```

- **password值：** 创建密码输入文本框

密码: <input type="password" />

- **radio值：** 创建单选框按钮

<!--

注：要想实现单选， 他们的name属性值必须相同

checked 表示该单选框默认是被选中的

如果的没有写value，则默认的选中时on。

有提供value的属性，指定其被选中后的提交值

-->

男:<input type="radio" value="m"
name="gender" checked/>

女:<input type="radio" value="f"
name="gender"/>

- **checkbox值：** 用来表示复选框按钮， 设置input元素为复选控件， 用户可以进行多个选项

福州<input type="checkbox" value="福州"
name="city"/>

厦门<input type="checkbox" value="厦门"
name="city"/>

泉州<input type="checkbox" value="泉州"
name="city"/>

成都<input type="checkbox" value="成都"
name="city"/>

- **hidden值：** 隐藏域， 隐藏域在页面上不显示， 用来存储与传递表单的值， 当用户提交表单时， 隐藏域的内容会一起被提交。

<input type="hidden" value="提交的值"
name="test"/>

- **file值：** 表示文件，当提交图片、视频等数据时，就需要依靠它

```
<input type="file" name="file"/>
```

- **reset值：** 设置重置按钮

```
<input type="reset" value="重置"/>
```

- **submit值：** 提交按钮，点击它，会自动的将数据进行提交

```
<input type="submit" value="提交"/>
```

- **button值：** 设置一个普通按钮

```
<input type="button" value="按钮"/>
```

- **image值：** 将图像定义为提交按钮。，其效果和submit的效果一样，点击可以进行表单数据提交，不过其可以设置src属性值，来修改按钮图片

```
<input type="image" src="test/btn.png" value="提交"/>
```

- ④、HTML5新增的type属性值（了解）

属性值	描述
date	html5. 日期. 用于输入日期(年、月、日，没有时间)的控件
datetime-local	html5. 无时区 日期和时间. 用于输入日期和时间的 控件，没有时区。
time	html5. 无时区 时间. 用于输入 没有时区的 时间值的控件。
month	html5 年月. 用于输入月份和年份的控件，没有时区。
week	用于输入日期的控件，该日期由一个星期编号和一个没有时区的星期编号组成。
number	html5 数字 . 用于输入数字的控件。
range	html5 范围值 不精确数字. 一种用于输入 精确值不重要的数字 的控件。
tel	html5 电话号码 . 输入电话号码的控件。
email	html5. 电子邮件. 用于编辑电子邮件地址的字段。
url	html5. url. 用于输入URL的字段。
search	html5 搜索文本. 用于 输入搜索字符串 的单行文本字段。换行符 将自动从输入值中 删除。

8.3、textarea标签

textarea 标签用来定义一个多行的文本输入控件， 文本区域中可容纳无限数量的文本， 其中的文本的默认字体是等宽字体

```
<!-- cols: 指定列数， 每一行有多少个字符 rows 默认多少行 -->  
    <textarea cols="20" rows="5"  
name="desc"></textarea>
```

8.4、select 和 option 标签

select元素用来创建下拉列表， option定义了列表中的可用选项。

selected属性： 表示选中的那个条目（option）， 可以在某个option 设置该值， 表示默认选中

option标签中的value值， 表示要提交给后台的数据

```
<select name="province">
    <option value="">--请选择--
</option>
    <option value="1">北京</option>
    <option value="2">上海</option>
    <!-- selected默认选项 -->
    <option value="3"
selected="selected">福建</option>
</select>
```

9、多媒体标签

- 插入音频、视频和flash

在网页中插入音频、视频和flash都是使用embed标签。

```
<embed src="多媒体文件地址" width="播放界面宽" height="播放界面高"></embed>
<!-- 视频标签 -->
    <embed
src="https://haokan.baidu.com/v?
pd=wisenatural&vid=14143452875728353280"
width="300" height="500"></embed>
```

10、块级元素与行内元素

在html中，元素主要分为**行内元素**和**块级元素**，

在接下来的课程中，我们需要学会对页面进行排版布局，所以我们必须对这两种元素的特点加以掌握。

- ①、**块级元素与行内元素区别：**

行内元素指的是书写完成后，可以同其他元素同行显示，并且元素不能设置宽和高；

块级元素写完后，其会独占一行，不能与其他元素共存，并且可以设置宽高；

- ②、**常见的行内元素和块级元素：**

行内元素：a b br img input select span
strong texttarea

块级元素：h1~h6 li ul p table div from hr

注：由其要注意 div 和span这两个元素，在布局中经常会被使用到

11、浮动框架iframe

iframe标签：浮动框架标签

浮动框架是一种较为特殊的框架，它是在浏览器窗口中嵌套的子窗口，整个页面并不一定是框架页面，但要包含一个框架窗口。iframe框架可以完全由设计者定义宽度和高度，并且可以放置在一个网页的任何位置，这极大地扩展了框架页面的应用范围。

- ①、基本语法：

```
<!--
```

1、src属性是iframe的必须属性，它定义浮动框架页面的源文件地址。

width和height这2个都是可选属性。

```
-->
```

```
<iframe src="浮动框架的源文件" width="浮动框架的宽" height="浮动框架的高"></iframe>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>测试浮动框架iframe</title>
```

```
</head>
```

```
<body>
```

```
    <div id="main">
```

```
        <h3>我是外部父页面</h3>
```

```
        <iframe src="表格.html"
```

```
width="400px" height="300px">
```

```
</iframe>
```

```
    </div>
```

```
</body>
```

```
</html>
```

- ②、设置浮动框架是否显示滚动条scrolling

对于浮动框架iframe的滚动条，我们可以使用scrolling属性来控制。scrolling属性有3种情况：根据需要显示、总是显示和不显示。

scrolling 属性值	说明
auto	默认值，整个表格在浏览器页面中左对齐
yes	总是显示滚动条，即使页面内容不足以撑满框架范围，滚动条的位置也预留
no	在任何情况下都不显示滚动条