# HTML

## 一、二阶段和一阶段有什么区别

一阶段主要是CoreJava的学习，也就是纯的Java内容

二阶段主要学习的是J2EE的内容，基于Java的网络编程

在二阶段中会引入网络编程必须的基础知识html、css、js等内容

## 二、什么是HTML

Hyper Text Markup Language

HTML 指的是超文本标记语言（标签语言）

tips：这就是Java之外的又一种语言，Java是高级编程语言，HTML是标记语言，是一个全新语种。

一个标记语言的小例子：

<note>

<to>George</to>

<from>John</from>

<heading>Reminder</heading>

<body>Don't forget the meeting!</body>

</note>

在标记语言中，核心结构是

<标签名 属性名=属性值 ……>内容（文字、子标签）</标签名>

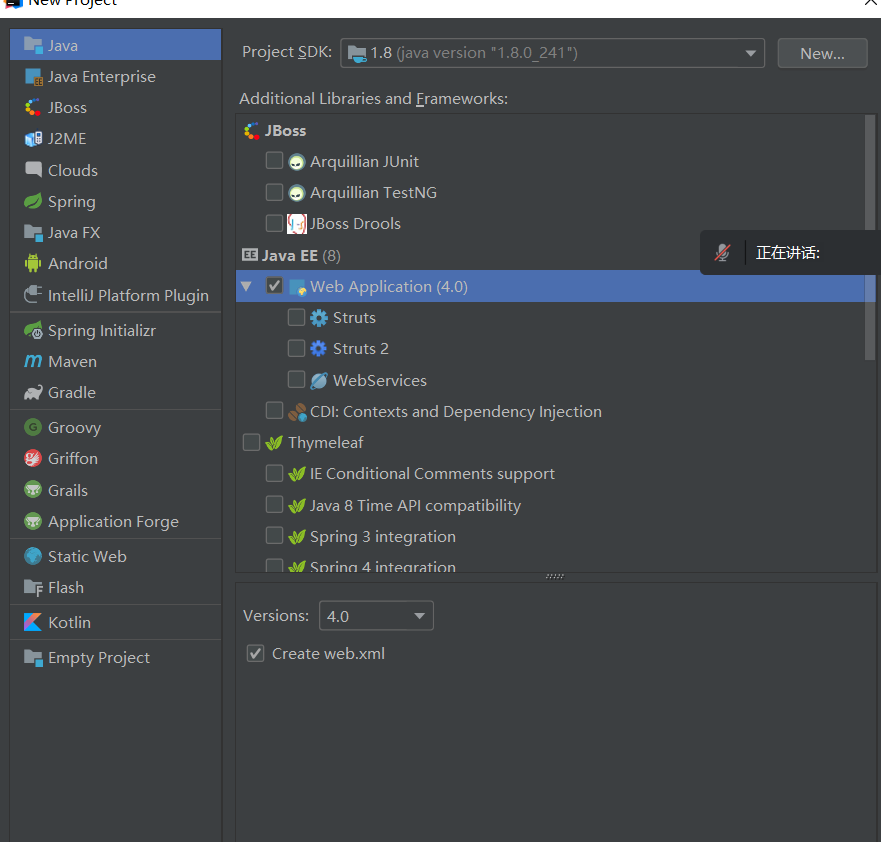
一个完整的标签一般由三部分组成1、标签名 2、内容 3、属性

HTML是针对网页实现的XML的标签库，在HTML的规则下有一系列的预设标签，通过编写这些标签可以组成一个完整的网页。

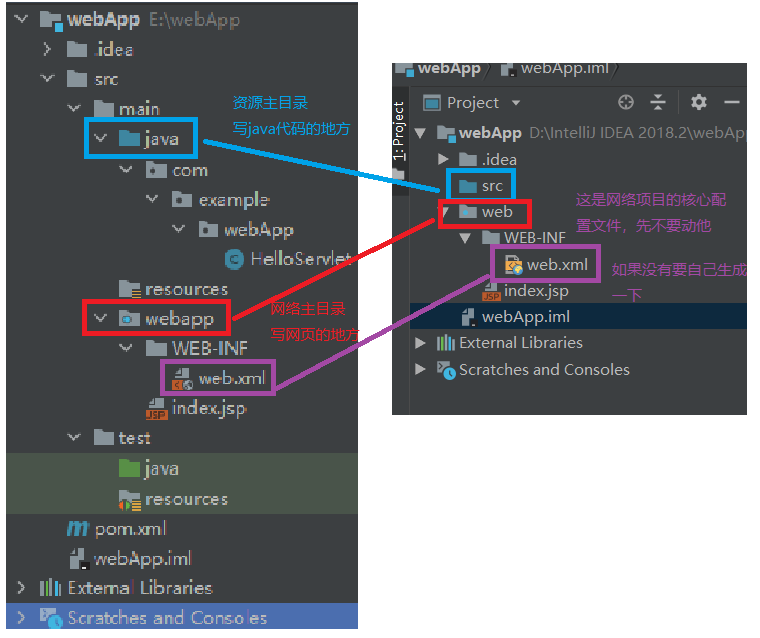
## 三、HTML入门

1、第一个网页

这里不再使用一阶段的项目，尝试新建一个项目，这个项目必须是web application网络项目。



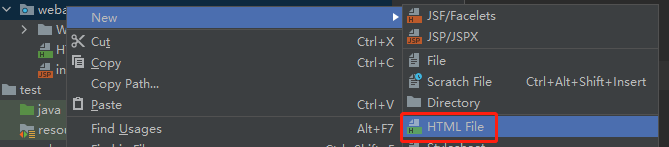
由于IDEA的版本不同， 生成的时候有的同学是基于Maven结构的项目（左边），有的是单纯webApp项目（右边），代码的结构有所不同，但是使用上是没什么区别的。都有类似的文件夹，只要把代码放到对应的文件夹中即可。



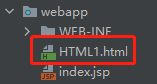
手动生成web.xml的方式



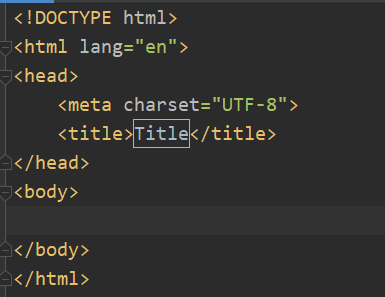
在webapp文件夹下新建一个 .html 文件



在弹框中输入文件名，回车即可



双击打开html文件

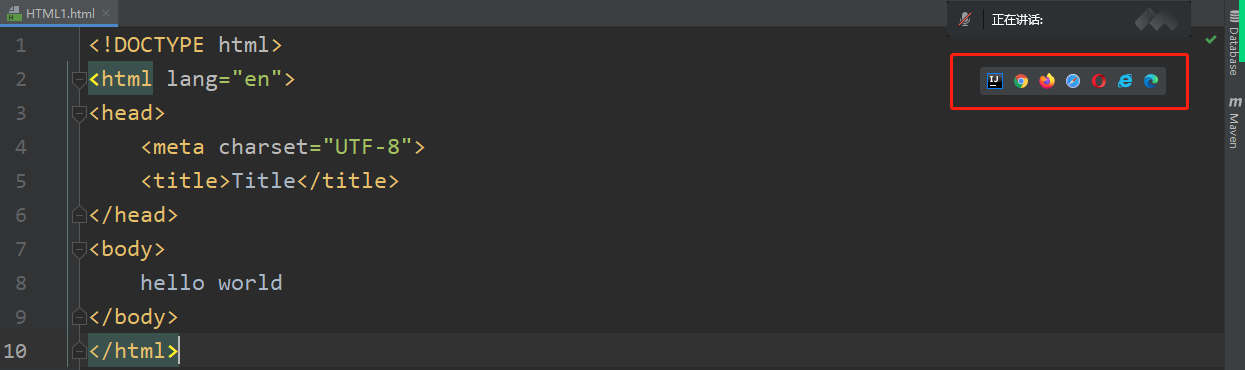


这就是最基础的html代码结构

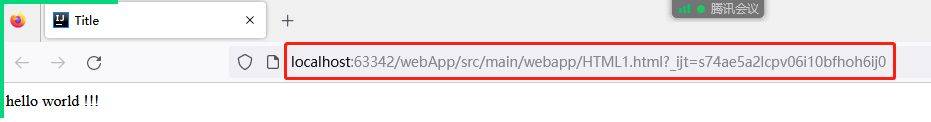


接下来要学习的，就是在基本结构上添加其他html库中标签。

尝试在页面上输出一个helloworld



在页面的右侧，当你输入完毕后会弹出一个有各种浏览器标签的浮框，点选指定浏览器即可。



在浏览器的地址栏中可以看到一个网络路径（不是指向磁盘上的文件的路径）

localhost 本地主机 63342 端口号 ……

这里其实是IDEA提供了一个测试用的服务器，它把你的页面部署到了这个服务器上，供你使用网络地址进行访问。和直接访问区别不大，能看到页面就可以。

你已经学会如何在IDEA中编辑和打开一个网页了。

如何使用浏览器的开发者模式，点击键盘上的F12按钮，就可以开启大部分浏览器的开发者工具

以火狐为例



不同厂商的浏览器的开发者工具略有不同，但是大致是相似的，可以找到功能相似的选项卡。不存在技术隔离的问题，会一个就全会了。

## 四、HTML标签

1、标题标签

<hn> n是1~6之间的数字

<h1>标签是最高级别的标题标签，会使文字加大加粗黑体

后续的标签级别逐渐降低，h1~h3都是变大，h4就和默认字体一样大，h5~h6是缩小

tips：在尝试编写过h1~6之后，尝试写入h7这个标签库中不存在的标签，发现在页面上虽然看不到h7，但是它标签中的内容就和直接写的字符串是一样的，没有任何变化。

不能使用html标签库之外的标签，他们不会生效，也没有任何效果。

2、修饰标签

b 标签

文字加粗bold，将字符串写在b标签之间就能实现加粗效果

<b>被加粗的内容</b>

i 标签

文字斜体italic，将字符串写在i标签之间就能实现斜体效果

<i>被斜体的内容</i>

tips：字形的改变

展示给用户看的文字，实际上是一张图片，他们保存在一个字形库中。

所谓的加粗、斜体、默认字形，就是同一个字的不同的图片展示。斜体和加粗是否生效要看当前系统中字形库是否支持这些特殊字形。

tips：标签的嵌套

能否让同一个字符串，同时加粗并斜体

<b><i>hello</i></b>

标签是存在嵌套关系的，且部分效果是可以叠加生效的

<i><b>world</b></i>

针对修饰标签，里外关系无所谓的，只要嵌套成功即可

tips：错误的嵌套

标签的开闭标签要成对出现，像括号一样出现在被修饰内容的两侧

<b><i>hello</b></i>

不能这样写，这不是成对出现，标签的嵌套结构是中心对称的，先出现的要最后 关闭。

但是如果你强行写的话，会发现页面上出现的是正确的代码，是因为HTML的代码就较强的自纠性，浏览器在读取标签的时候会尽可能的修复错误的代码，按照正确的格式展示。

3、字体标签

刚刚的样式修改，每个标签只能修改文字的一个样式，在这里介绍一个新的标签，font，字体的总的设置标签，可以用这一个标签修改多个字体的属性。

<font color="red" face="MV boli" size="1">world</font>

font标签有三个属性可以设置

color：文字的颜色

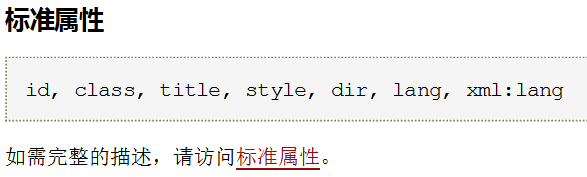
face：文字的字形

size：文字的大小（1~7）默认3

tips：标签中的属性

属性分为两类

1、共有属性，几乎所有标签上都存在这些属性



2、当前标签的独有属性，每个标签都会或多或少的有一些自己的独有属性，这是需要熟能生巧的（太多了，用到多少记多少）



属性值的内容

1、完全自定义的内容

face：在输入字形的时候，只要你的计算机支持你可以在属性值中写入任何字形的名称。这种值在标签定义中是没有约束的，想怎么写就怎么写。

2、给出具体的枚举值的内容

size：font预定size的值在1~7之间，就意味着你只能输入1234567这几个数字，输入其他内容都是无效的

3、可以使用多种值的内容

color：颜色有多种描述方式

1、颜色名：red、blue、yellow

2、色号：#rrggbb 使用两位十六进制的数字描述红绿蓝中的一位

#ff0000 #0000ff #ffff00 使用红绿蓝为基准色调出其他的颜色

3、函数：rgb(red, blue, yellow); 函数的入参是0~255十进制数字，分别描述红绿蓝三种颜色

4、单标签

不是所有的标签都是成对出现的，有些标签只有开标签没有闭标签。

1、换行<br />

在开标签的名字后面输入 / 表示当前标签自关闭（没有闭标签）

tips：关于html中的空格、回车、制表符

在html中，空格、回车、制表符连续出现的时候最多表达为一个空格。

所以换行必须要使用<br/>换行标签实现。

2、分割线<hr />

从页面最左边到最右边画一条分割线（也有换行的功能）

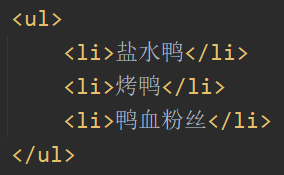
5、列表标签

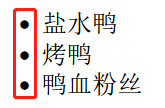
1、无序列表 ul

这是第一个见到的组合标签，ul是作为父标签出现的，它本身没有什么样式

靠的是内部的子标签 li 每一个li标签表示列表中的一项。

tips：序表示的是序号，无序列表就是没有递增的序号用一些图标表示一项内容。





tips：在html中存在大量的组合标签，单个标签不能实现一些复杂页面结构，需要使用多种标签复合使用才能生效，例如这里的ul + li

tips：能否修改无序列表前的图标？

是可以的在父标签ul上提供了type属性，是由枚举值的属性

disc 实心黑点

square 实心黑方块

circle 空心圆圈

如果要使用无序列表，但不希望出现图标如何设置？

在属性值中输入空值none

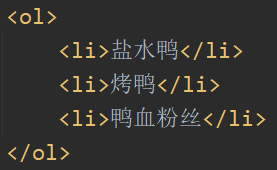
<ul type=”none”>

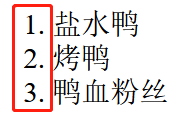
tips：在网页上，空值新增了一个关键字 none

2、有序列表 ol

也是组合标签，也需要使用li表示列表中的一项

所谓的有序列表就是在每一项的前面添加一个递增的序号





tips：有序列表的序号的类型也可以通过type进行修改

属性值也是枚举值

A 大写字母的序号 A~Z

a 小写字母的序号 a~z

I 大写罗马数字 I II III IV V VI VII X…

i 小写罗马数字 i ii iii iv v vi vii x…

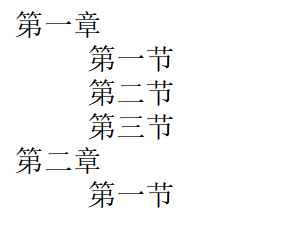
1 数字 1~n

tips：li这种标签，在不同的父标签下有不同的表现形式，这里就不是一个li的问题是多个标签组合的问题，无序列表 = ul + li; 有序列表 = ol + li; 这是多个整体，不讨论单个标签的意义。

3、说明列表 dl

在说明列表中，不使用li，转为dt说明标题 dd 说明内容 这两个标签





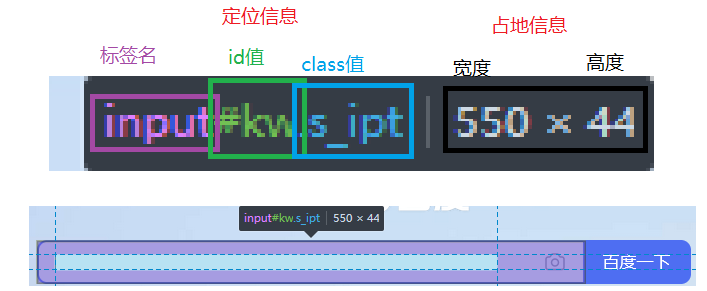
dt总是顶头显示，dd总是缩进显示

家庭作业：

1、请尝试使用F12查看百度搜索的主页，请找出



可使用浏览器的点选工具查看指定组件的基本信息



红框中的输入框是用什么标签实现的？

2、预习table、div、span、input、form标签

table表格（tr行 td行中的一格 thead tbody tfoot）

div 块级框架标签

span 行级框架标签

input 表单组件标签

form 表单标签

3、请自行预习CSS和JS相关内容，并尝试百度预习Jquery。

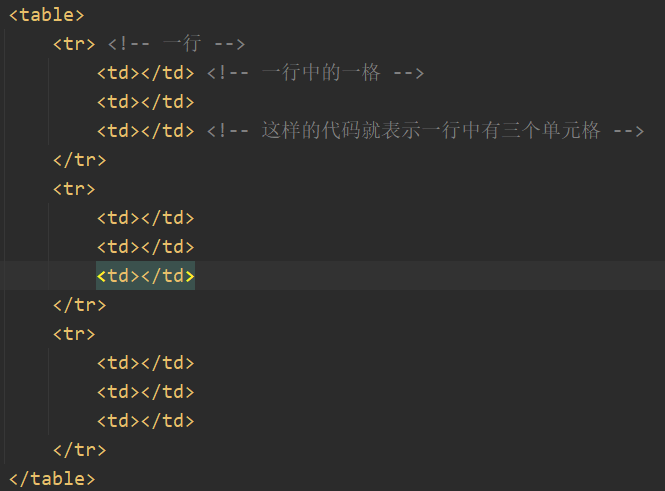
6、表格标签

table

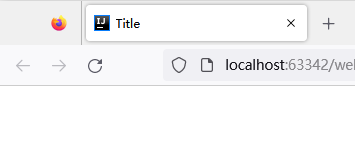
table也是一个组合标签，它自己是没有显示的，要使用子标签搭建一个表格的样式，tr表示表格中的一行，每个tr标签表示一行内容，td表示一行中的一个单元格，每个td标签表示一个单元格。

如果想在页面上实现一个3x3的表格，应该如何编码？

tips：table是第一个见到的没有初始大小和样式的标签了，当你编写了

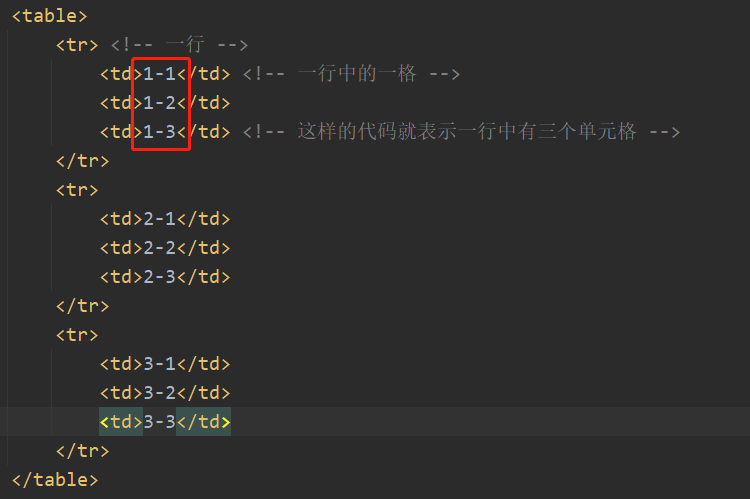


这么多代码之后，发现页面上没有任何可见的显示

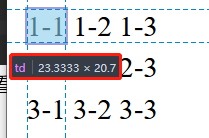


这是因为表格大小是根据内部的内容“撑”出来的，也就说它没有初始大小，内容有多大它就被撑到多大。

可以尝试向表格中写入一些文字信息作为内容



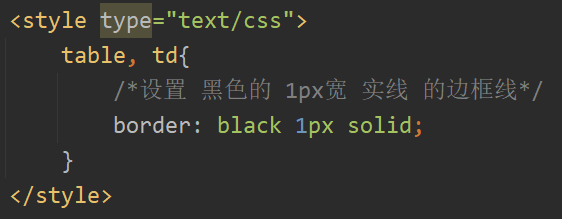
此时刷新页面，并用点选工具查看td的样式



在背景中看到 n-m 的文字显示出来了，并按照设置的行列展示在页面上

td有了大小 23x20 这个大小就是靠文字内容撑出来的。

table是不自带边框线的，只是内容按照表格的布局进行放置。如果要按照常见的表格添加边框线的话就需要用到一些CSS的知识了。



thead

tbody 这两个是后期用的比较多的，一般用在分页查询上

tfoot

上面的三个标签对表格的显示没有任何影响，存在的目的是给tr进行分组。

例如在后续编程中最重要的一个页面效果“分页”就需要使用标签进行实现

在页面上展示表格中的内容的时候，最少也要分两部分展示

1、表头，显示每一列的列名

2、表体，显示数据

在分页的时候，如果要按照页码进行翻页，需要刷新哪些内容？

表头部分不管是表中的哪一页都是完全相同的，所以不需要被刷新的部分

表体中的数据要根据不同的页展示不同的信息，是要被刷新的部分

虽然都是表中的行，它的含义和处理方式都是不一样的

thead仅仅是将一些tr标记了表头行，但是不会真的按照标题展示列名，如果要修改样式的话，需要引入一个新的标签

th 表头标签

它可以替代td写到一行中，表示这个格子是表头的单元格

写入th中的文字信息会 加粗 居中显示

7、框架标签

div，span

直接写div和span不会对页面产生任何影响，他们存在的意义是搭建页面的基本结构，其他的标签就可以在结构上完善页面效果了。

div和span一般是结合CSS和JS使用搭建页面的基本框架（实现页面效果的），一般来说都不会在页面上直接显示div或者span本身，他们更接近于容器，展示的是他们内容的其他的标签。

tips：块级标签、行级标签

html中的标签分为两类

1、块级标签

块级标签总是独占一行，在它的前面和后面都不能有同行的元素，必须要独占一整行。

常见的块级标签有哪些？

div br hr ul ol li table h1~6

2、行级标签

行级标签可以在同一行中。

span b i font

8、表单元素

表单：就是登录、注册、填写个人信息时，页面上的输入框、单选框之类的收集用户信息的标签。存在的目的是收集用户输入的信息并提交到指定的地址上。

1、表单标签 form

tips: 请仔细看一下这个单词是怎么拼的，千万不要写出from

<form method="" action="" onsubmit="" enctype="">

form中有很多属性，常有一般是这四个

1、method：表单的提交方式

form主要的功能就是将用户输入的信息提交到指定地址上，这个指定地址一般就是服务器，例如你在电商网站上进行登录操作，form表单就要将你的账号和密码发送到对应的服务器去验证是否登录成功。这是二阶段后期要学习请求发送的一个内容。

提交方式主要分为两种 1、post打包发送，将用户输入的数据打成一个数据包，在后台进行发送，用户不能在前台看到发送的任何数据 2、get 随url进行发送，在当前url的后面用 ? 开始传值 值传递的形式是 属性名 = 属性值 & 属性名 = 属性值 …用户可以直观的在url上读到你传递的数据，这样的方式不太安全。

2、action：表达发送的目标地址

设置将用户输入的数据发送到哪里去，一般写的是你自己的服务器上接受数据的服务的地址。

3、onsubmit：在提交表单时执行的回调函数

tips：在后续的编程中这种 onXXX的属性是常用的属性，表示当XXX事件发生的时候。onclick 当鼠标点击的时候 onblur 当失去焦点 onload 当页面加载完成 onchange当表单中的值发生变化的时候。它会设置一个指定的事件进行监听，当这个事件发生的时候就触发一段JS代码执行。这一般搭配JS来使用。

4、enctype：设置发送的数据类型

在后续的学习中只有一个地方要设置这个内容，文件上传

在表单中填写的信息说到底都是字符串，他们的类型是单一的，如果要传递一些复杂类型，例如这里举例的文件上传，至少不该是简单的字符串，就需要手动设置这里的数据传输的类型，调整成合适的对应的类型。

2、表单组件 input

input几乎涵盖了所有的表单组件



input标签中存在type这个属性，根据属性值的不同，input会展示出不同的表单形式。

2.1、单行文本框

input的type属性默认值就是text所以在页面上直接写一个input就会展示一个单行的文本框

<input type=”text”>



大部分的表单组件都不是html自带的，而是引用当前系统中的组件。虽然都是单行文本框，在不同操作系统上，展示出的样式是不一样的（随当前系统而定的）

html5 是基于html 4.01版本的大更新带来了一系列的新标签和新属性，这里针对输入框就是有一个

<input placeholder="请输入用户名">

这是一个占位字符的设置，当用户没有在输入框中输入任何内容的时候，就会展示这个占位文字信息。主要用户输入了内容，不管输入多少信息，占位内容都会消失。

在查看w3school的时候可以根据属性后有没有红色的5的图标判断属性是新的还是旧的



这里再讲一个特殊的内容，浏览器支持



不是所有的标签、属性、效果在任何浏览器上都可以展示，有的浏览器是不支持部分标签、属性、效果的。在这些浏览器上代码再怎么改也不会显示不支持的效果的。

2.2、密码框

密码框的样式和单行文本框是一样的，特点是做了参数的遮盖

<input type=”password”>



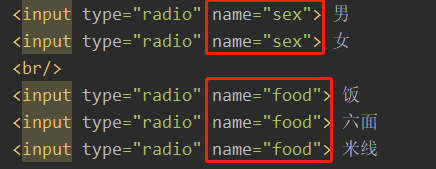
将单行文本框和密码框并排放，发现样式是一样的，区别只是文本框能直接读到输入的信息，密码框就没有了，替换为实心黑点（替代值也是随系统变化的）

tips：这俩框能放在一行，所以input是行级标签。

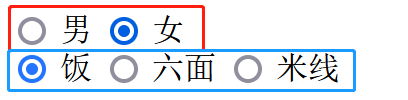
2.3、单选框

单选：只能选一个，默认单选框就应该是多个一组出现的。

单选是组内单选，要先将多个单选框编为一组，才能实现组内的选中状态互斥。

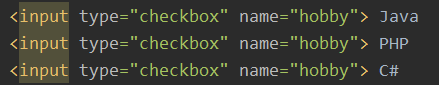


使用name属性对单选按钮进行分组，name属性相同的在同一组中，他们是组内单选的



在同一组中，最多有一个选项被选中，但是不同组之间是互不影响。

2.4、多选框



多选框也有name属性，且也根据name属性进行分组，但由于是多选框不存在组内互斥这种现象，不能直观的从选中状态分组。

tips: 结合单选框和多选框发现input只表示前面的选择框（不管是单选还是多选），实际用户看到的选项是都是和标签没有任何关系的一个写的比较近的字符串。选择框本质上不会提供备选项给用户查看都是程序员另外配置的。

2.5、文件选择器



要注意一个问题，文件上传是比较复杂的一个功能不是靠一个input就实现的，如果页面上只有一个input你最多上传一个文件的名字，文件的内容是无法上传的。需要在后台使用web相关的知识传递文件的内容，并手写IO流才能实现文件上传。

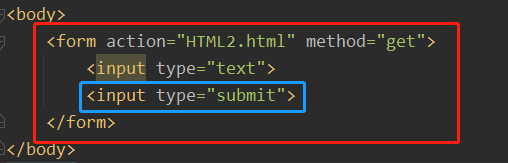
所谓的文件上传就是将用户电脑上的一个文件通过网络复制到服务器上存储，本质是远程的文件复制。需要多种技术的整合使用才能实现，input中的文件选择器只是其中的一环。

tips：html完全没有提供文件选择器的实现，都是依赖当前系统中的文件选择功能实现的，这也是样式变化最大的组件，完全依靠网页所在的操作系统提供选择的功能。

2.6、表单相关组件

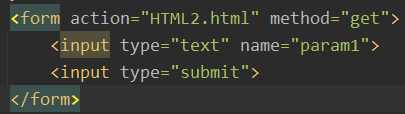
重置按钮、提交按钮

之前介绍的input的分支都是输入的组件，他们的目的收集用户输入（选择）的内容。这里介绍的就不是在收集信息，而是在操作form表单。



以提交按钮为例，当用户点击这个按钮的时候，它会把自己所在的form中的所有表单的数据按照method设置的方式发送到action设置的地址上。

tips：所有要发送数据的组件必须有name这个属性，因为发送的数据的逻辑和map相似，name属性就相当于key，用户输入的内容是对应的value。是以key=value的形式发送的，如果你没有设置name属性，它就不知道这个值是依附于哪一个属性上的，就发送不出去。



这里第一个输入框中的name就是param1，发送的数据就会绑定在这个参数名上发送出去。



在目标页面上可以看到最后出现了action中指定的目录，和input发送的数据。

由于在url中直接看到传递的内容，这一定是get方式发送的数据。

在第二个页面是接受不到这些数据的，表单提交的数据在后续的学习中是要用Java代码去获取的，在一个网页上是拿不到的。

现在借助get方法传递，在url中尝试查看之前学过的表单组件发送的数据是怎样的，以及不同的表单组件分别是怎么传值的。

1、单行文本框

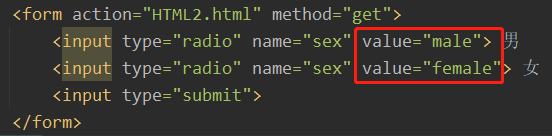
用户在页面上输入的文字信息

2、密码框

虽然在页面上密码框会遮盖输入的内容，但传递的内容是原文字

3、单选按钮

输入框中内容是用户自定义的，选择框（单选、多选、下拉菜单）中的内容都是程序员预设好了，让用户多选一的。



标签后面的男女这个文字只是页面上的独立的字符串和标签没有任何关系，所以要在标签内部用value属性为这个选项赋值，当用户选择这个选项的时候就等于他选择这个值（这个选项表示的值）。

4、多选按钮

赋值方式和单选按钮一样，但是区别在接受方

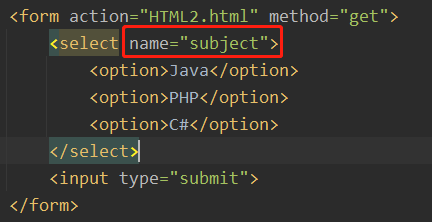


如果在同一个属性名下传递多个值，可以用&连接多组传递（以后在Java代码中是使用数组接受这多个值）

5、下拉菜单

input虽然覆盖了大部分的表单元素，但没有覆盖全部，有部分表单组件需要使用其他的标签来实现，下拉菜单就是其中一个

下拉菜单是组合标签，父标签是select，每一个选项是一个option



下拉菜单的name属性要写在select标签上，虽然有很多个选项，但是最终能被选中的只有一个，所以这里还是一个单选的结构，只返回一个值就不可能在每一个option上都设置name属性，只可能是写在父标签上作为统一的属性名。

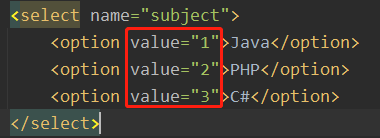
在测试中发现传递目标页面上的值就是option双标签之间的文字内容。

option有两种赋值方式

1、不写任何属性，双标签间的内容就是value

2、给每一个option写value属性，标签之间的内容只是给用户看的，实际提交的是value的值。

这里一般会用第二种方式，以之前数据库模型为例，要用户选择一个员工或者一个学生的时候，给用户看的是人的名字，传递到后台的是人的id。



6、多行文本框

<input type=”text”> 单行的文本框，只能输入一行文字信息（不接受换行符）。如果要在页面上提供编辑大量文本的地方（简介、个人信息、说明…）就需要用到多行文本框

<textarea></textarea>

由于area可以翻译成区域，也称为文本域



textarea是一组双标签，写在标签之间的内容是它的初始值

tips：由于这个初始值的问题，写textarea标签的时候请一定要开闭标签紧紧相邻，不然回车空格制表符会污染希望出现的内容。

7、隐藏域

需要在页面上记录一些数据，又不希望用户能直接在页面上看到，就可以放入隐藏域中，如果所在表单提交了，保存在隐藏域中的值也会一起被提交。



隐藏域中的值，一般不会由用户自定义（否则为什么要隐藏呢），值设置好也一般不会去修改。

常用场景有：修改页面上隐藏用户id或者其他的锚定信息

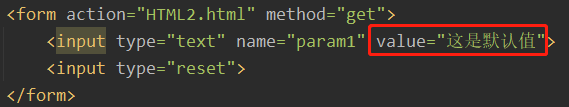
重置按钮

初始化：变量的首次赋值

清空：组件内没有输入的值

重置：恢复到起始状态

以单行文本框为例介绍一下什么叫恢复到起始状态。

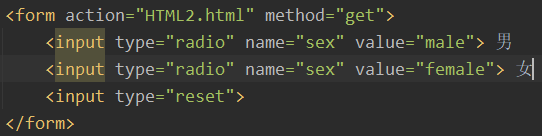


重置按钮并不是单纯清空输入框，而是根据进入页面的时候设置的默认值，恢复到默认值这个状态。

各个组件的默认值设置

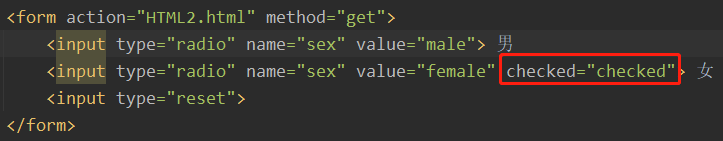
输入框之类的基本上都是value属性，属性值就是默认值

1、单选按钮



单选按钮的获取的内容实际上是选择框的选中状态

初始值并不是每个选项表示的值，而是你选了哪一个，换种说法哪个选项被选中了



设置初始值的方法是在初始选项上写入checked = “checked”表示被选中了。

tips：问如果两个选项都checked了会怎么样？

只会有最后一个被选中，因为组内互斥的，在单选中选中一个操作有两步

1、选中自己

2、取消选中其他人

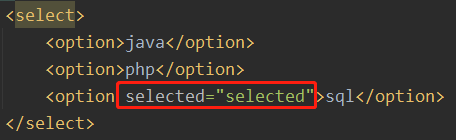
当你两个都checked的时候，第一个是选中了，在第二个也被选中的时候，第一个选中状态被取消了。

2、多选按钮

同上

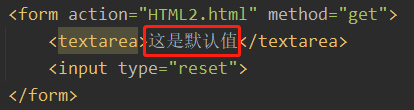
3、下拉菜单

选中状态要设置在一个选项上



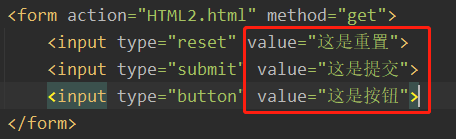
这里关键字变为了selected

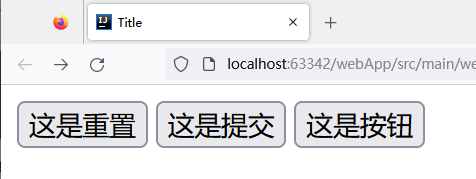
4、多行文本框



textarea标签没有value这个属性，写在两个标签之间的内容就是初始值

tips：value是一个多用途属性，在按钮上表示的是按钮文字信息





tips：最后出现的type=button的按钮在后续的编程中是最常用的按钮，在事件中有一个onclick事件，可以在JS部分任意编辑触发后的内容，自由度更高，也更好用。

tips：按钮在网页上一般不用真实按钮，上文提到onclick事件在几乎所有html标签都有，谁都可以是一个按钮（触发点击事件），给div或者图片或者一段文字都可以作为变种的按钮出现。

a 超链接

给出一个地址，将页面跳转到目标地址指向的资源上

tips：标签中的属性，有一些是必须出现的属性，如果不出现标签是不生效的。

例如a标签中的地址属性

tips：a标签作为超链接标签，存在的目的就是指向一个新的地址并跳转过去，你都不给出地址，它的预设的逻辑怎么生效呢？地址作为必须属性是合理的。



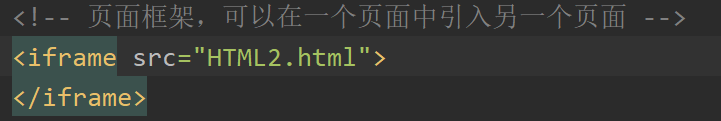
至少写成这样，超链接才是有效的

除了href属性外，a标签还有一个关键的属性target目标属性，是用来定义在哪里打开目标页面

属性值是枚举

1、\_self 在当前页面中打开，新页面会覆盖原页面

2、\_blank 在另一个新页面中打开，原页面不会收到影响



在html中有一个现在已经很少用的标签iframe标签，它支持在一个页面中内嵌另一个页面

3、\_parent 在页面彼此嵌套的关系下，在上一层页面打开目标页面

4、\_top 不管嵌套多少层，总在最外侧打开

5、framename 指定在一个内嵌iframe框架中打开页面



将target写成iframe的name属性的值，目标页面就会在指定iframe中展示

可以将链接做成菜单，这样就可以在不跳转页面的情况下，在同一个页面中切换目标页面显示了。