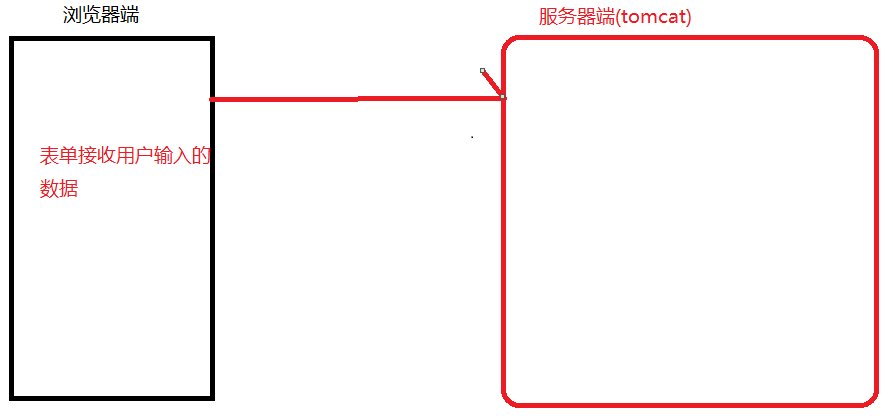
# Day02笔记HTML表单+CSS

# 1、表单（重点）

## 1.1、表单的作用

用于接收用户输入的数据，将这些数据提交给服务器，从而实现浏览器和服务器的交互



表单标签：<form action=”” method=””> </form>

## 1.2、HTML-表单输入项标签<input/>

* **<input>** 标签用于获得用户输入信息，type属性值不同，会成为不同的组件。最常用的标签。

input属性：

|  |  |
| --- | --- |
| 属性名 | 属性说明 |
| **type** | 定义表单输入项input的组件类型  详见下一个表格*type属性值说明* |
| **name** | 组件名。如果需要将你输入的数据提交到服务器，那么必须设置name属性值    参数名：name属性的值  参数值：用户输入的 |
| **value** | 组件默认值。 参数值  用户名、密码、手机号.... 咱们可以通过输入框手动输入，value值就是咱们输入的值  单选框/复选框：value值只能选不能输入，所以必须给其设置一个value |
| **checked** | 设置单选框/复选框的默认选中状态  默认是不选中状态  Checked = “checked” |
| readonly | 设置该标签的参数值只读，用户无法手动更改。  **数据是可以正常提交** |
| disabled | 设置该标签不可用（禁用），参数值无法更改，且**参数值也无法提交** |
| size | 组件的长度 |

input-type属性值说明：

|  |  |
| --- | --- |
| 属性值 | 说明 |
| **text** | 文本框，单行的输入字段，用户可在其中输入纯文本。  默认值 |
| **password** | 密码框。 内容非明文 |
| **radio** | 单选框。  必须再设置name属性，并且name属性的值必须相同 |
| **checkbox** | 复选框。  必须再设置name属性，并且name属性的值必须相同 |
| **submit** | 提交按钮。用于控制表单**提交数据**。 |
| **reset** | 重置按钮。 用于将所有的 表单输入项恢复到 **默认状态** |
| **file** | 附件框。用于文件上传。 |
| **hidden** | 隐藏域。  用户id、商品id |
| **button** | 普通按钮  后面结合js、jq 使用 |
| **date** | 定义日期字段 |
| **number** | 定义数字字段 |
| **email** | 定义email地址字段 |
| **Image** | 定义图片作为按钮  提交表单数据 |
| **Color** | 取色器 |

## 1.3、HTML-表单输入项标签<select>

列表：ol、ul下的是li

* <select> 定义一个下拉列表。可以进行单选或多选。
* <option>定义一个下拉列表的列表项。

select属性：

|  |  |
| --- | --- |
| 属性名 | 属性说明 |
| **name** | 组件名。  提交的参数名 |
| Multiple | 设置该标签选项全部显示，并且可以进行多选提交。。 |

option属性：

|  |  |
| --- | --- |
| 属性名 | 属性说明 |
| **value** | 设置需要提交的参数值。  如果不设置value属性的值，默认提交的是option的内容提  如果设置value，提交的就是value的值 |
| **selected** | 设置某个列表项默认选中。 |

## 1.4、HTML-表单输入项标签<textarea>

* <textarea></textarea>标签

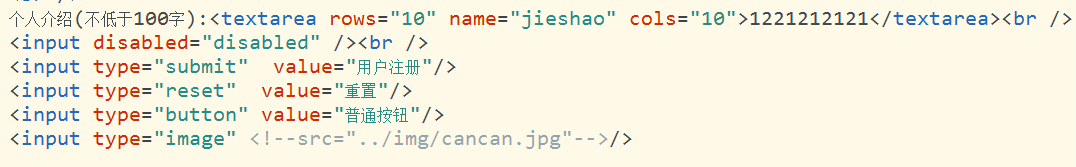
文本域标签，可支持用户输入并提交大量纯文字数据。

textarea属性：

|  |  |
| --- | --- |
| 属性名 | 属性说明 |
| **name** | 组件名。  如果需要下拉选择框数据提交到服务器，必须提供name属性 |
| **cols** | 设置文本域列 |
| **rows** | 设置文本域行数 |
|  |  |

课堂代码：





## 1.5、表单标签<form>-常用属性

from属性：

|  |  |
| --- | --- |
| 属性名 | 属性说明 |
| **action** | 将表单数据提交到何处。  目前,action不用写(默认提交到本页面)  后面，action后写的是服务器的地址 |
| **method** | 将数据以何种方式提交。  Method的取值有七种  常用的：get(默认) post |

Get 和 Post的区别？

1. 提交数据的方式不同

Get：数据在地址栏上



Post：数据在请求体中



1. 安全性方面

Get：数据直接暴露在地址栏上，数据不安全

Post：请求体中，相对而言比较安全

1. 提交数据量不同

Get：提交数据量有限

Post: 提交数据量无上限

文件上传：

请求方式必须是post

## 1.6 标签span和div

### 1.6.1、HTML-标签-<span>

<span>行级的标签，用于在效果中 一行上定义一个块，进行内容显示。

1. span有多少内容，就会占用多大空间。

②Span**不会自动换行**

### 1.6.2、HTML-标签-<div>

块级的标签，用于在效果中 定义一块，默认占满一行，进行内容的显示

1. 默认占满一行
2. 会**自动换行**

**Div和span的应用场景：**

**Div：用于做网站的布局**

**Span：用于展示少量的信息**



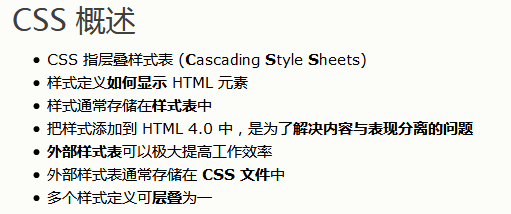
# CSS

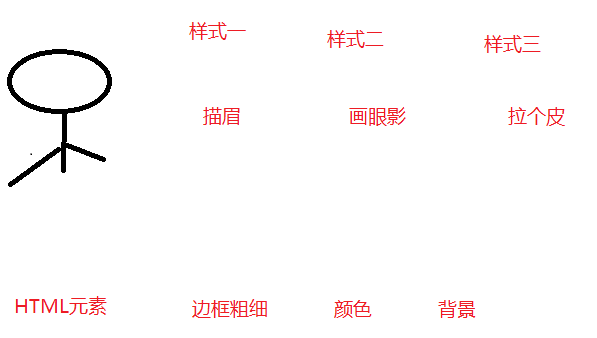
对大家的要求：能看看懂别人的写的CSS代码

## 2.1、css简述

### 2.1.1、css是什么？为什么要学习css?

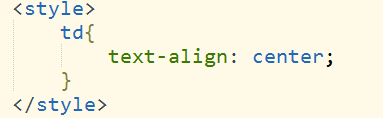
CSS：美化网页





CSS的好处：

1. 功能比HTML更加强大

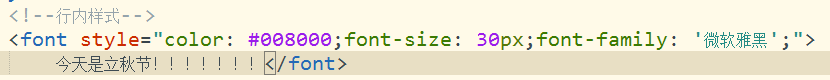


1. 代码的耦合度降低(协同开发)

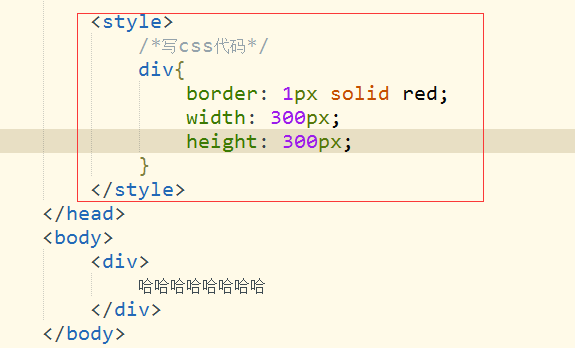
前后端分离

### 2.1.2、css的位置

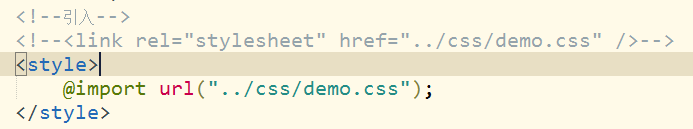
1、行内样式(样式直接嵌入在html元素上)



2、内部样式



3、外部样式



如果以上三种方式同时作用在html元素上，遵循就近原则：行内》内部》外部

### 2.1.3 CSS的格式规范

选择器名称{

属性名：属性值;

属性名:属性值;

......

}

注意：

1. 选择器有很多种
2. 属性名和属性值是一种键值对的关系，用冒号隔开
3. 如果一个属性名有多个值 ，那么多个值之间用空格分开



4、要以分号结尾

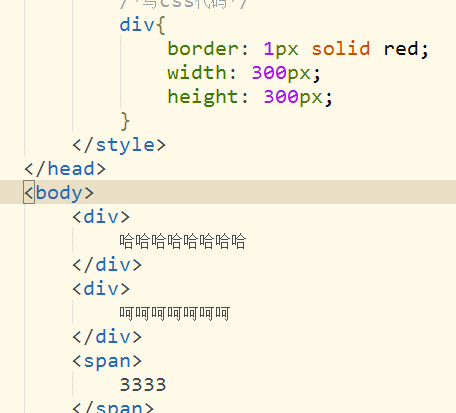
## 2.2、css选择器

### 2.2.1、元素选择器

HTML标签又名HTML元素。

元素选择器：即以HTML标签名作为选择器名称。

作用：选择CSS样式代码 作用于 对应标签名的标签上。



### 2.2.2、类选择器

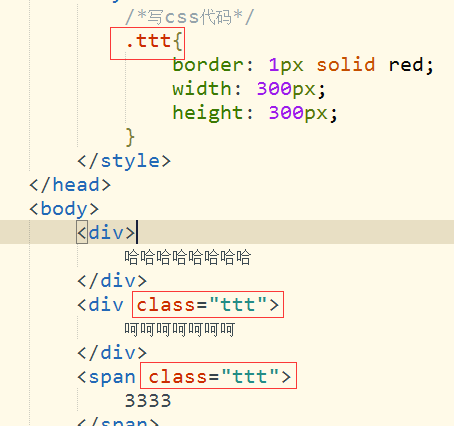
每个HTML标签都有一个class属性，class属性值即为类名

类选择器：即以HTML的类名（class属性值）作为选择器名称。

作用：选择CSS样式代码 作用于 对应类名的HTML标签上

格式：.类名{

}



### 2.2.3、id选择器

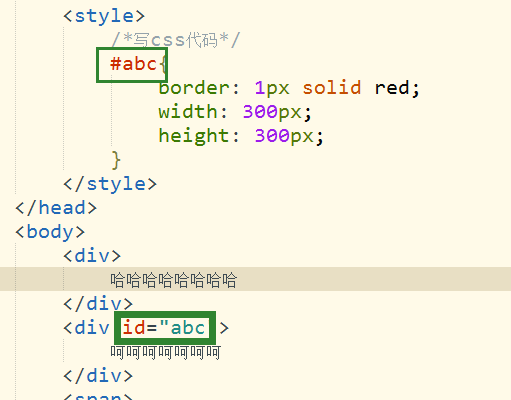
每个HTML标签都有一个id属性，id的属性值必须在本页面是唯一的。

id选择器：即以HTML的id（id属性值）作为选择器名称。

格式：#id名{

}

作用：选择CSS样式代码 作用于 某个规定id值的html标签上



## 2.3、css样式

### 2.3.1、边框/背景属性

所有的HTML标签都有边框，默认边框不可见

#### 2.3.1.1、border

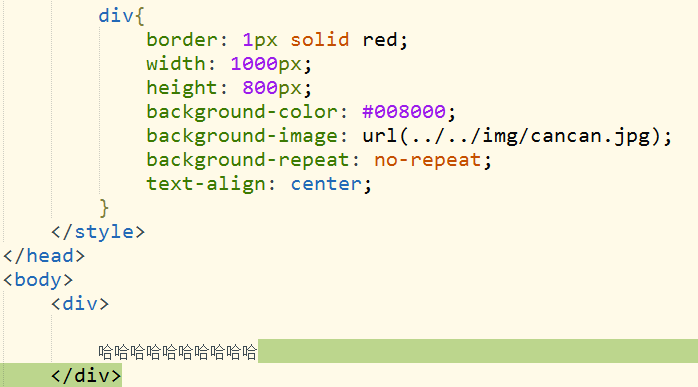
#### 2.3.1.2、width

#### 2.3.1.3、height

#### 2.3.1.4、background-color

#### 2.3.1.5、background-image

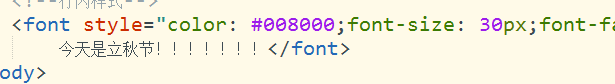
#### 2.3.1.5、background-repeat



### 2.3.2、字体

#### 2.3.2.1、font-size 字体大小

#### 2.3.2.2、color 字体颜色



### 2.3.3、布局

#### 2.3.3.1、float

通常默认的排版方式，将页面中的元素从上到下一一罗列，而实际开发中，需要左右方式进行排版，就需要使用浮动属性

* + 格式：

选择器{float:属性值;}

常用属性值：

none：元素不浮动（默认值）

left：元素向左浮动

right：元素向右浮动

注意：因为元素设置浮动属性后，会脱离原有文档流（会脱离原有的板式），从而会影响其他元素的样式，所以设置浮动以后，页面样式需要重新调整

原有的规则：原有的文档流

脱离原有的规则，必须脱离原有的文档流

**浮动属性，会使得原有的布局被打乱**

#### 2.3.3.2、clear

由于浮动元素不再占用原文档流的位置，所以它会对页面中其他元素的排版产生影响。

如果要避免影响，需要使用clear属性进行清除浮动。

* 格式：选择器{

clear:属性值;

}

常用属性值：

left：不允许该元素左侧有浮动元素（清除左侧浮动的影响）

right：不允许该元素右侧有浮动元素（清除右侧浮动的影响）

**both**：同时清除左右两侧浮动的影响（一般用both）

代码实现(见案例)

### 2.3.4、转换

#### 2.3.4.1、display

HTML提供丰富的标签，这些标签被定义成了不同的类型，一般分为：块级元素和行内元素。

* 块级元素：以区域块方式出现。每个块标签独自占据一整行或多整行。块结束会自动换行

<div> 、标题标签、<p> ul、ol

* 行内元素：根据内容多少来占用行内空间，不会自动换行（设置宽、高属性无效）

span a

想让行内元素可以设置宽高，只需要让行内元素转换成块级元素

display属性可以使得元素 在行内元素和块元素之间相互转换。

* 格式：

选择器{display:属性值}

* 常用的属性值：

block：此元素将显为块元素（块元素默认的display属性值）

inline：此元素将显示为行内元素（行内元素默认的display属性值）

none：此元素将被隐藏，不显示，也不占用页面空间。

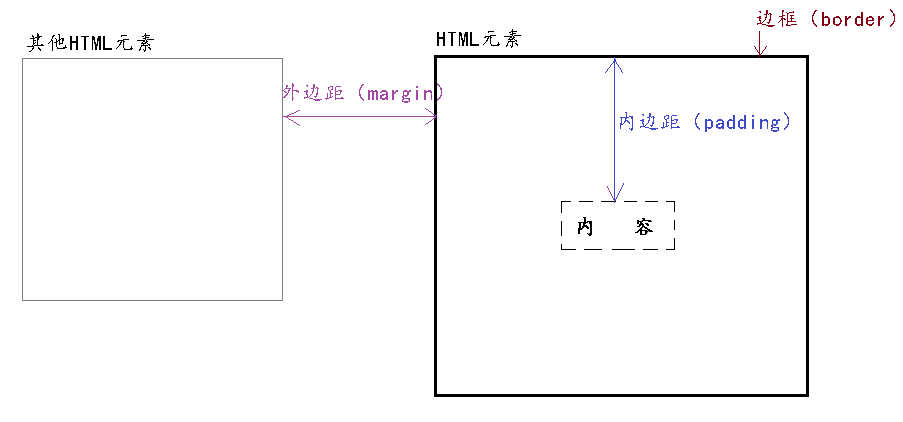
## 2.4、css盒子模型

### 2.4.1、什么是盒子模型

所有的HTML元素，我们都可以看成一个四边形，即一个盒子。

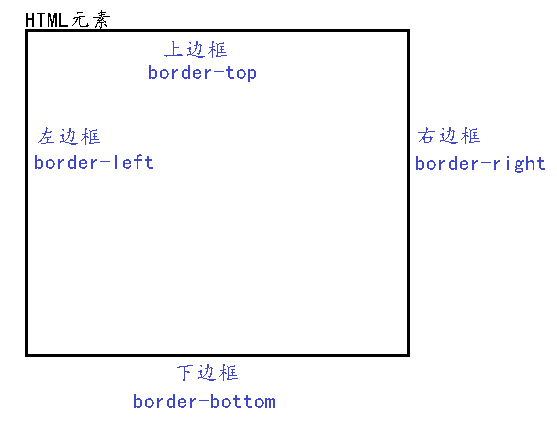
用CSS来设置元素盒子的 内边距、边框 和 外边距 的样式的方式，

相当于设置盒子的样式，所以我们将其称之为 盒子模型



### 2.4.2、边框

边框：HTML元素盒子的框体



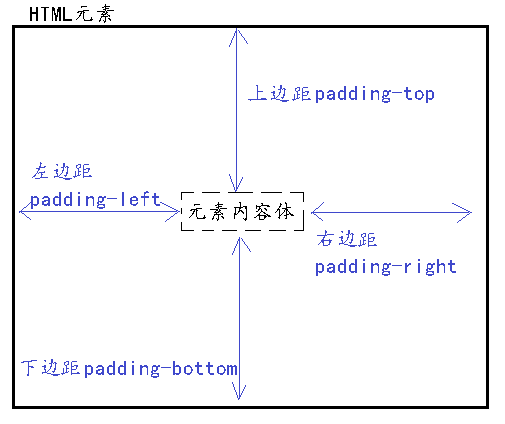
边框有四个属性可以设置：

* border-top:上边框
* border -right:右边框
* border -bottom:下边框
* border -left:左边框

通用设置：border

### 2.4.3、内边距padding

内边距：HTML元素里的内容体 到 HTML元素边框 的距离

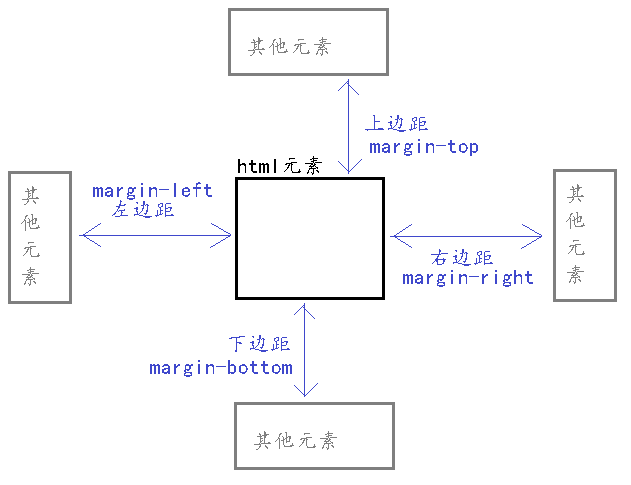


内边距有四个属性可以设置：

* padding-top:上边距
* padding-right:右边距
* padding-bottom:下边距
* padding-left:左边距

### 2.4.4、外边距margin

外边距：HTML元素边框 到 其他HTML元素边框的距离



外边距有四个属性可以设置：

* margin-top:上边距
* margin -right:右边距
* margin -bottom:下边距
* margin -left:左边距