# CSS样式

## display

\* - 这个样式可以用来设置元素的显示方式

\* - 可以设置一个元素到底是块元素还是内联元素

\* - 可选值：

\* - block,元素将会作为块元素显示

\* - inline,元素将会作为内联元素显示

\* - inline-block,元素将会作为行内块元素显示

\* 所谓行内块，就是元素既有块元素的特点，也有内联元素的特点，既可以设置宽高，又不会独占一行

\* - none,元素不会在页面中显示,并且不再占据页面的位置

## visibility

- 用来设置元素是否可见

- 可选值：

visible，默认值，元素在页面中可见

hidden，元素在页面中不可见 ，但是元素依然占据页面的位置

## overflow

子元素都是在父元素的内容区中摆放的，如果子元素的大小超过了父元素，则超出父元素的内容将会在父元素以外的位置显示，超出父元素的内容，我们称为溢出的内容，可以通过overflow来设置父元素，处理溢出内容的方式

overflow

- 可选值：

visible，默认值，溢出的内容会在父元素外显示

hidden， 溢出的内容将会被减去，不再显示

scroll，将溢出的内容隐藏，通过拖动滚动条来查看完整内容，该样式不论内容是否溢出，都会默认为父元素添加水平和垂直两个方向的滚动条

auto，根据内容自动生成滚动条，需要哪个方向的就生成那个方向的，如果都不需要则都不生成

# 默认样式

默认情况下，浏览器都会为一些元素添加一些默认的样式，比如 body都会有一个默认的外边距，但是这些默认样式我们开发中并不需要，所以我们就需要去除这些默认样式

去除所有元素的默认内外边距：

\*{

margin: 0;

padding: 0;

}

将列表的项目符号去除：

list-style: none;

# 文档流

文档流是网页的最底层，我们所创建的元素默认情况下都在文档流中

元素在文档流中的特点

块元素

1.块元素在文档流中自上向下排列（垂直排列）

2.块元素在文档流中默认宽度auto，也就是父元素的100%

3.块元素在文档流中默认高度被内容（子元素）撑开

内联元素

1.内联元素在文档流中自左向右水平排列，如果在一行中不能容纳所有的元素，则会换到下一行，继续自左向右排列

2.内联元素的宽度和高度都由内容撑开

# 浮动

使用float可以设置元素的浮动，当一个元素设置了浮动以后，将会脱离文档流，然后向页面的左上或右上移动，当元素脱离文档流以后，它将不会在文档流中占据位置，文档流中的其他元素将自动上移

- 可选值：

none，默认值，元素不浮动，在文档流中排列

left，元素向左浮动

right，元素向右浮动

1.当浮动元素遇到父元素的边框时，会停止移动

2.浮动元素移动时，如果遇到了其他的浮动元素，也会停止移动

3.如果浮动元素的上边是一个没有浮动的块元素，则浮动元素不会超过该元素

4.浮动元素不会超过他上边的浮动的兄弟元素，最多最多一边齐

浮动元素的层级要高于文档流中元素的层级

文字不会被浮动元素所覆盖，文字会环绕在浮动元素的周围，我们可以通过浮动来设置文字环绕图片的效果

当元素浮动以后会完全脱离文档流

1.块元素脱离文档流以后，不会再独占一行

2.块元素脱离文档流以后，宽度和高度都被内容撑开

3.内联元素脱离文档流，会变成块元素，它的特点和块元素脱离文档流的特点一样

总结：一旦脱离文档流，所有的元素都是块元素

# 高度塌陷

父元素的高度，默认被子元素撑开，此时如果子元素设置浮动，则会导致其完全脱离文档流，子元素脱离文档流将无法撑开父元素，导致父元素的高度丢失，就是我们说的高度塌陷问题。父元素一旦高度塌陷，则它下边的元素会向上移动，导致整个页面的布局混乱

解决方法：指定父元素的高度，但是将父元素的高度写死将无法自动适应子元素的高度，不太灵活。或者开启BFC (Block Formatting Context) 块级格式化环境

BFC是元素的隐含属性，默认情况下BFC是关闭的，当元素开启BFC以后将会具有如下特性：

1.开启BFC以后，元素不会被浮动的元素所覆盖

2.开启BFC以后，子元素的垂直外边距不会传递给父元素

3.开启BFC以后，元素可以包含浮动的子元素

我们并不能直接去开启BFC，而需要使用一些特殊的样式来间接的打开元素的BFC,

开启BFC的方式很多，但是都会有一些副作用，我们需要选择一些副作用小的方式

1.设置元素的浮动，但会丢失宽度，而且浮动后还会影响页面布局

2.设置元素的display为inline-block，会丢失宽度，但不会影响布局

3.设置元素绝对定位，和浮动差不多

4.将元素的overflow设置为一个非visible的值，一般设置为hidden

# 定位

定位

- 通过定位可以将元素摆放到页面中的指定位置，但是默认情况下元素并没有开启定位，所以使用定位之前需要先开启定位。

- 通过position属性来设置元素的定位

- 可选值：

static，默认值，此时元素没有开启定位

relative，开启元素的相对定位

absolute，开启元素的绝对定位

fixed，开启元素的固定定位

## 相对定位

当元素的position属性值设置为relative时，则开启了元素的相对定位

- 相对定位的特点：

1.开启相对定位以后，如果不指定偏移量，元素位置不会发生任何变化 2.相对定位的元素是相对于其自身在文档流中的位置进行定位的

3.相对定位的元素不会脱离文档流

4.相对定位会使元素提升一个层级（文档流中的元素 < 浮动元素 < 定位元素）

5.相对定位不会改变元素的性质，块还是块，内联还是内联

- 偏移量（offset）

- 当元素开启了定位以后，可以通过四个偏移量来设置元素的位置

- top：元素相对于其定位位置的上边的偏移量

- right：元素相对于其定位位置的右边的偏移量

- bottom：元素相对于其定位位置的底部的偏移量

- left：元素相对于其定位位置的左侧的偏移量

一般情况下，我们对一个元素进行定位，只需要使用四个偏移量中的两个

top left

top right

bottom left

bottom right

## z-index

- z-index，只对开启了定位的元素起作用

- 他可以用来设置元素的层级

- 它需要一个整数作为值，数越大层级越高，层级越高越优先显示，如果层级一样，谁在下边显示谁

- 父元素永远都不会盖住子元素

## 绝对定位

- 当前元素的position属性设置为absolute时，则开启了元素的绝对定位

- 绝对定位的特点：

1.绝对定位的元素会完全脱离文档流

2.绝对定位的元素如果不设置偏移量，位置不会改变

3.绝对定位的元素会相对于离它最近的开启了定位的祖先元素进行定位，如果所有的祖先元素都没有开启定位，则相对于浏览器的窗口进行定位，所以一般我们为一个元素开启绝对定位的同时，都会为它的父元素开启相对定位

4.绝对定位会提升元素的层级

5.绝对定位会改变元素的性质，内联元素变成块元素，块元素的宽度被子元素撑开

## 固定定位

固定定位也是一种绝对定位，它是一种特殊的绝对定位，它的大部分特点都会绝对定位一致，不同的是固定定位永远相对于浏览器的窗口进行定位，而且固定的定位的元素一旦定位，则会固定在浏览器窗口中不动，即使浏览器的滚动条滚动，它也不会移动。如果只设置固定定位，而没有设置偏移量，是基于body定位的

## 设置元素透明

通过opacity，可以设置元素的透明效果

该样式需要一个0-1之间的值

1 表示完全不透明

0 表示完全透明

\* 通过opacity设置的透明度会使子元素也跟着透明（尽管opacity属性继承性为no），如果需要子元素不透明，不能使用opacity，需给父元素的背景颜色设置为background-color：rgba(r,g,b,a);括号中r,g,b为 颜色浓度值，a取值范围为0-1.

opacity这个属性不支持IE8及以下的浏览器

IE8及以下的浏览器需要使用其他的方式来设置

filter: alpha(opacity=60);

这里的透明需要一个 0 - 100之间的值

100表示完全不透明

0 表示完全透明

# 表格

## 表格简介

在日常生活中经常使用表格来表示一些格式化的数据：

比如：课程表、账单、人名单

同样在网页中也可以来创建表格，来表示格式化的数据

我们使用table标签来创建表格 table是一个块元素

在table中使用tr来代表表格中的一行，有几行就有几个tr

在tr中使用td代表一行中的一个单元格，有几个单元格就有几个td

th标签专门用来表示表头中的单元格的，所以在表头中可以使用th来替换td

rowspan表示纵向合并单元格

<td rowspan="2">B4</td> 合并2个纵向单元格

colspan表示横向的合并单元格

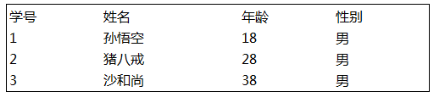
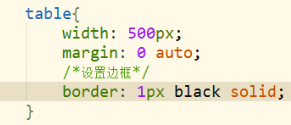
<td colspan="2">D3</td> 合并2个横向单元格

## 表格样式

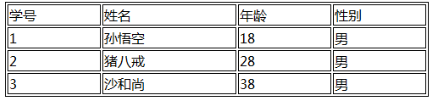
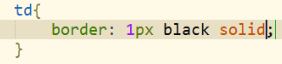
当创建一个表格后，默认是没有格的



可以给table设置样式



但只有table有边框，还需要给td设置边框



但单元格和单元格是独立的，没有合并，而且边框之间还有距离

border-spacing可以用来设置表格边框之间的距离

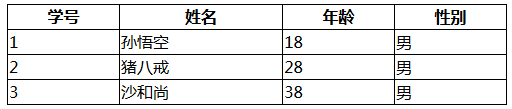
border-spacing: 0;



但发现是把两个边框合并起来了，边框有点粗

border-collapse可以用来设置表格的边框合并，当设置border-collapse以后border-spacing自动失效

border-collapse: collapse;



th标签专门用来表示表头中的单元格的，所以在表头中可以使用th来替换td，如上图第一行，加粗并居中

在我们的HTML中还为我们提供了 thead tbody tfoot 来将表格分成三个部分

thead 表示表格的头部，可以将头部的tr放到thead中，thead中的内容，永远会显示在表格的头部，不论thead在源码中位置在哪

tfoot 表示表格的底部，可以将底部的tr放到tfoot中，tfoot中的内容，永远会显示在表格的底部，不论tfoot在源码中位置在哪

放到thead和tfoot中的内容，打印时每页都有thead和tfoot

tbody 表示表格的主体，可以将主体内容的tr放到tbody中

我们在创建一个表格时，如果没有指定tbody，则浏览器会在页面中自动添加tbody，并且将所有的tr都放到tbody中，一定要注意，默认情况下tr是tbody的子元素，而不是table的子元素

# 表单

在网页中，我们通过表单来向服务器提交信息

在网页中使用form标签来创建一个表单

- form中还必须有一个action属性

action需要的是一个服务器的地址，这样当提交表单时，我们所填写的信息将会提交到action对应的地址

使用form创建完表单以后，还需要向表单中添加表单项

## 文本框

- 使用input标签来创建一个文本框 input为内联元素

文本框需要一个type属性，属性值是text

- 如果希望表单中的内容会提交到服务器中，

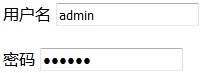
还必须为元素指定一个name属性,当添加了name属性以后，填写的内容才会被提交到服务器

## 密码框

- 密码框使用input标签，他的type属性是password

用户名 <input type="text" name="username" />

密码 <input type="password" name="password" />



- 用户填写的内容，默认会以查询字符串的形式发送给服务器：

username=admin&password=123123

- 查询字符串实际上就是一个名值对结构，名字就是表单项的name属性值，而值是用户填写的内容，多个名值对之间使用&连接，这样数据发送给服务器以后，服务器就可以根据name来获取用户填写的值

## 单选按钮

- 使用input标签来创建单选按钮，它的type属性值是radio

- 单选按钮需要通过name属性进行分组，name属性值相同属于一组

- 像这种选择框，不需要用户填写内容的，还必须指定一个value属性，如果指定了value属性，则当提交表单时，value属性的值将会提交到服务器

- 如果需要设置某个单选按钮或多选框为默认选中状态，可以在元素中添加一个属性checked="checked"

性别 <input type="radio" name="gender" value="male" />男

<input type="radio" name="gender" value="female" checked="checked" />女 

## 多选框

- 使用input来创建多选框，它的type属性是checkbox

爱好 <input type="checkbox" name="hobby" value="lq" /> 篮球

<input type="checkbox" name="hobby" value="ymq" /> 羽毛球

<input type="checkbox" name="hobby" value="ppq" checked="checked" /> 乒乓球

<input type="checkbox" name="hobby" value="zq"/> 足球



## 下拉列表

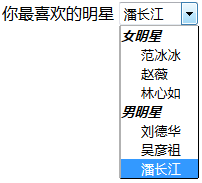
- 使用select标签来创建一个下拉列表

- 使用option标签来向下拉列表中添加选项

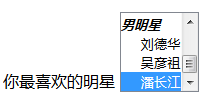
- name属性需要指定给select，value需要指定给option

- 在select标签中还可以使用optgroup来对选项进行分组

- 可以通过为option添加一个属性 selected="selected" 来将其设置为默认选中



- 如果在select中添加一个 multiple="multiple" ，则下拉列表会变成多选的下拉列表，按住Ctrl多选



## 提交按钮

- 提交按钮可以将表单中的信息提交到服务器

- 使用input标签来创建一个提交按钮，提交按钮的type属性值是submit

- 在提交按钮中可以通过value属性来指定按钮上的文字

<input type="submit" value="注册" /> 

## 重置按钮

- 重置按钮可以将表单恢复到默认状态

- 重置按钮的type属性值是reset

<input type="reset" value="Reset" /> 

## 单纯按钮

如果将type属性值设置为button则会创建一个单纯的按钮，这个按钮只有一个功能就是被点

<input type="button" value="按钮" /> 

## button标签

也可以使用button来创建按钮，它的作用和input的按钮是一样的，不同的是button是成对的标签，而input是自结束标签，相比来说button标签更灵活一些

<button type="submit">提交</button>

<button type="reset">重置</button>

<button type="button">按钮</button>

