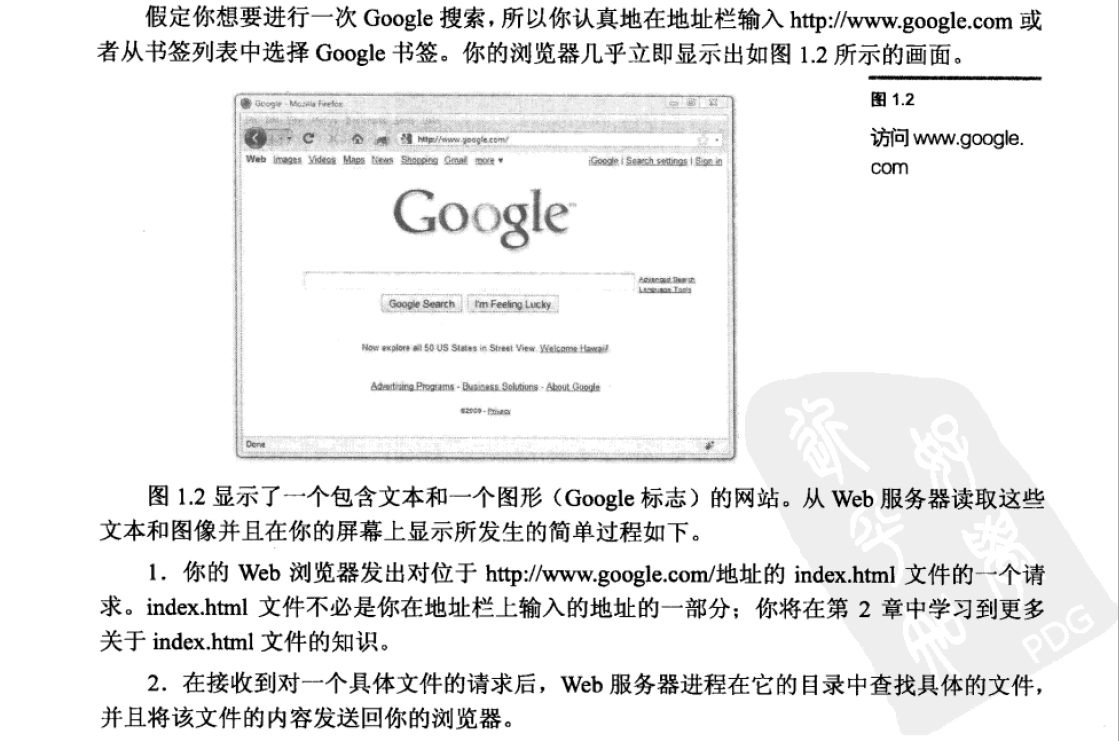
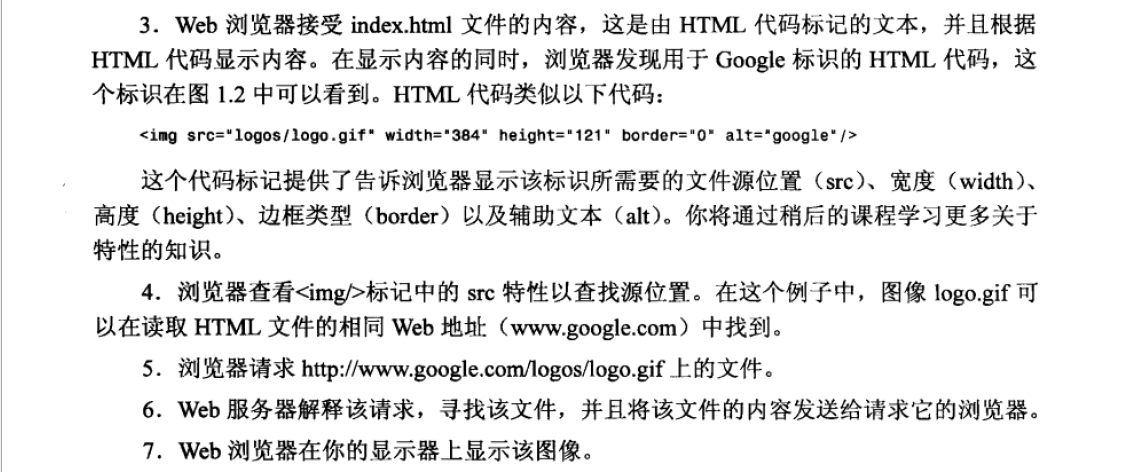
# Html

## 浏览器和服务器的访问过程：



### Web内容和网页

万维网比作图书馆。这个图书观众，单独的网站是书。网站上的单独文件就“页”。

一个“网站“由一起创建并且内容相关的一页或多页组成。“主页”通常指的是人们访问网站时看到的第一个网页。

### Web内容：是描述Web服务器传递到Web浏览器的文本、图像、音频、视频和其他的媒体文件的全称。

### html的结构：

1、结构Html用于描述页面的结构。

2、表现：Css用于控制页面元素的样式。（一般看到的）

3、行为：JavaScript用于响应用户的操作。

### 互联网的三大基石

HTML语言被称为互联网的三大基石之一(其余两大基石分别为：HTTP协议、URL)。它解决了如何以丰富的效果展示数据内容的问题。互联网中，数据是在服务器和浏览器之间互相传送和执行。三大基石分别解决了如下问题：

HTTP协议：解决了服务器和浏览器之间数据如何传送、传送格式的问题！实现了分布式的信息共享。

URL协议：解决了众多服务器中资源定位问题。从而让浏览器可以访问不同的服务器资源，实现了全球信息的精确定位。

HTML语言：解决了数据在浏览器中如何丰富多彩的展示，及如何合理标示信息的问题。

## 什么是html ？（htlm是前的版本）

**HyperText Markup language:**超文本标记语言，是用超链接的方法，将各种不同空间的文字信息组织在一起的网状文本。

**纯文本**：在纯文本编辑器中编辑的内容，纯文本中只能保存文本内容，其他格式的内容都不能设置。

**超文本（(hypertext）**：超出文本的范畴。是用超链接的方法，将各种不同空间的文字信息组织在一起的网状文本。使用html可以轻松的实现这样的操作。

**标记（Markup）**：html：里面所有的操作都可以用标记实现，标记就是标签。

标签的名称：，<标签名称>。

## 创建html文件

### HTML文件的后缀是 .html或 .htm

.jsp：java服务器页面。

.asp:Microsoft 活动服务页面。

.phpL超文本预处理器。

### Html标签

格式：<标签名称> 文本 </标签名称>

开始标签 结束标签“/”开头

标签分类：单标签

双标签

### Html的语法格式：

1)HTML中不区分大小写，但是一般都会用小写。

2) HTML的注释不能嵌套

3)HTML的标签结构必须完整，要么成对出现，要么自结束标签。

4)不符合语法规范的内容，浏览器尽最大的努力修正。例如将内容写到html标签的外面，他会修正将它放到body里面。

5)HTML中的标签可以嵌套，但是不能交叉嵌套。

6)HTML标签中的属性必须有值，且值必须加引号。

### html的格式

html的基本你结构：<html>标题和正文</html>

<html>

<head></head>

<body></body>

</html>

html中所有的内容都要放进去。head和body是html标签的子标签head和body是兄弟标签。

html中所有的内容都要放进去。head和body是html标签的子标签

head和body是兄弟标签。

<head>设置相关信息</head>

标题部分对页面进行描述，并不显示在网页浏览器中，帮助浏览器解析网页。

head中的标签

<title></tiltle> head中的标签，内容会显示在网页的标题栏，搜索引擎在检索页面时，会首先检索title中的内容，他是对搜索引擎的最重要的内容，会影响网页在搜索引擎的中的排名。

<meta>META标签是HTML语言HEAD区的一个辅助性标签，META标签用来描述一个HTML网页文档的属性，例如作者、日期和时间、网页描述、关键词、页面刷新等。它提供的信息虽然用户不可见，但却是文档的最基本的元数据。

<body>正文部分</body> html的主体。该处的所有信息将会显示在Web浏览器窗口的主显示区域。

1.特殊的标签

在html中，字符之间写再多的空格，浏览器也会当成一个空格解析，换行也会当成一个空格解析。

换行标签， <br/> 斜杠在后面指的是在标签内结束。（也可以不加/）

一条水平线 <hr/>。

2.html页面的显示不会kaolv行尾和文本间的空格。

3.在html中是不区分大小写的。

4.head和tiltle的区别

<tilte>:指定整个网页的标题，但并不显示在网页上，而是显示在浏览器窗口的标题栏中。

<head>:使一些文本体现在网页上

作用：标记的文本部分并搞死Web浏览器这写文本的的类型，使这些浏览器正确的显示这些文本。

例如：

<font size="5" color="red">这是一个html文件。（html主要学习这些标签）

### 属性

可以通过属性来处理标签中的内容，可以在开始标签中添加属性。

注：1、属性写在开始标签中。

2、属性要和标签用空格分开。

3、属性名 = “属性值”

4、一个标签中可以设置多个属性，多个属性之间用空格隔开。

<h1>这是我<font color = "red" size = "7">的第</font>二<>个</h1>

### 乱码问题：

GB2312：中文系统默认编码

utf-8:万国码

ANSI：自动以系统的默认编码来保存文件。

产生乱码的原因：编码和解码采用的字符集不同。

在中文系统的浏览器里，默认都是GB2312

<meta> :设置网页的一些元数据，比如字符集、关键字、简介、

是一个自结束标签（在标签后面加/）

### html的操作思想

网页中有很多的数据，不同的数据可能有不同的显示效果，这个时候需要用标签把要操作的数据包起来（封装）通过修改标签的属性值实现标签内数据要是的变化。

一个标签相当于一个容器，想要修改容器的样式，只要改变容器的属性值，就可以实现容器内数据样式的变化。

### doctype声明

HTML的版本挺多，至少有三个版本广泛使用，为了让浏览器知道我们使用的html版本，我们还需要在网页的最上边调价一个doctyoe声明，来告诉浏览器网页的版本。

Html5中：<!DOCTYPE html>声明当前的网页是按照html5的标准编写的。如果不写文档声明，回到使浏览器进入一个怪异模式，进入怪异模式后，浏览器解析页面会导致页面无法正常显示，所有避免进入该模式，一定要写文档声明，文档声名的位置在html代码中的第一行。

## html的head中的标签

### title标签

在<title>和</title>标签之间的文字内容是网页的标题信息，它会出现在浏览器的标题栏中。网页的title标签用于告诉用户和搜索引擎这个网页的主要内容是什么，搜索引擎可以通过网页标题，迅速的判断出网页的主题。每个网页的内容都是不同的，每个网页都应该有一个独一无二的title。

### meta标签

meta：META标签是HTML语言HEAD区的一个辅助性标签，META标签用来描述一个HTML网页文档的属性，例如作者、日期和时间、网页描述、关键词、页面刷新等。它提供的信息虽然用户不可见，但却是文档的最基本的元数据。

元数据(Metadata)是用来概括描述数据的一些基本数据。

meta标签共有两个属性：http-equiv和name;不同的属性又有不同的参数值，这些不同的参数值就实现了不同的网页功能。

name属性： name属性主要用于描述网页，与之对应的属性值为content，content中的内容主要是便于搜索引擎机器人查找信息和分类信息用的。

语法格式：＜meta name=”参数” content=”具体的参数值”＞

name的属性值：

Keywords (关键字)：keywords用来告诉搜索引擎你网页的关键字是什么。

例如：＜meta name ="keywords" content="science, education,culture,politics,ecnomics，relationships, entertaiment, human"＞

description (网站内容描述)：description用来告诉搜索引擎你的网站主要内容。

例如：＜meta name="description" content="This page is about the meaning of science, education,culture."＞

author(作者)：标注网页的作者

例如：＜meta name="author" content"root,root@21cn.com"＞

http-equiv水性: 顾名思义，相当于http协议中文件头的作用，它可以向浏览器传回一些有用的信息，以帮助正确和精确地显示网页内容，与之对应的属性值为content，content中的内容其实就是各个参数的变量值。

语法格式：＜meta http-equiv=”参数” content=”参数变量值”＞

http-equiv属性：

content-Type(显示字符集的设定)：设定页面使用的字符集。

＜meta http-equiv="content-Type" content="text/html; charset=gb2312"＞

Expires(期限) ：可以用于设定网页的到期时间。一旦网页过期，必须到服务器上重新传输。

＜meta http-equiv="expires" content="Fri, 12 Jan 2001 18:18:18 GMT"＞

注：时间格式必须是ＧＭＴ格式。

Pragma(cache模式)：禁止浏览器从本地计算机的缓存中访问页面内容。

格式：＜meta http-equiv="Pragma" content="no-cache"＞

注意：这样设定，访问者将无法脱机浏览。

Set-Cookie(cookie设定)： 设置cookie, 如果网页过期，那么存盘的cookie将被删除。

＜meta http-equiv="Set-Cookie" content="cookievalue=xxx; expires=Friday, 12-Jan-2001 18:18:18 GMT； path=/"＞

meta标签对搜索引擎的作用：

meta标签的一个很重要的功能就是设置关键字，来帮助你的主页被各大搜索引擎登录，提高网站的访问量。在这个功能中，最重要的就是对Keywords和description的设置。因为按照搜索引擎的工作原理,搜索引擎首先派出机器人自动检索页面中的keywords和decription，并将其加入到自己的数据库，然后再根据关键词的密度将网站排序。因此，我们必须设置好关键字，来提高页面的搜索点击率。下面我们来举一个例子供大家参考

＜meta name="keywords" content="政治,经济, 科技,文化, 卫生, 情感，心灵，娱乐，生活，社会，企业，交通"＞

　　＜meta name="description" content="政治,经济, 科技,文化, 卫生, 情感，心灵，娱乐，生活，社会，企业，交通"＞

设置好这些关键字后，搜索引擎将会自动把这些关键字添加到数据库中，并根据这些关键字的密度来进行合适的排序。

### link

1、引用外部样式表

使用方法 : <link rel="stylesheet" type="text/css" href="1.css"/>

type="定义当前文本内容以层叠样式表（CSS）来解析。

rel(relationship)="定义当前文档与被链接文档之间的关系，这里是调用外部文件，stylesheet即代表css样式表。

href="定义被链接文档的位置，这里链接所指向的是：同一目录下名为style.css的文件。

2、设置网页图标

语句一：<link rel="shortcut icon" href="图片地址" />

语句二：<link rel="icon" href="图片地址" type="image/gif" />

注意 : icon是一种图标格式，用于系统图标、软件图标等，这种图标扩展名为\*.icon、\*.ico。

1. href指向图片的路径,一般将图标放置在网站的根目录下,便于搜索引擎抓取.

2. 图片不强制要求为ico文件.jpg或者png等图片格式的文件也可以

　type可以设置多种图片类型。如：gif/png/ico

3.一般的图标建议是16\*16或者32\*32的，必须是8位色或者24位色的，格式必须是png或者ico或者gif。

16\*16/32\*32 且 8位或24位色 且 png/ico/gif。

4. 静态的图标文件使用

<link rel="shortcut icon" href="1.png" type="image/x-icon" />

5. 动态图 gif（动画也是16\*16）使用：‘

<link rel="icon" href="1.gif" type="image/gif" >

案例：使用link rel="shortcut icon"为网页标题加图标

<title>会员卡券</title>

<link rel="shortcut icon" href="http://GT/img/favicon.png" type="image/x-icon">

实现效果：https://images2017.cnblogs.com/blog/1074458/201709/1074458-20170908132851319-522668898.png

添加到收藏夹：

## html的body中的标签

### 文字标签：修改文字的样式：<font> </font>（基本不使用，要用css来设置）

属性：size:文字的大小。字体大小范围：0-7。注：设置的字体大小大于7，字体默认为7。

color：文字颜色。

两种表示的方法是：英文单词：red、green、blue..

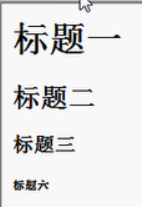
十六进制表示：#ffff（RGB）通过工具来获取颜色。

### 注释标签：<!—注释内容—>

源文件存在注释，页面不打印注释。

### 、标题标签

标题标签：一共有6级标签，在显示效果上h1最大，h6最小，到那时文字的大小我们不关心，使用html标签时，关心的是标签的语义，我们使用的标签都是语义化标签，6级标题中h1最重要。表示一个网页中的主要内容h2-h6重要性依次降低，对于搜索引擎来说

h1的重要性仅次于title，搜索检索完title，会立即查看h1中的内容，所以h1标签非常重要，它会挺像页面的搜索引擎的排名，页面只能写一个h1。

般页面中标签只是用h1,h2.h3,后面的基本不使用。

<h1>标题一</h1>

<h2>标题二</h2>

<h3>标题三</h3>

…….

<h6>标题六</h6>

注：从h1-h6，标题的大小依次变大，同时标题会自动换行。

标题标签的属性：

### 水平标签

水平线标签：</hr>

属性：size：水平线的粗细。范围1-7.

color：水平线的颜色。

### 特殊字符

特殊字符：在html中，一些如<>这种特殊字符是不能直接使用的，需要使用一些特殊字符，这些特殊符号我们称之为转义字符（实体），浏览器解析到实体时，会自动转换为对应的字符。

需求：想要在页面显示“<html>这是网页的开始”

例：<body> <html>这是网页的开始 </body>

显示效果：这是网页的开始（这里将<html>当做是标签来解析）

如果想要对这<html>进行显示，需要使用特殊字符。、

“<”的转义字符是 &lt; less than

“>”的转义字符是 &gt; greate than

“<html>”的转义字符：&lt;html&gt;

空格的转移字符：&nbsp/;（Non-Breaking Space）

版权符号的转义字符：&copy；

### 段落标签

段落标签：用于表示内容的一个自然段，使用p标签来表示一个段落。p标签的文字，会默认独占一行，并且段和段之间会有间距，同时p标签是一个块元素，align元素都适用于块元素。

效果：段落标签 比br标签多一行 ,不能放div元素。

<p>这是p标签</p>

### 列表标签：

1、无序列表

2、有序列表

3、定义列表

实现如下效果

显示效果：阿里巴巴

财务部

人事部

工程部

1、定义列表：<dl></dl>项目及其含义的列表，这个项目前面没有特殊的字符和数字、字母。

在dl中有 <dt></dt>上层内容

<dd></dd>下层内容

例：<dl>

<dt>阿里巴巴</dt>

<dd>财务部</dd>

<dd>人事部</dd>

<dd>工程部</dd>

</dl>

2、有序列表：<ol></ol>:每个列表项目前有数字或者字母的缩进列表。

显示效果：阿里巴巴

1、财务部

2、人事部

3、工程部

<ol>列表里有属性type 设置排序的方式：

a：为小写字母作为序号

A：为大写字母作为序号

1(默认情况) i

<ol>列表的里面有：<li>具体内容</li>

例:<ol type=“a”>

<li> 财务部</li>

<li> 人事部</li>

<li> 工程部</li>

</ol>

3、无序列表：

使用 ul标签来创建一个无序列表

使用li在ul中创建一个一个的列表项

一个li就是一个列表项

通过type属性可以修改无序列表的项目符号

注意：默认的项目符号我们一般都不使用，不同浏览器显示的效果不一样。如果需要设置项目符号，则可以采用为li设置背景图片的方式来设置。

去掉项目符号：

ul{

list-style:none

}

显示如下效果

特殊符号方框 财务部

特殊符号方框 学工部

标签：<ul></ul> 每个列表项目前有一个项目符号或者其他的符号缩进列表。

ul总可以添加属性值type：空心圆(circle)、实心方块(square)、实心圆(disc)默认为disc

例：<ul type=“circle”>

<li> 财务部</li>

< li> 人事部</li>

<li> 工程部</li>

</ul>

### 图像标签（image）

使用img标签来向网页引入一个外部图片，图像标签的作用要将图像放在网页上，首先将图像文件放在HTML文本文件所在的文件夹下，或者为了更容易组织放在名为Images的目录下。

JPEG(jpg):支持颜色比较多，图片可以压缩，但是不支持透明。一般使用JPEG来保存照片等颜色丰富的照片。

GIF：GIF支持颜色少，只支持简单的透明，支持动态图

PNG:支持颜色多，并且支持复杂透明，可以用来显示颜色复杂的透明的图片。

图片的使用原则:如果效果不一致，使用效果好的。效果一样，使用效果小的。

标签格式：

<img src=”图片的路径” alt=“my image” width=”300” height = “400” />（source） 标签属性：

src：标识图像文件，设置一个外部图片的路径

width：图片的宽度（像素为单位）

height：图片的高度（像素为单位）

alt（alternate text 替换文本））：关于图片的描述性文本。当用户看不到图像是，将看到alt属性指定的文本。搜索引擎可以通过alt属性来识别不同的图片。如果不写alt属性，则搜索引擎不会对img的图片进行收录。

宽度和高度两个属性如果只设置一个，另一个也同时会等比例调整大小。若两个值同时设定，则按照你指定的值来设置。

title：在图像可见时提供的图像的描述。

例：<img src = “b.jpg” width=”300” height = “400” title=”这是一个美女”/>

当图片不存在是 显示显示图片的内容（alt这个属性兼容性差，有些浏览器不显示）

考虑到可能看到消息的用户情况是不同的，因此网页上的每个<img>标签必须包含一个合适的alt属性。通常最好使用对图像的简短描述。

2、对齐图像（图像一般为左对齐left 是text-align的默认值）css

1.设置图片的位置

<div style="text-align:center"><img /></div>

<img style=”float:right”>

1.水平对齐文本：

<div src=”路径” style=“text-align:center”>

<div src=”路径” style=“text-align:left”>

<img style=”float:left”>

<img style=”float:right”>

2.垂直对齐图像（图像和文本的位置关系）

有时可能希望在一行文本中插入一个小图像，或将一行文本放在图像旁边作为标题。在这些情况下，如果可以控制文本和图像在垂直方向上如何对齐将很方便。图像的底部与文本的底部对齐，还是文本和图像的中间对齐?可以在下列选项和其他一些选项中进行选择。

要使图像的顶部与同一行中最高的图像或文本的顶部对齐，使用

<img style="vertical-align:text-top">。

要使图像的底部与文本的底部对齐，使用

<img style="vertical-align:text-botom">。

要使图像的中心与同一行中其他内容的中心对齐，使用

<img style="vertical-align:middle">。

要使图像的底部与文本的基线对齐，使用

<img style="vertical-align:baseline">。

例如：现在在网页中添加一些图像，并尝试使用text-align、vertical-align和float。

开始之前，先看看如何在网页中添加一张鱼的图像（fish.jpg）。

（1将图像文件fish.jpg复制到HTML文件所在的目录下（或者将图像放在它的当前位置并且确认记住这一位置）。

(1)使用文本编辑器，在HTML文件中要显示图像的位置添加<imgsrc="fish.jpg"alt="Bright Orange Goldfish with Bulging Eyes\">.

<p><img src="elcap\_sm.jpg"alt="E1 Capitan"width="100"height="75"style="vertical-align:text-top;"/>图片

<strong>E1Capitan</strong>is a 3,000-foot (910 m)vertical rock formationin Yosemite National Park.</p>（文本）

（3）如果要使图像居中，在<img>前添加<div style=\"text-align:center\">,然后在它后面加上</div>。要使文本围绕图像，在<img>标签中添加style=\"float:right\"或style=\"float:left\"。最后，直接在<img>标签中使用vertical-align来控制图像与相邻图像或文本之间的垂直对齐。

（4）如果还有时间的话，可以试着将多个大小不同的图像以不同的垂直对齐设置进行组合。

#### 图像变为连接

如果要显示多幅图像，这样的图像就不是很理想。这种情况下，最好创建较小的缩略图，链接到每幅图像的较大版本。这样就可以将缩略图安排在一个网页上，用户可以比较方便地看到所有内容，即使他们看到的只是要显示的一些样本。缩略图是使用图像链接改善网页的一种方式。

格式<a href=”路径”><img src=“图片名称”/></a>

例如：<a href="http://www.flickr.com/photos/nofancyname/614287355/\"><img

src\"tunnelview\_sm.jpg\"alt="Tunnel View\"width="100"

height="80"/></a>

(1)<a>标签将这些特殊图像链接到更大的版本，在这个例子中这些图像保存在外部服务器（Flickr)。

(2)除非另做声明，否则Web浏览器在图像链接边缘显示一个彩色矩形。与文本链接一样，通常矩形在还没有访问链接前是蓝色的，而访问后变为紫色——除非你在样式单中指定不同颜色的链接。因为通常都不希望显示这个难看的边框，因此应在<img>标签中包含样式style="border-style:none"。在这个例子中，border:none成为img元素的样式单输入项的一部分，因为我们两次使用相同的样式。

#### 使用背景图像

background-color:指定元素的背景颜色。虽然不与图像相关，但是它是背景相关属性集的一部分。

background-image:指定作为元素背景的图像，语法如：url（“imagename.gip)。

background-repeat:指定图像重复的方式，可以是水平或者垂直方向。默认（不指定任何值）情况下，背景将在垂直和水平方向都重复。其他的选项为：repeat（与默认

值相同）、repeat-x（水平）、repeat-y（垂直）和no-repeat（图像仅出现一次）。

background-position:指定图像开始相对于其容器的位置。选项包括：top-left(左上）、top-center(上中）、top-right(右上）、center-left(中左）、center-center(居中）、center-right（中右）、bottom-left（下左）、bottom-center（下中）、bottom-right(右下），以及具体的像素和百分比位置值。

可将所有规格放到一个属性中

body{

background:#ffffff url('imagename.gif") no-repeat top-right;

}

#### 图像映射

定义：图像映射是一个能对链接指示作出反应的图形或文本框。单击该图形或文本框的已定义区域，可转到与该区域相链接的目标 (URL)。

作用：有时候，你可能想要将图像作为导航，但是超出在网站上常见的基于按钮或者链接的简单导航的范畴。例如，你可能有一个具有医药信息的网站，希望显示人体的图像，链接到提供关于身体各部分的信息的页面。或者你有一个提供让用户可以点击的世界地图，用于访问各国的信息。你可以将图像根据用户点击的位置分为链接到不同文档的区域。这被称为图像映射（imagemap)，任何图像都可以组成一个图像映射。

除非在很特殊的情况下，否则不需要使用图像映射。通常使用多幅图像，将它们排列在一起，每幅图像作为一个单独链接指向不同的页面，这会比较简单，也更高效。

要创建图像映射，首先需要知道图像内要变为可单击链接的每个区域的数字像素坐标。

这些可单击链接又称为区域（area)。创建图像映射需要几个部分的信息：坐标、目标URL、链接标题、链接替换文本。

(1)创建用于图像映射的html

<map name=”map name”>(任何图像映射以此代码开始)

<area shape=”rect”cords=”100,136,116,152”href=<http://www.libin.com/> alt=”李斌公司” title=“李斌公司”>

area属性：

shape：指定该区域的形状 rect（矩形）circle（圆形） poly（不规则多边形）

coords给出区域的像素坐标。对于矩形，给出左上角的x、y坐标以及右下角的x、y坐标；对于圆形，给出圆心的x、y坐标和半径，以像素为单位；对于多边形，按连

接顺序列出所有顶点的x、y坐标。

href指定区域链接的页面。可以像在普通<a hre>链接标签中那样使用任何地址或文件名。

alt 让你能够提供与该区域形状相关联的一小段文字。当用户将鼠标指向该区域时，

大多数浏览器（除Firefox)将显示一个小窗口。这些文字为不理解图像映射的用户

提供提示，虽然小却很重要。Firefox在提供提示时，正确地在alt 属性之外使用title

属性，这也就是为什么应尽量为图像同时提供两个属性。

图像映射中的每个可单击区域都必须描述为一个区域，也就是说典型的图像映射由一系列区域组成。编写完<area>标签后，图像映射的定义就完成了，因此最后加上</map>结束标签。

### 路径的介绍：（路径到不到会出现404）

如果你有许多文件，或者开始为网站中的内容建立一个组织结构，就应该将文件放到目录（或者“文件夹”）中，目录的名称反映其中的内容。例如，你的所有图像应该在“image”目录中，公司信息可以在“about”目录中等。不管你在自己的Web服务器上如何组织文档，都可以使用相对地址，它仅包含从一个页面找到另一个页面所需要的足够信息。

回忆一下第2章，你的Web服务器的文档根目录是Web内容的最高级目录。在Web地址中，文档根目录由前向斜杠（/)来表现。所有下级目录由相同类型的斜杠来分隔。

注：HTML总是使用斜杠“/”来分隔目录。不要使用反斜杠“”（这通常用在Windows)来分隔目录。记住，Web的一切都是向前发展的，所以使用前向斜杠。

分类：两类

绝对路径：提供了完整的URL和域名。

E:\迅雷下载\下载合集 。

<https://www.baidu.com/b.jpg>

相对路径：缺少完整域名的地址。一个文件相对于另外一个文件的位置。

相当于当前资源所在的位置。

三种写法：

1、html文件和图片在一个路径下，可直接写文件的名称。

<img src = “b.jpg” width=”300” height = “400” />

在html文件中，

2、在html中是用img文件下的4.jpeg（下层目录表示方法）

D:\code\ 4.html

D:\code\ img\4.jepg

在html中使用图片4.jpeg可已使用img\4.jepg来表示

<img src = “img\4.jepg” width=”300” height = “400” />

3、上层目录中的表示方法

D:\ code\4.html

D:\ 4.jpeg

图片在html的上层文件中

表示上层文件：../ 表示上层路径

<img src = “../4.jepg” width=”300” height = “400” />

表示上层的上层

<img src = “../../4.jepg” width=”300” height = “400” />

注：1、如果在网页中统一使用相对地址，将页面移到另一个文件夹、硬盘或Web服务器时，就不必修改链接。

2、提示：关于使用相对地址还是绝对地址，有一个通用规则，那就是在链接存储在一起的文件时应用相对地址，链接到其他地方——其他计算机、其他硬盘或Internet的其他网站上的文件时，应使用绝对地址。

3、链接到自己的网站中的页面和链接到外部Web内容的唯一区别是，链接到网站之外时，你必须包含内容的完整地址。完整的地址包括域名之前的http://，然后是文件的完整路径名如一个HTML文件、图像文件等）。当

### 超链接标签

标签：“anchor”，锚：使用超链接可以从一个页面跳转到另一个页面。a标签可以包含任意元素。

1、链接资源：存储网站的Web内容，最简单的方法是将所有的文件放在统一文件夹下，当文件以这种方式存储时，可用<a>标签来指定文件名链接他们。

<a href=”链接到资源的路径”>显示在页面上的内容</a>

属性：href：链接资源处的目标地址。相对路径或绝对路径。

target：指定了所链接的文件出现在名称为“窗口名”的框架窗口里。

例如：设置一个框架的name为right，使用target=”right”会跳转到right的框架里面，在right框架里显示。

设置的打开方式、默认方式是当前的页面打开

属性值 \_blank :在一个新窗口打开

\_self :在当前页打开。（默认方式）

例如：你可能希望在较小的辅助浏览器窗口里提供信息，但是允许用户仍然看到主窗口中的信息。这种情况往往发生在单击一个动画演示、电影片段或者其他多媒体元素时。你还可能在链接到网站外的内容时选择新的浏览器窗口。

<a href=”链接到资源的路径” target= “self”>显示在页面上的内容</a>

如果在超链接链接资源不写任何内容时，默认打开的是该html文件位置。

在“链接到资源的路径”处填入#号时 不会跳出网页（当前的链接不需要跳转到任何地方时，加个#）。

2、定位资源（链接到想到的资源处）

1. 想要定位一个资源，需要定位一个位置。

<a id = “top”>顶部</a>

1. 回到这个位置

<a href=“#top”>回到顶部</a>

引入一个标签 pre：原样输出(是什么形式就输出成什么形式)

<pre>输出的内容<\ID

3、<a href="#">回到顶部</a>#号默认为回到顶部，一般还没获取你所哟链接的资源的时候，你就要设置为#

4、<a href="#id">回到id所在的位置</a> #+id

html中有一个属性，每个元素都可以设置，该属性可以作为标签的唯一标识，这个属性就是id，id属性子啊同一个页面只能有一个，不能重复。

注：1、通过<a>标签的id属性指定的锚名称必须以数字或者字母开头。所以

如果你想要为锚的ID编号，一定要以文本开始（如photol、photo2等）而不要只是用1、2。纯数字的ID在浏览器中能正常工作但是不能被验证为有效的XHTML代码。

1. 创建指定其他页面上指定锚的链接。

例如：<a href=”http://www.libin.com/a/b/c.html#top”></a>

如果已经在[www.libin.com](http://www.libin.com)页面上你连接到子目录下的资源可以是：

<a href= /a/b/c.html#top”></a>

### 表格标签 ：可以对数据进行格式化，使数据更加清晰

属性：<table></table>表示表格的范围

属性：border：表格线、数值表示表格线的宽度

bordercolor：表格线的颜色

cellspacing：单元格之间的距离

width：表格的宽度

height：表格的高度

table:<caption></caption>表示表格的标题

table：<tr> </tr>：表示行

（默认水平左对齐，并垂直居中）

tr中可以设置属性align ：表示表格中内容的对齐方式：

属性值：left center right

tr中<td> </td>表示一行中的单元格（table date）

td中可以设置属性align：表示表格中内容的对齐方式：

属性值：left center right

tr中<th></th>默认的表示居中和加粗

注：一个单元格的标签不会影响其他单元格，表格外的标签也不会影响表格中的信息。

表格的写法

<table border = “1” bordercolor = “blue” cellspacing= “0” width=“400” height=“150”>（）

表格起始

<tr align=“center”> <td> </td> <td> </td> <td> </td> </tr> 表格的第一行

<tr> <td> </td> <td> </td> <td> </td> </tr> 表格的第二行

<tr> <td width=“50%”> </td> <td width=30%> </td> <td width=“20%”> </td> </tr> 表格的第三行 （第一列占用百分之50 第二列占用百分之30 第三列百分之20）

<tr> <th> </th> <th> </th> <th> </th> </tr> 表格的第三行

</table >表格结束

注：表格设置为100%时表格将于浏览器等宽

1、合并单元格

colspan 跨列合并操作 <td colspan=“3”>人员信息 </td>

rowspan 跨行合并操作

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |
|  |  |

行数为3

单元格：第一行：3

第二行：2

第三行：2

<td rowspan=“3”>人员信息 </td>

从上到下 从左到右 依次数其中的格子数

### 表单标签 (这个标签特别的重要)

作用：可以提交数据到某个公司的服务器这个过程可以是使用表单标签实现。

例如：注册用户、收集信息、反馈信息和为网站体重搜索工具。

<form></form>:定义一个表单的范围

form属性：action 提交到的地址 ，不写时默认为当前的页面

例如<form action = “路径”></form>

属性 Method:表单的提交方式(常用的两种)

Get和post的两者的区别？

1、get：请求地址栏会携带提交的数据

post中不会携带

2、get请求的安全级别较低

3、get请求有数据大小的限制，post没有限制的

Enctype:一般请求下不需要这个属性，作文间上传时需要设置这个属性。

### 输入项：可以输入或选择的内容的部分（输入项里需要有一个name）

大部分的输入项都可以使用 标签<input type=“输入类型的类型”/>

1. 普通的输入项：<input type = “text”/>
2. 密码输入项：<input type = “password”/>
3. 单选输入项：<input type = “radio”> radio是收音机的意思，老式的收音机是一排按钮，按下一个其他的都会弹起，类似于单选框
4. 的功能

例<input type = “radio”>男 <input type = “radio”> 女

此时男女的两个都可以选择。

必须有一个value值

其中还有name 属性，改属性在选项当中必须一样，可变为单选

例：<input type = “radio” name=“sex” value=“nan” checked=”checked”>男<input type = “radio” name=“sex” value=“nv” checked=”checked”> 女

默认选中：checked = “checked”

1. 复选输入项（多选） ： <input type= “checkbox”> 可以选择多个

其中还有name 属性，改属性在选项当中必须一样

必须有一个value值

默认选中：checked = “checked”

1. 文件输入项（文件上传），

<input type = “file”/>

1. 下拉输入项（不是在input输入项内）

<select name=“birth”>

<option value=“0”>请选择</option>

<option value=“1992” selected=“selected>1992</option>

<option value=“1993”>1993</option>

<option value=“1994”>1994</option>

</ select >

通过selected=“selected”来实现下拉选择框的默认选项

1. 文本域：<textarea colums=“10” rows= “10”></textarea>
2. 隐藏项(内容不会在页面上，但是存在与html文件里)

<input type=hidden”/>

1. 提交按钮

<input type = “submit”> 默认为“提交查询内容

<input type = “submit” value = “注册”/>

点击提交之后 参数的上传方式为 ？输入项name的值=输入的值&

参数类似于key-value的传递方式

不做任何设置是提交到当前页面

提交的演示：使用图片来提交。

<input type = “imag” src =“图片路径”/>

1. 重置按钮:回到输入的初始状态

<input type=”reset”/> 效果为 重置

<input type=”reset” value=“重置注册”/> 效果 重置注册

1. 普通按钮：

<input type = “button” value = “”/>

### html中其他的标签的使用：

<b>Web</b> 效果： **Web 加粗**

<u>Web</u> 效果： Web 下划线

<i>Web</i> 效果： *Web*  斜体

<s>Web</s> 效果：~~Web~~删除线

<pre></pre> 效果：原样输出。

<sub></sub> 效果：下标 如6<sub>666</sub> 6666

<sup></sup> 效果：上标 如6<sup>666</sup> 6666

<div></div> 效果：<div>good</div> <div>good</div> 自动换行

### span

<span></span> 效果：<span>good</span> <span>good</span>

元素占网页的大小为自身的大小，当超过网页的宽度后会自动换行。 span没有任何语义，其他元素都有默认属性，span专门选中文字，来设置样式。

### html的头标签的使用

Html由两部分组成 head和body 在head中的标签称为头标签

title：表示在标签上显示的内容

meta:设置页面相关的内容

实现请求重定向：

<meta http-equiv(跳转请求)=“refresh” content=“3(三秒钟)；url=跳转的地址”/> 规定时间内跳转道对应的地址

base：超链接的基本设置

<base target=\_blank/> body中的所有超链接打开都是重新打开页面

Link：可以引入一个外部文件，使用link标签可以引入css文件

设置网页的关键字

还有属性name：<meta name = “keywords” content = “HTML5，javaScript，java”/>

keyword 告诉content中的内容是关键字，而不是其他内容。meta中表现关键字这两个是必须的。keywords 是复数，他的关键字可以有多个。

还可以指定网页的描述：

<meta name = “description” content = “发布h5，js等前段相关的信息”/>

搜索引擎检索网页是，会同时间锁网页中的关键字和描述，但是这两个值不会影响页面在搜索引擎中的排名。

### 框架标签

<frameset>

Rows:按照行进行划分

Cols：按照列进行划分

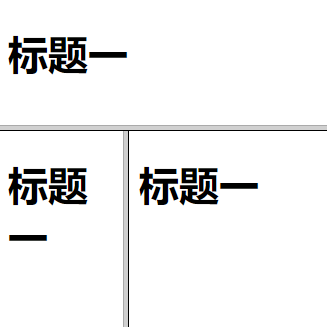
例：<frameset rows =”80,\*”> 两行第一行宽80，第二行为剩下的行宽

例：<frameset cols=“80，\*”> 两列第一列80 ，第二列为剩下的

<frame>：具体显示的页面

<frame name=”lowe\_left” (显示的页面名称) src = “b.html”（显示的页面路径）>

<frameset rows = "100,\*">

 <frame name="花花" src="a.html"/>

<frameset cols = "100,\*">

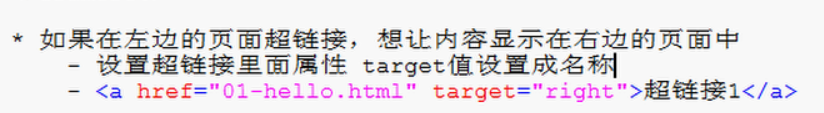
<frame name="left" />

标题二

<frame name="right" />

实现网盘效果：点击标题一中的超链接，在标题二中实现超链接的显示





使用框架标签不能是用body 只能把body去掉

### 邮件标签

发送电子邮件的超链接，点击链接以后可以自动打开计算机中默认的邮件客户端

<href = “mainto:邮件地址”>

<a href="mailto：449196838@qq.com">练习我们</a>

当点击这个连接时，会默认打开计算机的默认电子邮件客户端，并且收件人设为mainto后的邮件地址。

### em和strong标签

这两个标签都比哦啊好似一个强调的内容。

<em>标签用于表示一段内容的做重点，语气上的强调。em默认在浏览器中默认斜体。

<strong> strong标签用于表示一个内容的重要性。在浏览器中默认为粗体。

<i>i标签中的内容会以斜体显示。

<b>b标签的内容会以加粗显示。

h5规范中规定，对于不需要着重的内容而是单纯的加粗或者是斜体，就可以使用b和i标签。

### small和big标签

small标签中的内容会比他的父元素中的文字要小一些在h5中使用small标签来表示一些细则一类的内容比如：合同中小字，网站的版权声明都可以放到small

big基本淘汰了

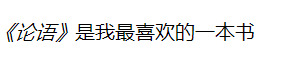
### cite标签

使用cite标签可以指明对某内容的引用或参考。例如，戏剧、文章或图书的标题，歌曲、电影、照片或雕塑的名称等。网页中所有的加书名号的内容都可以使用cite标签，表示参考的内容。

<p>

<cite>《论语》</cite>是我最喜欢的一本书

</p>



17、q标签表示一个短的引用(行内引用（这个样式基本不用。

子曰：<q>学而时习之，不亦说乎！</q>//自动给q标签的内容加引号

18、blockquote标签表示一个长引用（块级引用）

<div>

<blockquote>

知乎经常用这个标签

</blockquote>

</div>

19、sup和sub

sup：设置上标 <p>2<sup>2</sup></p> 22

sub：设置下标 <p>2<sup>2</sup></p> 22

### del标签

使用del标签来表示删除的内容,del标签中的内容，会自动添加删除线。

### ins表示插入一个内容

<ins>天空</ins> 天空下面会默认有下划线

### code和pre：需要在页面中直接编写一些内容

code和pre：需要在页面中直接编写一些内容。

pre是预格式标签，会将代码中的格式保存，不会忽略多个空格，也是一个块元素。

code：专门用来表示代码，不会保留格式。

常用的做法：输出代码

<pre>

<code>

需要表示的代码

<code/>

<pre/>

# Css简介

层叠样式表(cascading Style Sheets)用来定义网页的显示效果。一层盖的一层，层次高的会覆盖层次低的。可以解决html代码对样式定义的重复，提高了后期样式代码的可维护性，并增加了网页的显示效果功能。Html标签的外观样式比较单一，颜色只有黑白，字体类型和大小无变化。总之一句话：Css将网页内容和显示样式进行分离，提高了显示功能。

层叠：一层一层的

样式表：很多的属性和属性

## CSS的语法规范

1.选择器用于指定CSS样式作用的HTML对象，花括号内是对该对象设置的具体样式。

2.属性和属性值以“键值对”的形式出现。

3.属性是对指定的对象设置的样式属性，例如字体大小、文本颜色等。

4.属性和属性值之间用英文“:”连接。

5.多个“键值对”之间用英文“;”进行区分。

可以用段落和表格的对齐的演示。

## css和html的结合方式：四中结合方式（css可以写到哪里）

### 内联样式(行内样式)

在每个html的每个标签上面都有一个属性 style，把css和html结合在一起。将样式编写到style 属性中，这种样式我们成为内联样式，内联样式只对当前的元素的内容起作用。内联不利于重复使用。这种样式与结构与表现耦合，这种方式不推荐使用。直接在标签中添加如下：

style=”属性名称：属性值;属性名称：属性值;样式值:样式名”

例如：<p style="color:red";font-size：540px;"> 锄禾日党务，旱地禾下土</p>

例如：<div style=”background-color:red;color:green;font-size：20px”>天道酬勤</div>

### 嵌入式(内部样式)

使用HTML的标签head实现<style>标签，写在和head里面。

格式: <style type = “text/css”>CSS样式内容</style>

type = “text/css”：文档的类型为CSS样式

注：1、红色部分写不写都可以，只是兼容老版本浏览器。

2、选择器：选择页面中的指定标签，例如p元素中设置

<style type = “text/css”>

P{页面中所有的p采用次样式//选择器：选择body中所有的p元素 Background-color:red;//字体的背景颜色

Color：red；字体颜色//属性：属性值；font-size:40px;

}

</style> //声明文档样式表结束

将样式表编写的style标签中，然后通过css选择器选中指定元素，然后可以同时为这写元素一起设置样式，这样可以使样式进一步复用，将样式表编写到style中，也可以使表现和结构进一步分离也是我们推荐使用的。

### 外部样式

在style标签里面适用语句：写一个样式，在不同的样式使用。将样式表写到外部的css文件。

格式：

html文件中：

<style type = “text/css”> //css的文本

@import url(div.css)

</style>

CSS文件中：

Div{

Background-color:red;//字体的背景颜色

Color：red；字体颜色

}   
 注：

1. url括号后面必须要用分好结束。如果导入进来的样式与页面定义的样式重复，以本页定义的样式为准。

### link引入

使用头标签link，引入外部的css文件。写一个样式，在不同的样式使用。将样式表写到外部的css文件，还可以将外部样式表编写到外部的css文件中，然后通过link标签来将外部的css文件引入到当前的页面中。link标签为单标签，且位置在head标签中。

格式：<link rel=“stylesheet” type=”text/css” href=”文件路径”/>红色为固定值。

注：css样式统一编写到外部样式表中，完全使结构和表现分离，可以使样式表可以在不同的页面中使用，最大前度的使样式可以服用，将样式统一写到样式表中，然后通过link标签引入，可以利用浏览器的缓存姜块用户放的速度，提高了用户体验，所以开发中，我们最推荐的使用方式。

## css的三大特性

### 继承性

像儿子可以继承父亲的遗产一样，在CSS中，祖先元素上的样式，也会被他的后代元素所继承，利用继承，可以将一些基本的样式设置给祖先元素，这样所有的后代元素将会自动继承这些样式。但是并不是所有的样式都会被子元素继承，比如：背景相关的样式都不会被继承。

1.并不是所有的属性都可以继承, 只有以color/font-/text-/line-开头的属性才可以继承 。

2.在CSS的继承中不仅仅是儿子可以继承, 只要是后代都可以继承 。

3.继承性中的特殊性 。

a标签的文字颜色和下划线是不能继承的。

h标签的文字大小是不能继承的。

4.可以为在body标签中设置一些样式，用于给网页添加一些共性信息。

### 层叠性

层叠性：是浏览器处理冲突的一个特性。

1、如果一个属性通过两个选择器设置到同一个元素上面，那么这个时候一个属性就会将另一个属性层叠掉。

2、如果多个不同的属性通过两个选择器设置到一个元素上面，那么不会发生层叠。

### 优先级

由上到下、由外到内、优先级由高到低（，一般情况下后加载的优先级高）

## 选择器（选择符）：

**需求：**要使用css对HTML页面中的元素实现一对一，一对多或者多对一的控制，这就需要用到CSS选择器。(共享样式)

格式：选择器名称{属性名：属性值 ；属性名 属性值………..}

### 元素(标签)选择器：

标签选择器：使用标签名作为选择器的名称

<style type = “text/css”> //css的文本

Div{//开始制定样式规则

Background-color:red;//字体的背景颜色

Color：red；字体颜色

}

P {

Background-color:green;//字体的背景颜色

}

</style>

<div>天地不仁以万物为刍狗<div>

<p>天地不仁以万物为刍狗<p>

### Class选择器

class 类选择器：通过calss属性值选中一组元素，且一个元素可以设置多个class

可以给每个HTML标签都有一个属性class，多个值之间使用空格隔开。

<style type = “text/css”> //css的文本

div.haha{//开始制定样式规则

Background-color:red;//字体的背景颜色

Color：red；字体颜色

}

p.haha{

Background-color:red;//字体的背景颜色

Color：red；字体颜色

}

.haha{//只要类名相同的设置相同的属性值

Background-color:red;//字体的背景颜色

Color：red；字体颜色

}

</style>

<div class=”haha hello abc”>天地不仁以万物为刍狗</div>

//可以通过.haha .hello同时选择该元素。

<p class=”haha”>天地不仁以万物为刍狗</div>

### id选择器

id选择器：通过元素的id属性选中唯一的标签，id的值不能够重复，一个id对应一个元素，如果设置为相同的id，都会有效果。

定义：<div id=”01”>id选择器</div>

选择：

<style type = “text/css”> //css的文本

Div#1{

Background-color:red;//字体的背景颜色

Color：red；字体颜色

}

P#2 {

Background-coslor:red;//字体的背景颜色

Color：red；字体颜色

}

#1{//只要id相同的设置相同的属性值

Background-color:red;//字体的背景颜色

Color：red；字体颜色

}

</style>

<div id=”1”>天地不仁以万物为刍狗</div>

<p class=”1”>天地不仁以万物为刍狗</div>

注: 于是在这里就引申出了所谓的“唯一性”，它是针对JS而言的，当文档存在多个相同ID时，通过getElementById方法获取到的是在文档中第一个出现该ID的标签(DOM节点对象)，故都推荐ID在文档中最好只出现一次，即“唯一性”。

在实际开发中，也是遵循这样的原则，class用于CSS布局，id用于JS操作DOM节点对象。

### 交集选择器

选择器分组：通过选择器分组可以同时选中多个选择器对应的元素。多个选择器之间使用逗号相连接。

语法：选择器1，选择器2，选择器 N{}

例如：#p1 ,.p2,h1{} 选择class 为p1或id为p2或h1标签

### 交集选择器

交集选择器（复合选择器）：可以选中同时满足多个选择器的元素。

语法： 选择1 选择2 选择N{} 同时三组选择1、2、n 三个条件。

例如： div .p{}：选择<div>元素并且class为p的选择器。

注：对于id选择器来说不建议使用交集选择器，因为使用id选择器可以唯一确定一个元素。

### 通配符选择器

通配选择器：他可以用来选中页面中的所有的元素，

语法:\*{}

### 后代选择器

后代选择器：将E元素包含的所有后代F元素选出来。

格式：祖先元素 后代元素{} //可以写多个

div span{} 、

例如：#d1 span{} 选择id为d1，并且子元素为span的元素。

### 亲子选择器

亲子选择器：只能选择子元素,无法选择孙子元素。

格式：父元素>子元素{}

例如：div>span{} div元素下的span元素

.div1>.div2{} 选择class为div1的元素，并且这些元素的class为div2

注：ie6以下浏览器对此选择器不支持。

### 兄弟元素选择器：后一个兄弟元素选择器

作用：可以选中一个元素后紧挨着的指定的兄弟元素，中间不能隔其他的标签。

语法：前一个+后一个，

p+span{

background-color: goldenrod;

}

作用：选中后边的所有兄弟元素：

格式：前一个~后边所有

p~p{选中p元素后的所有怕元素。

background-color: goldenrod;

}

否定伪类：为所有p元素设置一个背景颜色为黄色，除了class值为hello的

作用：从已选中的元素中提出某些元素

语法：not(选择器)

p:not(.hello){

background-color: goldenrod;

}

### 三个选择器的优先级（class 用的较多）

class选择器大于标签选择器

iD选择武器大于class选择器

style选择器的优先级大于id

style>id选择器>class选择器>标签选择器

当使用不同的选择器，选中统一个元素是并设置相同的样式，这时样式之间产生了冲突。到低最终采用那个选择器定义样式，由选择器的权重决定。优选级高的优先显示。

内联样式，优先级 1000

id选择器，优先级100

类和伪类，优先级10

元素选择器 优先级1

统配选择器 优先级 0

继承的样式，没有选择器

当选择器中包含多种选择器是，需要将多种选择器的优先级相加然后在比较。但是注意，选择器优先级计算不会超过他的最大的数量级。如果选择的优先级一样，那么选择靠后的选择器。

加入 ！important 会提高优先级

复合选择器优先级

内部多种样式同时作用于一个元素，采取就近原则，行内样式的优先级最高，因为它例元素最近

### 属性选择器：

属性选择器1：通过属性来选择元素。

格式：[属性名字]{} 或 元素[属性名]{}

例如：[class]{} 选择具有属性class的元素。

p[class]{} 选择p标签，具有属性class的p元素。

属性选择器2：通过属性和属性值来选择元素。

格式：[属性名字=”属性值”]{} 或 元素[属性名字=”属性值”]{}

注：此处的属性值的双引号可有可无。

例如：[class=”class”]{} 选择具有属性class,并且属性值为class的元素。

p[class=class]{} 选择p标签，具有属性class,并且属性值为class的p元素。

### 伪类选择器

伪类元素选择器：伪类专门用来表示元素的一种的特殊的状态。

比如：访问过的超链接，比如普通的超链接，比如获取焦点的文本框。

当我们需要为处在这些特殊状态的元素设置样式时，就可以使用伪类。

浏览器是根据历史记录栏普阿暖是否为被访问过的。由于涉及用户的隐私问题，所以visited伪类只能设置字体的颜色。

比如超链接的状态

原始状态 鼠标放上去的状态 点击 点击之后（标题的颜色改变）

：link 没访问过的链接，正常的连接。

：hover : 伪类移入的状态，鼠标滑过的连接。

：active :超链接表示被点击时的状态，正在点击的链接。

：visited ：该链接被访问过后的状态，访问过的连接。

hover 和visited 也可支持其他元素的实质，ie6不支持对超链接以外的元素设置。

1 ：link

需求：为没访问过的链接设置一个颜色红

a :link{ 没访问的的连接设置为红色

color:red;

}

2 ：visited

需求：为访问过的链接设置为绿色。

a :visited {

backgrounf-color:green;

}

注：

1.浏览器通过历史记录来判断一个连接是否被访问过。

2.由于涉及到用户的隐私问题，所以使用visited伪类只能设置字体的颜色

3 ：hover

需求：鼠标移入到连接设置为蓝色。

a :hover{

backgrounf-color:blue;

}

4 ：active

需求：鼠标点击链接的状态

a : active{

backgrounf-color:red;

}  
 5 hover和active可以为其他一个元素设置伪类元素，IE6中不支持对超链接以外的元素设置hover和active

6.获取焦点：点击文本框后，闪动。

focus：获取焦点，当点input框，显示为黄色状态

input:focus{//选择文本框获取焦点。

background-color: yellow;

}

<input type="text"/>



7、:: selection(这个伪类元素有两个冒号)

selection:为p标签选中的内容使用样式。

 p::selection{//兼容打不分浏览器。

background-color: orange;

}

<p>我是一个状态</p>

火狐中使用该伪类要添加-moz-

p:: -moz-selection{//兼容火狐

background-color: orange;

}

注：这四个选择器的优先级是一样的，写的顺序和上边的一样

### 伪元素选择器：使用微元素来表示元素中的一些特殊位置

1、first-letter

需求：为p中的第一个字符设置一个特殊的字符

p:first-letter{

color: gold;

font-size：20px；

}

<p>我是一个状态</p> //p标签中的我变成黄色。

2、first-line

需求：为p中的第一行设置为黄色

p:first-line{

color: gold;

font-size：20px；

}

<p>我是一个状态</p> //p标签中的我变成黄色。

3、before

before: e表示元素最前边的部分一般before都需要结合content这个样式一起使用，通过content可以向before或after的位置添加一些内容。

p:before{

content: "我会出现在段落的最前边";//这部分内容在浏览器无法选中

}

<p > 这里是before，默认before这里什么都没有

我是一个状态

</p>



4after

after：结束标签的奇前边。

p:after {

content: "我会出现在段落的最前边";//这部分内容在浏览器无法选中

}

<p >

我是一个状态 //这里是after

</p>

### 子元素伪类选择器：

需求：为第一个p标签设置一个背景颜色为黄色

:first-child 可以选中第一个子元素（第一个子元素，是任何标签下第一个子元素）

:last-child：同理

:nth-child选中任意位置的子元素。该选择后边可以定制一个参数，指定第几个子元素。

even：表示偶数位置的子元素

odd：奇数位置的子元素。

案例一: 这是个p标签，并且，这个标签必须是第一个子元素。

p:first-child{//不写p相当与\*

background-color: olive;

}

案例二：div标签下的，p标签，并且这个p标签为第一个子元素。

div>p:first-child{//

background-color: olive;

}

案例三：body标签下的第三个子元素。

body:nth-child(3){//

background-color: olive;

}

案例四：p:last-child{//选中最后一个子元素，如果最后一个元素不是p元素，则不会有样式。

background-color: olive;  
 }

案例五：选中奇数位置的子元素。

body:nth-child(odd){// even

background-color: olive;

}

<p>我是一个段落</p> //该标签变颜色

<p>我是一个段落</p>

<p>我是一个段落</p>

<p>我是一个段落</p>

### 属性选择器：

作用:可以根据元素中的属性或属性值来选取指定的元素

语法：[属性名]先去含有指定属性的元素

[属性名=”属性值”]选取含有指定属性值得元素。

[属性名^=”属性值”]选取属性值以指定内容开头的元素。

[属性名$=”属性值”]选取属性值以指定内容结尾的元素。

[属性名\*=”属性值”]选取属性值以包含指定内容的元素。

### title属性

title：这个属性可以给任何标签指定当鼠标移入到元素上时，元素中的title属性的值将会作为提示文字显示。

需求：为所有具有title属性的p元素，设置一个背景颜色为黄色。并且title的值为hello。

p[title]{//p元素，并且元素的有属性为title。

background-color: yellow;

}

p[title = “hello”]{ 属性为title 并且属性值为hello

background-color: yellow;

}

p[title^= “he”]{ 属性为title 并且属性值为hello

background-color: yellow;

}

<p title="hello">我是一个段落</p> //只要title属性的都变为黄色

<p>我是一个段落</p>

### 选择器优先级

当使用不同的选择器，选中同一个元素时并且设置相同的样式时，这时样式之间产生了冲突，最终到底采用哪个选择器定义的样式，由选择器的优先级（权重）决定优先级高的优先显示。

优先级的规则：

优先级的规则

内联样式，优先级1000

id选择器，优先级100

类和伪类，优先级10

元素选择器，优先级1

继承的样式没有优先级。

当选择器包含多种选择器时，需要将多种选择器的优先级相加然后再比较。但是注意，选择器优先级计算不会超过它的最大的数量级，即类选择器相加之后最大为99不可能超过id选择器100。

当相同选择器的优先级一样，则使用后面的选择器。例如使用class选中同一个元素。

并集选择器的优先级是单独计算的。

内联样式的优先级最后，如果设置了内联样式，css文件无法对html修改。

！important：可以在样式的最后，添加一个！important，则此时该样式将会获得一个最高的优先级，将会优先于所有的样式显示甚至超过内联样式，但是在开发中尽量避免使用！使用important不利于维护，其他样式表会被覆盖。

background-color：red；！important

‘

# css的语法

### css的语法：

### css的注释

在css中不能使用html的注释和html中的代码 <!—啊啊啊 -->

css注释：/\* \*/ 它必须编写在style中。或者是css文件中。

### 选择器和声明块

选择器：通过选择器可以选中页面中的指定元素，并且将声明块的样式应用到选择器对应的元素中

声明块：声明块紧跟子啊选择器的后边，使用一对{}括起来，声明块中实际上是一组的名值对结构，这一组一组的名值对我们称为声明。在一个声明块中可以血多个声明，多个声明之间使用；隔开，声明的样式名和样式值之间使用：来连接

Div{//开始制定样式规则

Background-color:red;//字体的背景颜色

Color

}

学习css其实就是学习选择器和声明块

### 标签之间的关系

父元素：直接包含子元素的元素。

子元素：直接被父元素包含的元素。

祖先元素：直接或间接包含后代元素的元素。父元素也是祖先元素。

后代元素:直接或间接别祖先元素包含的元素。子元素也是后代元素。

兄弟元素：拥有相同父元素的元素。

### 内联元素和块元素

块元素：所谓的块元素就是会独占一行的的元素，无论他的内容有多少他都会独占一整行。；例如：p标签，h1标签 ，div标签

div元素：div这个标签没有任何语义，就是一个纯粹的块元素，并且不会为它里边的元素设置任何的默认样式，div元素主要用来对页面进行布局的。

特点：

1. 可以设置其宽度、高度，内外边距。

2.在不手动设置宽度的情况下，宽度默认为所在容器的100%（即容器宽度）。

3.可以容纳行内元素和其他块元素。

4.独占一行。

内联元素(行内元素)：所谓的行内元素，指的是只占自身大小的元素，不会占用一行。例如：a元素，img元素 ，span元素，ifram元素

span元素：如果想要对某段文字设置样式，此时可以使用span元素选中文本。span没有任何语义，专门用来选中文字，然后为文字设置样式。

特点:

1.总是和相邻的行内元素在同一行上（物以类聚）.

2.设置宽高无效，水平方向的padding和margin属性可以设置，但是垂直方向上的无效。

3.默认宽度是他自身内容的宽度。

4.行内元素只能容纳其他行内元素或者文本。

注：

1.块元素主要用来做页面的布局，内联元素只要用来选中文本设置样式。

2.一般情况下只使用块元素。

3.a元素可以包含任意元素，除了它本身。

4.元素不可以包含任何其他的块元素，但可以包含内联元素。

<p><span></span></p>这种方式是可以的。

<p><div></sdiv></p>这种方式不可以。浏览器会自动解析。

# 文本样式：

## 字体

### color字体颜色

作用：设置字体颜色，使用color来设置文字的颜色

格式：color：red；前景色

### font-size

作用：设置文字的大小，浏览器中一般默认的文字大小都是16px

格式：font-size设置的并不是文字的本身大小。每个文字都是处在一个看不见的框中，我们设置的font-size，实际上是设置的格的高度，并不是字体的大小。

一般情况下，文字要比这个格子小一些，也有时会比格大。

根据字体的不同，显示效果也不同。

通过font-family可以指定文字的字体、

font-family arial；当采用某种字体时，如果浏览器支持，则使用该字体，如果浏览器不支持，则使用默认字体。

### font-wight

## 十六精致颜色值

十六进制颜色格式为#rrggbb，其中rr、gg和bb是两位十六进制数值，分别表示颜色的红、绿、蓝部分。如果你不熟悉十六进制数字，不要着急。只要记住FF最大，00最小。在每个部分使用如下代码。–

FF意味着最大亮度。

CC意味着80%的亮度。

99意味着60%的亮度。

66意味着40%的亮度。

33意味着20%的亮度。

00意味着没有这个颜色成分。

## 行高

### 行高在html中的显示

在CSS并没有为我们提供一个直接设置行间距的方式，我们只能通过设置行高来间接的设置行间距，行高越大行间距越大使用line-height来设置行高。

行高类似于我们上学单线本，单线本是一行一行，线与线之间的距离就是行高，网页中的文字实际上也是写在一个看不见的线中的，而文字会默认在行高中垂直居中显示 。

所以，行间距= 行高-字体大小。

.p{

font-size:20px;

line-height:40px;//通过line-height可以简介的设置行高。

}

### 行高的属性值

1.直接就收一个大小

2.可以指定一个百分数，则会相对于字体去计算行高。

line-height：200%

3.可以直接传一个数值 ,则行高设置字体大小相应的倍数。

font-size:20px;

line-height:2；//40px

### 行高的作用

对于单行文本来说，可以将行高设置为和父元素的高度一致，这样可以使单行文本在父元素中垂直居中。

.box{

width:200px；

height:200px；

background-color:#bfa；

1ine-height:200px;//这里的行高与div的高度一样。所以居中。

### 在font中也可以指定行高

在字体大小后可以添加/行高，来指定行高，该值是可选的，如果不使用，则使用默认值。

例如: p{

font: 30px/100px "微软雅黑";

}如果

例如p{

line-height: 50px;

font: 30px "微软雅黑"; //这里会使用默认值,覆盖掉上边的行高，可以将行高放到下边。包括加粗，斜体，小型大写字母都有这个特点。

line-height: 50px;

}

## text-transform 设置文本的大小写

属性：text-transform

属性值：

none默认值，该怎么显示就怎么显示，不做任何处理

capitalize单词的首字母大写，通过空格来识别单词

uppercase所有的字母都大写

lowercase 所有的字母都小写

## 文本的对齐方式text-align

属性：text-align用于设置文本的对齐方式

属性值

1eft 默认值，文本靠左对齐

right,文本靠右对齐

center，文本居中对齐

justify,两端对齐。

原理：通过调整文本之间的空格的大小，来达到一个两端对齐的目的。

## 首行缩进text-indent

属性：text-indent

属性值：px ，em

注:1、当时用px赋值时，当在字体的大小变化是，缩进的大小还是一定的，因此一般会选用2em， 相当于当前字体的带下像素的两倍来设置。

2、当为它指定一个正值时，会自动向右侧缩进指定的像素。

如果为它指定一个负值，则会向左移动指定的像素。

作用：设置text-indent=-9999px该div的字体设置到看不见，但是该内容会被搜索引擎看到。

## 文本修饰(text-decoration)

属性：text-decoration可以用来设冒文本的修饰。

属性值：

none:默认值，不添加午何修饰，正常显示。

underline 为文本添加下划线。

overline为文本添加上划线。

line-through 为文本添加删除线。

注意：超链接会默认添加下划线，也就是超链接的text-decoration的默认值是underline

如果需要去除超链接的下划线则需要将该样式设置为none。

a{

text-decoration:none;

}

## 字符间距和单词间距

### 字符间距

属性：letter-spacing可以指定字符间距

属性值：0px默认为0px，

注：该属性对中英文都有效。

### 单词间距

属性：word-spacing 指定单词之间的间距。实际的原理是设置单词中间空格的大小。对中文没有什么意义。

属性值：0px默认为0px

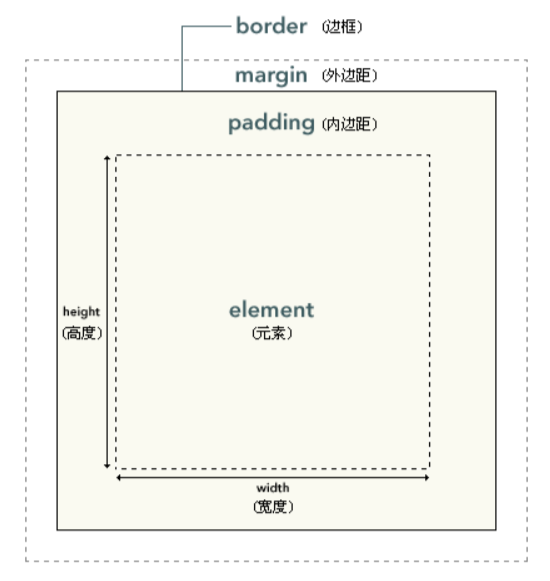
line-

# css中的盒子模型（块元素）：

## 盒子

CSS处理网页时，它认为每个元素都包含在一个不可见的盒子里。为什么要想象成盒子呢？因为如果把所有的元素都想象成盒子，那么我们对网页的布局就相当于是摆放盒子。我们只需要将相应的盒子摆放到网页中相应的位置即可完成网页的布局。

在进行布局前需要把数据封装到一块一块的区域内，这个区域的专业数据就叫做盒子，而且块元素会独占一行（一般用div表示这个区域）div不会对里面的元素进行任何的默认属性。就是个纯粹的块。



MARGIN：外边距，不同盒子之间的的间距。

BORDER:边框，盒子的宽度。

PADDING:内边距，盒子内容与外边距之间填充的部分。

content：内容区，盒子中的内容。

边距是元素之外的区域。它绝没有颜色，始终是透明的。

边框从元素的周围扩展，在填充之外。边框可以有多种类型、宽度和颜色。

填充存在于内容外围并且继承内容区域的背景颜色。

内容被填充包围。

### 可见框的大小

div#d2{

width：250px；

height:100px；

background-color:#ff0000；

border:5px solid #000000；

margin:10px；

padding：10px；

}

总宽度：宽度+左填充+右填充+左边框+右边框+左边距+右边距

总高度：高度+上填充+下填充+上边框+下边框+上边距+右边距

总宽度 300=200+10+10+5+5+10+10

总高度 100+10+10+5+5+10+10

## 边框（border）属性值

### 为边框必须设置的三个样式:边框宽度、边框颜色、边框样式

### 使用width设置盒子内容区的宽度。

width:100px;

### 使用height:设置盒子内容去的高度。

height:100px;

width和height只是设置了和内容区的大小，而不是盒子的整个大小。盒子框的大小由内容区，内边距和边框共同决定。

### 使用background-colo设置盒子背景颜色。

background-color:#bfa;//绿色

### 设置边框的宽度：修改该值会影响四边的值

例如：boder-width:10px;

例如：整体设置

border-width:10px 20px30px40px；//顺序，上下右做。

border-width:10px 20px 30px；//上，左右,下

border-width:10px 20px;//上下，左右

除初了border-width，CSS中还提供了四个border-xxx-widthxxx的值可能是top right bottom left专门用来设置指定边的宽度。

border统一设置

Border -top 上边框

Border- bottom 下边框

Border – left 左边框

Border – right 右边框

### 边框的颜色：

border-color:red;

同样也可以统一设置，边框宽度相同，同时color也提供了四个方向的样式，分别指定颜色。

格式：

border-color:red；

border-color:red yellow orange blue;

border-color:red yellow orange;

border-color:red yellow；

### 边框的样式border-style

属性：border-style:solid；

属性值：

none，默认值，没有边框

solid 实线

dotted 点状边框

dashed 虚线

double双线

style也可以分别制定四个边的边框样式，规则和width一致。同时他提供了border-xxx-style四个样式，分别来设置四条边。

### 整体设置border

边框的简写样式，通过它可以设置四个边框的样式，宽度，颜色，而且没有任何顺许要求。

例如：border：red solid 10px；

border-top：red solid 10px;//可以单独设置各个边的样式。

注：

1)border一指定就是四个边，不能分开指定各个边的样式。

2)也可用排除法设定，先设定整体，再使用 设定的新样式去覆盖原来的样式。

border：red solid 10px；

border-top：none;

3)border-style属性的默认值为none，这就是为什么除非将border属性设置为其他类型，否则元素没有边框的原因。最常见的border样式是solid和double。

## 内边距 padding

### 内边距

内边距：指的是盒子的内容区与盒子边框之间的距离。

设置Padding：四个外边距

Padding-top

Padding-bottom，

Padding-right

Padding-left

统一设置上下左右：

padding：12px 8px；元素上下距离边框为12px，左右距离边框为8px

padding：12px 8px 12px 8px；上右下左为12px 8px 12px 8px

注：1.内边距会影响可见框的大小。

2.元素的背景会延伸到内边距。

3.盒子的大小由内容去、内边距、和边框共同决定。

### 例子

<style type = “text/css”> //css的文本

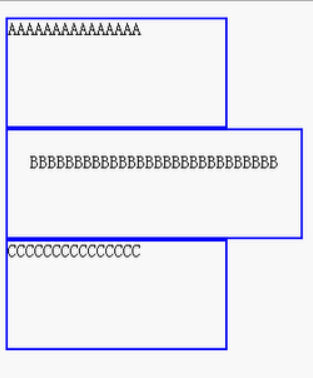
div{

width:200px;

height:s100px;

border：2px solid blue(粗细 直线 颜色)

}

#div21{

padding:20px;

}

#div22{

padding-left:40px;

}

</style>

<div id=”div21”>aaaaaaaaaaaaaa <div>

<div id=”div22>aaaaaaaaaaaaaa <div>

<div id=”div23>aaaaaaaaaaaaaa <div>

使用padding：20px 使用padding统一设置

也可以分别设置 有四条边

## 外边距

### 外边距

外边距指的是当前盒子与其他盒子之间的距离，他不会影响可见框的大小，而是会影响到盒子的位置。

### 外边距的设置

设置盒子四个方向的外边距：

margin-top

margin-right

margin-bottom

margin-left

统一设置外边距(顺势针)

margin：10px 20px 30px 40px; 上 右 下 左

margin：10px 20px 30px; 上 左右 下

margin：10px 20px; 上下 左右

margin：10px;

margin：0 auto; 上下0 ，左右居中。

注：1、外边距的值为px，有正值有负值，负值是向相反的方向移动。

2、magin还可以设置为auto，一般只设置水平方向的margin。如果只指定，左外边距或右外边距的margin为auto则会将外边距设置为最大值。

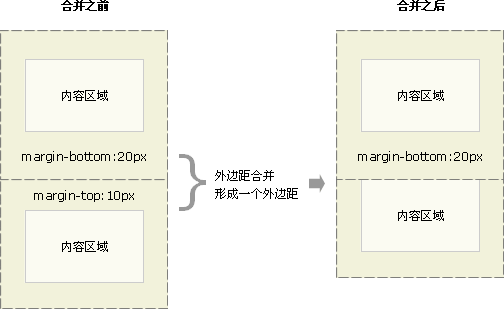
margin-left：auto;

margin-right：auto;

如果将1eft和right同时设置为auto，则会将两侧的外边距设置为相同的值，就可以使元素自动在父元素中居中，所以我们经常将左右外边距设置为auto以使子元素在父元素中水平居中。

### 垂直外边距重叠

重叠：在网页中垂直方向的相邻外边距会发生外边距的重叠。所谓的外边距重叠指兄弟元素之间的相邻外边距会取最大值而不是取和。



.box1{

width: 100px;

height: 100px;

background-color: red;

margin-bottom: 100px;

}

.box2{

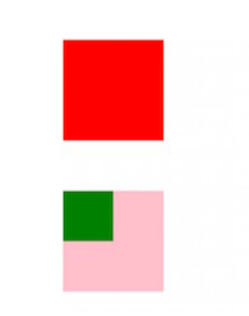
width: 100px;

height: 100px;

background-color: yellow;

margin-top: 100px;

}



例如：绿色div是粉色div的子元素，绿色没有上边距，此时他们的边框是同一个，也就是发生重叠现象，此时给绿色设置上边距，粉色会产生这个上边距，而绿色相对于粉色不会有上边距。

注意：1、要点：相邻的，中间不能有其他东西，并且是垂直的div。

2、上述代码为兄弟元素，如果父子元素的处置外边距相邻了，则子元素的外边距会设置给父亲。

例如：将子元素向下移动100px，相当于父元素移动。

|  |
| --- |
| .box3{  width: 100px;  height: 99px;  background-color: yellow;  border-top:1px ,  padding-top: 1px;  法一：将使用border将上边距隔开。  法二：使用padding将上边距隔开。  因为添加border或者是padding会影响到整个盒子的大小，所以会在height-1  法三：可以在父元素中设置上边距，会影响整个盒子的大小，再从height中减去相同的上边距。  }  .box4{  width: 50px;  height: 50px;  background-color: red;  margin-top: 100px;  } |

## 浏览器的默认样式

浏览器为了在页面中没有样式时，也可以有一个比较好的显示效果，

所以为很多的元素都设置了一些默认的margin和padding，而它的这些默认样式，正常情况下我们是不需要使用的。所以我们往往在编写春式之前需要将浏览器中的默认的margin和padding统统的去掉。

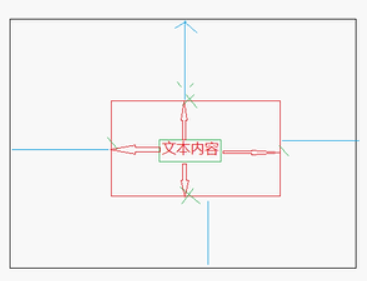
可以使用统配符\*去掉

\*{

margin: 0px;

padding: 0px;

}



Margin 可以统一设置也可分别设置

长度可以为,像素、点、ems或者是百分比

百分比：指占整个页面的大小为25%

margin-top：25px；

margin-right：25px；

margin-bottom：25px；

margin-left：25px；

顺序为从上开始，顺时针旋转，称之为TRBL其他属性亦可。

也可以统一设置为

margin：25px 25px 25px 25px； 相当于 margin：25px；

<style type = “text/css”> //css的文本

Div{

Width:200px;

Height:100px;

Border：2px solid blue ;(粗细 直线 颜色)

}

#div22{

Margin：20px

}

#div22{

Margin -left:30px;

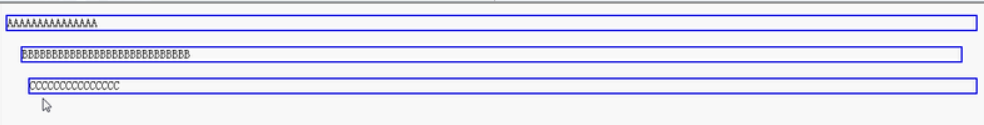
}

</style>

<div id=”div21”>aaaaaaaaaaaaaa <div>

<div id=”div22>aaaaaaaaaaaaaa <div>

<div id=”div23>aaaaaaaaaaaaaa <div>

div 边框的浏览器还是有一些边距，可加入以下代码消除

Body{

margin：0px

}

div与div之间仍然有边距，在算边距时要考虑到div的边框的像素，否则可能会有边框重叠。

附件：

## 内联元素的盒子模型

<span>这是一个span<span>

1、内联元素不能设置width和height,没有效果。

.box{

width: 100px;

height: 100px；

}

2、设置水平内边距，内联元素可以设置水平方向的内边距。

.box{

padding-left:100px ;

padding-right:100px ;

}

3、垂直方向内边距，内联元素可以设置垂直方向内边距，但是不会影响页面的布局。不会对其他的div产生影响，只是显示的效果变化了。

4、为内联元素设置边框，左右会影响，上下不会影响布局。

border：100px blue solid；//

5、为内联元素设置左右边距,这两个外边距不会重叠，而是求和。不能设置垂直的外边距。

margin-left：100px;

margin-right：100px;

## display

### display

display：可以通过display修改样式的属性。

可选值：

inline：将元素作为内联元素显示。

block：可以将一个元素设置为块元素显示。

inline-block:将一个元素转换为行内块元素

-可以使一个元素既有行内元素的特点又有块元素的特点，既可以设置宽高，又不会独占一行。例如img标签。

none：此元素不会被显示，并且元素不会在页面中继续占有位置。

使用该方式隐藏的元素，不会在页面中显示，并且不再占据页面的位置。

visbility：可以用来设置元素的隐藏和显示状态。

visbility:hidden：隐藏的元素虽然不会在页面中显示，但是它的位置会依然保持。

### 将超链接元素放大

因为超链接为内联元素，所以无法使用width和height，那么可以将内两元素变为块元素,成为块之后可以使用块中的样式。

a{

width: 100px;

height: 100px;

background-color: #bfa;

display: block;

}

<a href="#">这是一个超链接</a>

## overflow

### overflow

box1{

width: 100px;

height: 100px;

background-color: yellow;

}

.box2{

width: 100px;

height: 200px;

background-color: red;

}

子元素默认是存在于父元素的内容区中，理论上讲子元素的最大可以等于父元素内容区大小如果子元素的大小超过了父元素的内容区，则超过的大小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。

父元素是默认将溢出的内容在元素的外边显示。

### overflow

overflow：通过overflow可以设置父元素如何处理溢出内容：

可选值：

-visible，默认值，不会对溢出内容做处理，元素会在父元素以外的位置显示。

-hidden，溢出的内容，会被修剪，不会显示。

-scro，会为父元素添加滚动条，通过拖动滚动条来查看完整内容。

缺点：-该属性无论内容是否溢出，都会添加水平和垂直双方向的滚动条。

-auto，会根据需求自动添加滚动条，

需要水平就添加水平。

需要垂直就添加垂直。

都不需要就都不加。

### 效果

|  |  |
| --- | --- |
|  |  |
|  |  |

## 文档流

### 文档流

文档：指的就是一个html文档

html是由多层结构叠加起来的，文档流处在网页的最底层，它表示的是一个页面中的位置，我们所创建的元素默认都处在文档流中。素在文档流中的特点。

### 块元素的特点

<div style="width: 100px;height: 100px;background-color: #bfa;"></div>

<div style="width: 100px;height: 100px;background-color: #ff0;"></div>

1.块元素在文档流中会独占一行，块元素会自上向下排列

2.块元素在文档流中默认的宽度是父元素百分之百。默认值是auto

3.块元素在文档流中的默认为被内容撑开。子元素多大，他就多大。

### 内联元素的特点

1.内联元素在文档流中只占自身的大小，会默认从左向右排列。

2.如果一行中不不足以容纳所有的内联元素，则换到下一行。

3.内联元素的宽度高度都是被内容撑开。

当元素的高度或宽度的值为auto时，此时指定内边距不会影响可见框的大小，而是会自动修改宽度和高度，以适应内边距。



## 浮动

块元素在文档流中默认垂直排列，所以这个三个div自上至下依次排开，如果希望块元素在页面中水平排列，可以使块元素脱离文档流。使用float来使元素浮动，从而使元素分离。

### float

可选值：

none，默认值，元素默认在文档流中排列

1eft,元素会立即脱离文档流，向页面的左侧浮动

right,元素会立即脱离文档流，向页面的右侧浮动

当为一个元素设置浮动以后（float属性是一个非none的值），元素会立即脱离文档流，元素脱离文档流以后，它下边的元素会立即向上移动元素浮动以后，会尽量向页面的左上或这是右上漂浮浮，直到父元素的边框或者其他浮动的元素。

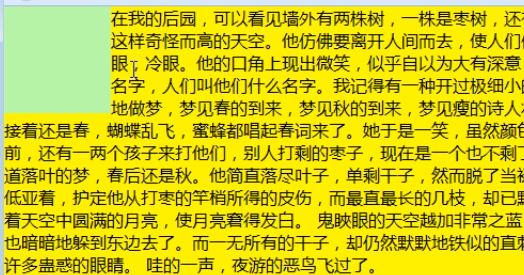
如果浮动元素上边是一个没有浮动的块元素，则浮动元素不会超过块元素。

当一行中无法容纳所有的浮动元素，那么浮动元素会换行处理。但是浮动的元素不会超过他上边的兄弟元素。

|  |  |
| --- | --- |
| .box1{  width: 200px;  height: 200px;  background-color: red;  }  .box2{  width: 200px;  height: 200px;  background-color: blue;  }  .box3{  width: 200px;  height: 200px;  background-color: green;  } | 块元素在文档流中默认垂直排列 |
| .box1{  width: 200px;  height: 200px;  background-color: red;  float：right;  } | float为right会飘到右上 |
| .box1{  width: 200px;  height: 200px;  background-color: red;  float：left;  } | box1向左上浮，会将box2给遮注，因为box2与box1的大小是相同的。 |
| .box2{  width: 200px;  height: 200px;  background-color: blue;  float:right;  } | 将box1和2都改为向右时，两个元素会一次排开。不会覆盖。 |
| .box3{  width: 200px;  height: 200px;  background-color: blue;  float:right;  } | 当只有box3块元素设置浮动时，它会排到box2的左上而不是box1的一行，应为一个块元素占一行。  如果想box3放到最上边，可以将他的代码方法box前边。或者将他设置浮动。 |
| box1向左浮  box2向左浮  box3向右浮  box3不会超过box2 |  |

### 浮动元素与文字

浮动的元素不会盖住文字，文字会自动环绕在浮动元素的周围，所以我们可以通过浮动来设置文字环绕图片的效果。



### 块元素脱离文档流

在文档流中，子元素的宽度默认占父元素的全部，子元素的高度为被别荣撑开的高度。当元素设置浮动以后，会完全脱离文档流。块元素脱离文档流以后，高度和宽度都被内容撑开。

### 内联元素图脱离文档流会变成文档流。

在文档流中设置width和heigth不能显示，但是在使用float脱离之后，可以使用。

## 布局

|  |
| --- |
| /\*清除默认样式\*/  \*{  margin: 0;  padding: 0;  }  /\*设置头部div\*/  .header{  /\*设置一个高度\*/  width:1000px;  /\*设置一个高度\*/  height:200px;  /\*设置一个背景颜色\*/  background-color: yellowgreen;  /\*设置居中\*/  margin: 0 auto;  }  /\*设置一个content\*/  .content{  /\*设置一个高度\*/  width:1000px;  /\*设置一个高度\*/  height:400px;  /\*设置一个背景颜色\*/  background-color: orange;  /\*设置剧中\*/  margin:10px auto;  }    /\*设置content中小div的样式\*/  .left{  width: 200px;  height: 100%;  background-color: skyblue;  /\*向左浮动\*/  float:left ;    }  .center{  width: 580px;  height: 100%;  background-color: yellow;  /\*向左浮动\*/  float:left ;  /\*设置水平外边距\*/  margin: 0 10px;  }  .right{  width: 200px;  height: 100%;  background-color: pink;  /\*向左浮动\*/  float:left ;  }    /\*设置一个footer\*/  .footer{  /\*设置一个高度\*/  width:1000px;  /\*设置一个高度\*/  height:150px;  /\*设置一个背景颜色\*/  background-color: silver;  /\*设置剧中\*/  margin: 0 auto;  }  <!--头部div-->  <div class="header"></div>  <!--主体内容div-->  <div class="content">  <!--一般命名为宽中的内容-->  <!--左边的div-->  <div class="left"></div>  <!--中间的div-->  <div class="center"></div>  <!--右边的div-->  <div class="right"></div>  </div>  <!--底部信息div-->  <div class="footer"></div>  效果： |

## 高度塌陷问题

### 高度塌陷

.box1{

border: 10px red solid;

}

这里没有设置div中的内容，显示为一条红线，为上下两个边界线的和20px。如果其中有div，则父元素被资源速撑开。

在文档流中，父元素的高度默认是被子元素撑开的，也就是父元素多大，子元素就多大。当为子元素设置浮动以后，子元素会完全脱离文档流，导致子元素无法撑起父元素的高度，导致父元素的高度塌陷。由于父元素的高度塌陷，，则父元素下的所有元素都会向上移动，这样将会导致页面布局混乱。所以在开发中要避免高度塌陷问题。

我们可以将父元素的高度写死，父元素的高度不能自动适应子元素的高度，所以这种方案是不推荐使用的。

### 解决高度塌陷

根据W3C的标准，在页面中元素都一个隐含的属性叫做Block Formatting Context简称BFC，该属性可以设置打开或者关闭，默认是关闭的。

当开启元素的BFC以后，元素将会具有如下的特性：

1.父元素的垂直外边距不会和子元素重叠

2.开启BFC的元素不会被浮动元素所覆盖

3.开启BFC的元素可以包含浮动的子元素

开启BFC

1、设置父元素元素浮动，

缺点：-使用这种方式开启，虽然可以撑开父元素，但是会导致父元素的宽度丢失，导致不能为父元素的父元素宽度，而且使用这种方式也会导致下边的元素上移，不能解决问题

2.设置元素绝对定位

3.设置元素为inpine-block

display：inline-block;可以解决问题，但是会导致宽度丢失，不推荐这种方式。

4、将元素的overflow设置为一个非visible的值

推荐方式：将overflow设置为hidden是副作用最小的开启BFC的方式。

overflow：hidden;

但是在IE6及以下的浏览器中并不支持BFC，所以使用这种方式不能兼容IE6。在IE6中虽然没有BFC，但是具有另一个隐含的属性叫做hasLayout,该属性的作用和BFC类似，所在IE6浏览器可以通过开hasLayout来解决该问题开启方式很多，我们直接使用一种副作用最小的：直接将元素的zoom设置为1即可。

zoom：1；

zoom表示放大的意思，后边跟着一个数值，写几就将元素放大几倍

zoom：1表示不放大元素，但是通过该样式可以开启hasLayout

所以在写的时候要把两种方式都写进去

zoom：1；

overflow：hidden；

### 高度塌陷最终解决方法

由于受到box1浮动的影响，box2整体向上移动了100px

我们有时希望清除掉其他元素浮动对当前元素产生的影响，这时可以使用clear来完成功能clear可以用来清除其他浮动元素对当前元素的影响

可选值：

none，默认值，不清除浮动

1eft,清除左侧浮动元素对当前元素的影响

right,清除右侧浮动元素对当前元素的影响

both，清除两侧浮动元素对当前元素的影响

清楚对它影响最大的大哥元素的浮动。

清除浮动以后，元素会回到其他元素浮动之前的位置

缺点：使用这种法师虽然可以解决问题，但是会在页面中添加多余的结构。W3c推荐的方法。

### 解决高度塌陷方案二

解决高度塌陷方案二：

可以直接在高度塌陷的父元素的最后，添加一个空白的div，由于这个div并没有浮动，所以他是可以撑开父元素的高度的，然后在对其进行清除浮动，这样可以通过这个空白的div来撑开父元素的高度，基本没有副作用。

|  |
| --- |
| 1、通过在html中添加clear元素来解决  .box1{  border: 1px red solid;  }  .box2{  width: 100px;  height: 100px;  background-color: blue;  float:left;/\*设置漂浮会高度塌陷\*/  }  /\*box2与clear为同级元素，box2漂浮后，clear可以撑起box1，前提是要清除box2漂浮后对clear的影响\*/  .clear{  clear:both;/\*清除两侧浮动元素对当前元素的影响\*/  }  <div class="box1">  <div class="box2"></div>  <div class="clear"></div>  </div>  2、通过after伪类，选中box1的后边。  .clearfix:after{  /\*添加一个内容\*/  content:"";  /\*转化为一个块元素\*/  display: block;  /\*清除两侧的浮动\*/  clear:both;  }  <div class="box1 clearfix">  <div class="box2"></div>  </div>  效果：  在IE6中不支持伪类，所以子在IE6中还需要使用haslayout来处理。  。clearfix{  zoom：1;  } |

## 布局的定位

### 定位

定位：定位指的是就是将

position：不能继承，并且静止的、静态的。通过position属性来设置元素的定位

可选值：

static:默认值，元素没有开启定位。如果定位设置为其，将不会有任何变化。

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（也是绝对定位的一种）

当开启了元素的定位（position属性值是一个非static的值）时，可以通过left right top bottom四个属性来设置元素的偏移量。

1eft:元素相对于其定位位置的左侧偏移量

right：元素相对于其定位位置的右侧偏移量

top:元素相对于其定位位置的上边的偏移量

bottom：元素相对于其定位位置下边的偏移量|

### 1、relative

1、相对的位置为相对于元素原来的位置。

2、相对定位的元素不会脱离文档流。

3、相对定位会提成一个层级，如果发生于其他元素的重叠，相对定位的元素在上边。

4、通常偏移量只需要使用两个就可以对一个元素进行定位，一般选择水平方向的一个偏移量和垂直方向的偏移量来为一个元素进行定位。

5、相对定位不会改变元素的性质，块还是块，内联还是内联。内联不能设置宽高，因为他还是在文档流中。

position：relative；//开启定位

left：100px;

|  |
| --- |
| .box1{  width: 200px;  height: 200px;  background-color: red;  }  .box2{  width: 200px;  height: 200px;  background-color: yellow;  position:relative;  left:200px;  top:200px;  }  .box3{  width: 200px;  height: 200px;  background-color:greenyellow;  } |

### 决对定位

1.开启绝对定位，会使元素脱离文档流。

2.开启绝对定位以后，如果不设置偏移量，则元素的位置不会发生变化。

3.绝对定位是相当于离他最近的开启了定位的祖先元素进行定位的。postion只要不是默认值或者是static。如果所有的祖先元素都没有开启定位，则会相对于浏览器窗口进行定位。

4.一般情况下，开启了子元素的绝对定位都会开启父元素的相对定位。

5.决定定位会使元素提升一个层级。当一个元素没有设置宽高，脱离父元素使会变空。

6.绝对定位会改变元素的性质，内联元素变成块元素。块元素的高度和宽度默认被内容撑开。

### 固定定位

当元素的position属性设置fixed时，则开启了元素的固定定位固定定位也是一种绝对定位，它的大部分特点都和绝对定位一样

不同的是：

1.位永远都会相对于浏览器窗口进行定位。

2.固定定位会固定在浏览器窗口某个位置，不会随滚动条滚动。

3.ie6不支持固定定位。

### 层级问题

1、如果定位的元素层级是一样的，则下边的元素会盖住上边的元素。这里指的是代码的前后顺序。

2、如果想使低层级盖住高层级的，可以使用z-index属性来手动设置元素的层级。

元素层级高的将覆盖底层级的。前提是，必须开启定位才能使用z-index。

3、父元素的层级再搞，也不会盖中子元素。

## 设置元素的透明（opacity）

### opacity

opacity可以用来设置元素的背景的透明。

范围：0-1。

1：完全透明。

0.5：半透明。

0：透明。

注：opacity 属性在IE8及以下的浏览器中不支持。

### filter

opacity属性在IE8及以下的浏览器中不支持。IE8及以下的浏览器需要使用如下属性代替

alpha(opacity=透明度）

透明度，需要一个0-100之间的值

0表示完全透明

100表示完全不透明

50半透明

例如：filter：alpha(opacity=50）

## 背景

### 设置背景图片

格式：background-color：url(相对路径)；

注：1.如果背景图片大于元素，默认会显示图片的左上角。

2.如果背景图片和元素一样大，则会将背景图片全部显示。

3.如果背景图片小于元素大小，则会默认将图片平铺以充满元素。

4. 可以同时为一个元素指定背景颜色和背景图片，这样背景颜色将会作为背景图片的底色。一般情况下设置背景图片时都会同时指定一个背景颜色。

简写：background：url() no-repeat center center

### 设置背景图片的重复方式

属性：background-repeat

属性值：

可选值：

repeat，默认值，背景图片会双方向重复（平铺）。

no-repeat，背景图片不会重复，有多大就显示多大。

repeat-x,背景图片沿水平方向重复。

repeat-y,背景图片沿垂直方向重复。

截取小的图片，使用平铺来达到效果。

### 定位背景图片的位置

通过background-position可以调整图片在元素中的位置。

可选固定位置：

该属性可以使用top right left bottom center中的两个值来指定一个背景图片的位置。

top left 左上

bottom right 右下

如果只给出一个值，则第二个值默认是center

可选任意位置：

也可以直接指定两个偏移量，

第一个值是水平偏移量

-如果指定的是一个正值，则图片会向右移动指定的像素

-如果指定的是一个负值，则图片会向左移动指定的像素

第二个是垂直偏移量

-如果指定的是一个正值，则图片会向下移动指定的像素

-如果指定的是一个负值，则图片会向上移动指定的像素

background-position:100px 180px;向右移动100px，向下移动180px。

background-position:-80px -40px;向左移动-40px，向上移动40px。

### 背景图片是否随页面滚动

background-attachment用来设置背景图片是否随页面一起滚动

可选值：

scroll：背景图片随窗口滚动。

fixed：背景图片会固定在某一位置，不随页面滚动。

fixed：

background-attachment：fixed；//广告

当背景图片的background-attachment设置为fixed时，背景图片的定位永远相对于浏览器窗口。

不随窗口滚定的图片一般设置给body，而不设置给其他元素。div会随滚动条消失，图片也会随着div的消失而消失。

### 使用图片作为超链接（雪碧技术）。

|  |
| --- |
| /\*做完功能以后，第一次切换图片的时候，会发现图片有一个非常块的闪烁。  \* 背景图片是以外部资源的形式加载进网页的，浏览器每加载一个外部资源就需要单独的发送一次请求，  但是我们外部资源并不是同时加载，浏览器会在资源被使用才去加载资源  我们这个练习，一上来浏览器只会加载link.png由于hover和active的状态没有马上触发，  所以hover.png和active.png并不是立即加载的  当hover被触发时，浏览器才去加载hover.png  当active被触发时，浏览器才去加载active.png  由于加载图片需要一定的时间，所以在加载和显示过程会有一段时间，  背图片无法显示，导致出现闪烁的情况。  为了解决该问题，可以将三个图片整合为一张图片，这样可以同时将三张图片一起加载，就不会出现闪烁的问题了，  然后在通过background-position来切换要显示的图片的位置，这种技术叫做图片整合技术（CSS-Sprite)  优点：  1将多个图片整合为一张图片里，浏览器只需要发送一次请求，可以同时加载多个图片，  提高访问效率，提高了用户体验。  2将多个图片整合为一张图片，减小了图片的总大小，提高请求的速度，增加了用户体验  \*/  .btn:link{  /\*将a转化为块元素\*/  display: block;  /\*这里根据图片的大小来设置宽高\*/  width: 93px;  height: 29px;  background-image: url(按钮/btn.png);  /\*设置背景图片不重复\*/  background-repeat: no-repeat;  }  /\*使用伪类切换不同的状态\*/  .btn:hover{  /\*当是hover状态时，希望图片向左移动，移动一个按钮的位置\*/  background-position: -93px 0px;  }  .btn:active{  /\*当是active状态时，希望图片向左移动，移动一个按钮的位置\*/  background-position: -186px 0px;  }  <a class="btn" href="#"></a>  D:\Webcode\day02\按钮\btn.png |

### 背景的简写属性

background通过该属性可以同时设置所有背景相关的样式 。

1.没有顺序的要求，谁在前谁在后都行。

2.也没有数量的要求，不写的样式就使用默认值。

background: #bfa url(img/3.png) center center no-repeat fixed;

|  |
| --- |
| /\*设置一个背景颜色\*/  /\*background-color: #bfa;\*/  /\*设置一个背景图片\*/  /\*background-image: url(img/3.png);\*/  /\*设置背景不重复\*/  /\*background-repeat: no-repeat;\*/  /\*设置背景图片的位置\*/  /\*background-position: center center;\*/  /\*设置背景图片不随滚动条滚动\*/  /\*background-attachment: fixed;\*/  background: #bfa url(img/3.png) center center no-repeat fixed; |

### 雪碧图

## 表格

表格在日常生活中使用的非常的多，比如excel就是专门用来创建表格的工具，表格就是用来表示一些格式化的数据的，比如：课程表、银行对账单。在网页中也可以来创建出不同的表格。table是一个块元素。

### 创建表格

|  |
| --- |
| 1、在HTML中，使用table标签来创建一个表格  <table border="1" width="40%" align="center">  //设置表格的边框的的宽度，表格宽度为浏览器的40%，位置为居中。  在table标签中使用tr来表示表格中的一行，有几行就有几个tr  2、使用caption标签设置表格的标题，且位置在最前面。  <caption>标题名称</caption>  3、<!-- 在tr中需要使用th创建一个表头，大多数的浏览器将表头中的内容设置为加粗居中。 >  <tr>  <th>A1</th>  <th>A2</th>  <th>A3</th>  <th>A4</th>  </tr>  <tr>  4、<!-- 在tr中需要使用td来创建一个单元格，有几个单元格就有几个td -->  <td>A1</td>  <td>A2</td>  <td>A3</td>  <td>A4</td>  </tr>  </table> |

### 合并单元格。

1、首先要删除要合并的单元格。

2、横向合并还是纵向合并。

colspan横向的合并单元格

rowspan：用来设置纵向的合并单元格。

横向

|  |
| --- |
| <table border="1" width="40%" align="center">    <!--  在table标签中使用tr来表示表格中的一行，有几行就有几个tr  -->  <tr>  <!-- 在tr中需要使用th来创建一个单元格，有几个单元格就有几个th ，使用th来表示表头中的内容，它会有一些默认的效果，居中和加粗 -->  <th>学号</th>  <th>姓名</th>  <th>性别</th>  <th>住址</th>  </tr>    <tr>  <td>B1</td>  <td>B2</td>  <td>B3</td>    <!--  rowspan用来设置纵向的合并单元格  -->  <td rowspan="2">B4</td>  </tr>  <tr>  <td>C1</td>  <td>C2</td>  <td>C3</td>  </tr>  <tr>  <td>D1</td>  <td>D2</td>  <!--  colspan横向的合并单元格  -->  <td colspan="2">D3</td>  </tr>    </table> |

### 表格的样式

width：表格的宽度

height：表格的高度

border：表格的边框

border-spacing：表格之间的空隙。

cellspacing：表格之间的空隙。

border-collapse：collapse

设置表格线是否合并，没有表格空隙，合并后为一个像素

table{

width: 300px;

/\*居中\*/

margin: 0 auto;

/\*边框\*/

border: 1px solid yellowgreen;

/\*边框之间的距离,这里设置为之后是两个边框的合并，2个像素\*/

border-spacing: 0px;

/\*边框的合并,不给边框也可以，如果设置边框合并，border-spacing自动失效\*/

border-collapse:collapse ;

}

### 设置表格的背景颜色

1.隔行变色

tr:nth-child(even){

//伪类:nth-child 表示选中tr的子元素td

//cenev表示偶数行，odd表示奇数行

background-color: greenyellow;

}

2.鼠标移动到哪行，哪行变颜色。

tr:hover {

background-color: greenyellow;

}

### 长表格

长表格：有一些情况下表格是非常的长的，这时就需要将表格分为三个部分，表头，表格的主体，表格底部。

在HTML中为我们提供了三个标签：

thead 表头。

tbody 表格主体。

tfoot 表格底部。

这三个标签的作用，就来区分表格的不同的部分，他们都是table的子标签，都需要直接写到table中，tr需要写在这些标签当中。

thead中的内容，永远会显示在表格的头部。

tfoot中的内容，永远都会显示表格的底部。

tbody中的内容，永远都会显示表格的中间。

如果表格中没有写tbody，浏览器会自动在表格中添加tbody，并且将所有的tr都放到tbody中，所以注意tr并不是table的子元素，而是tbody的子元素

通过table > tr 无法选中行 需要通过tbody > tr

|  |
| --- |
| <thead>  <tr>  <th>日期</th>  <th>收入</th>  <th>支出</th>  <th>合计</th>  </tr>  </thead>    <tfoot>  <tr>  <td></td>  <td></td>  <td>合计</td>  <td>100</td>  </tr>  </tfoot>  <tbody>  <tr>  <td>10.24</td>  <td>500</td>  <td>300</td>  <td>200</td>  </tr>  <tr>  <td>10.24</td>  <td>500</td>  <td>300</td>  <td>200</td>  </tr>  <tbody> |

### 表格的布局（已经被淘汰了）

1.以前表格更多的情况实际上是用来对页面进行布局的，但是这种方式早已被CSS所淘汰了。

2.表格的列数由td最多的那行决定。

3. 表格是可以嵌套，可以在td中在放置一个表格。

|  |
| --- |
| <table border="1" width="100%">  <tr height="100px">  <td colspan="2"></td>  </tr>  <tr height="400px">  <td width="20%"></td>  <td width="80%">  <table border="1" width="100%" height="100%">  <tr>  <td></td>  </tr>  <tr>  <td></td>  </tr>  </table>  </td>  </tr>  <tr height="100px">  <td colspan="2"></td>  </tr>  </table> |

### 完善clearfix

子元素和父元素相邻的垂直外边距会发生重叠，子元素的外边距会传递给父元素

解决方法：使用空的table标签可以隔离父子元素的外边距，阻止外边距的重叠。

在父元素的第一行添加<table></talbe>但是结构添加了没有用的部分。

解决方法：在css中添加table

1. display:table可以将一个元素设置为表格显示

.box1:before{

content: "";

display: table;

}

2.最终解决方法：clearfix

.clearfix:before,

.clearfix:after{

content: "";

display: table;

clear: both;

}

<div class="box3 clearfix">

<div class="box4"></div>

</div>

<div class="box1 clearfix">

<div class="box2"></div>

</div>

## 表单

### 表单

表单的作用就是用来将用户信息提交给服务器的。比如：百度的搜索框 注册 登录这些操作都需要填写表单。

### 创建表单

使用form标签创建一个表单，

<form action=" target.htm " method="post">

表单项

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

</form>

属性：

action：form标签中必须指定一个action属性，该属性指向的是一个服务器的地址，当我们提交表单时将会提交到action属性对应的地址。

method：该属性定义浏览器将表单中的数据提交给服务器处理程序的方式，该项的默认值为get请求。

第一，使用get方式提交表单数据，Web浏览器会将各表单字段元素及其数据按照URL参数格式附在<form>标签的action属性所指定的URL地址后面发送给Web服务器；由于URL的长度限制，使用get方式传送的数据量一般限制在1KB以下。

第二，使用post方式，浏览器会将表单数据作为HTTP请求体的一部分发送给服务器。一般来说，使用post方式传送的数据量要比get方式传递的数据量大；根据HTML标准，如果处理表单的服务器程序不会改变服务器上存储的数据，则应采用get方式（比如查询），如果表单处理的结果会引起服务器上存储的数据的变化，则应该采用post方式（比如增删改操作）。

### 表单项

使用form创建的仅仅是一个空白的表单，我们还需要向form中添加不同的表单项

1.input

使用input来创建一个文本本框，他的type属性是text。

例如：用户名<input type=”text” name=”sx”/>

查询字符串：sx =”sx”& hello=”sx”

type：指定表单元素的类型，它的不同值代表不同的类型，例如：

text代表文本框，button代表按钮，password代表密码输入框

radio代表单选框

name：

1.如果希望表单项的数据会提交到服务器中，还必须给表单指定一个name属性。name表示提交内容的名字。

2.用户填写的信息会附在url地址的后边以查询字符串的形式发送给服务器

url地址?查询字符串

格式：名值对结构

属性名=属性值&属性名=属性值&属性名=属性值&属性名=属性值

该值到达服务器以后，后台人员获取到属性名，以此来获取到属性值。并且可以有多个提交内容，内容之间用&符连接。

3.在文本框中也可以指定value属性值，该值将会作为文本框的默认值显示。不写的话默认值就是value值。

2.提交按钮

<input type= “submit” value=”注册”/>

1.提交按钮可以将表单中的信息提交个服务器，使用input创建一个提交按钮，他的属性时submit。

2.在提交按钮中可以通过value属性来指定按钮上的文字。

3.重置按钮reset

<input type= “reset” value=”注册”/> 清除所有表单的信息。

4.普通按钮

button类型只是一个普通按钮，而submit有一个提交的作用。在使用js代码的时候

3.密码框

密码<input type = “text”/>//这里是明文显示

密码<input type = “password” name = “password”/>只要想提交服务器，name属性必须有。

使用input创建一个密码框，他的type属性是password。显示为

4.单选按钮(选项是圆的)

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

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

<br /><br />

单选按钮

1.使用input来创建一个单选按钮，它的type属性使用radio。

2.单选按钮通过name属性进行分组，name属性相同是一组按钮，只能选择一个按钮。这样能达到单选的效果，。

3.像这种需要用户选择但是不需要用户直接填写内容的表单项，还必须指定一个value属性，这样被选中的表单项的value属性值将会最终提交给服务器。如果不玩value无法区分选择的哪一个，而gender 会显示on。

4.如果希望在单选按钮或者是多选框中指定默认选中的选项，则可以在希望选中的项中添加checked="checked"属性

5.多选框(选项是方的)

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

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

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

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

1.使用input创建一个多选框，它的type属性使用checkbox。

2.name属性进行分组，name属性相同的是同一组按钮。

3.value值表示锁提交的内容，多值之间用&连接。

4. 如果希望在单选按钮或者是多选框中指定默认选中的选项，则可以在希望选中的项中添加checked="checked"属性

6.下拉列表

下拉列表

|  |
| --- |
| 你喜欢的明星  <select name="star" multiple="multiple">  <!--  在select中可以使用optgroup对选项进行分组  同一个optgroup中的选项是一组  可以通过label属性来指定分组的名字  -->  <optgroup label="女明星">  <!-- 在下拉列表中使用option标签来创建一个一个列表项 -->  <option value="fbb">范冰冰</option>  <option value="lxr">林心如</option>  <option value="zw">赵薇</option>  </optgroup>  <optgroup label="男明星">  <!-- 在下拉列表中使用option标签来创建一个一个列表项 -->  <option value="zbs" selected="selected">赵本山</option>  <option value="ldh">刘德华</option>  <option value="pcj">潘长江</option>  </optgroup>  </select> |

1.使用select来创建一个下拉列表。

2.下拉列表的name属性需要指定给select，而value属性需要指定给option。

3.可以通过在option中添加selected="selected"来将选项设置为默认选中。

4.当为select添加一个multiple="multiple"，则下拉列表变为一个多选的下拉列表。可以同时选中多个下拉边框的值。

6文本域

使用textarea创建一个文本域，可以用css来调整。

自我介绍 <textarea name="info"></textarea>

7.重置(重新回到默认值)

<input type="reset" />

可以创建一个重置按钮，点击重置按钮以后表单中内容将会恢复为默认值

8.单纯的按钮

<input type="button" value="按钮" />

使用input type=button可以用来创建一个单纯的按钮，这个按钮没有任何功能，只能被点击。

通过js绑定一个事件，可以响应js代码。一般结合js代码使用。

9.button按钮

<button type="submit">提交</button>

<button type="reset">重置</button>

<button type="button">按钮</button>

这种方式和使用input类似，只不过由于它是成对出现的标签使用起来更加的灵活，可以使用图片作为按钮。

10.提示文字label

作用：可以结合input元素使用，当他的for属性和某个按钮的id值相同的时候，点击该label元素，可以选择到inout。

<label for="um">用户名</label>

直接点击用户名就可以输入信息。

<input id="um" type="text" name="username" />

在html中还为我们提供了一个标签，专门用来选中表单中的提示文字的label标签，可以为该标签添加样式。该标签可以指定一个for属性，该属性的值需要指定一个表单项的id值。

## 框架集

框架集和内联框架的作用类似，都是用于在一个页面中引入其他的外部的页面，框架集可以同时引入多个页面，而内联框架只能引入一个，在h5标准中，推荐使用框架集，而不使用内联框架。

使用frameset来创建一个框架集，注意frameset不能和body出现在同一个页面中。所以要使用框架集，页面中就不能使用body标签。所以在该页面中不能在写其他东西，只能引入一个或多个页面html页面。

属性：

rows，指定框架集中的所有的框架，一行一行的排列

cols， 指定框架集中的所有的页面，一列一列的排列

这两个属性frameset必须选择一个，并且需要在属性中指定每一部分所占的大小。

为什么不推荐的使用frameset

1.frameset和iframe一样，它里边的内容都不会被搜索引擎所检索，所以如果搜索引擎检索到的页面是一个框架页的话，它是不能去判断里边的内容的，

2.使用框架集则意味着页面中不能有自己的内容，只能引入其他的页面，而我们每单独加载一个页面，浏览器都需要重新发送一次请求，引入几个页面就需要发送几次请求，用户的体验比较差。

3.如果非得用建议使用frameset而不使用iframe

|  |
| --- |
| <frameset cols="30% , \* , 30%" frameborder=”no” 去除框架边框>\*表示剩下给中间。  <!-- 在frameset中使用frame子标签来指定要引入的页面 引入几个页面就写几个frame  -->  <frame src="01.表格.html" />  <frame src="02.表格.html" />引入两个页面之后要设置摆放方式。  <!-- 嵌套一个frameset -->  <frameset rows="30%,50%,\*">  <frame src="04.表格的布局.html" />  <frame src="05.完善clearfix.html" />  <frame src="06.表单.html" />  </frameset>  </frameset> |

## IE6 png的修复

在IE6中对图片格式png24支持度不高，如果使用的图片格式是png24，则会导致透明效果无法正常显示。

解决方法：

1.可以使用png8来代替png24，即可解决问题，但是使用png8代替png24以后，图片的清晰图会有所下降。

2.使用JavaScript来解决该问题，需要向页面中引入一个外部的JavaScript文件，然后在写一下简单的JS代码，来处理该问题。

DD\_belatedPNG\_0.0.8a.js //13kb 用在开发中，程序员去看代码。

DD\_belatedPNG\_0.0.8a-min.js//7kb 用来源代码中。

上边的两个文件的内容实质是一样的，但是格式不一样，所以大小也不一样，min版本去除了格式，代码只显示一行。

3、先将该文件复制到js文件夹下。

4、在body标签的最后引入外部的JS文件

|  |
| --- |
| <!-- 在body标签的最后引入外部的JS文件 -->  1、先引入外不的js代码。  <script type="text/javascript" src="js/DD\_belatedPNG\_0.0.8a-min.js"></script>  <!--再创建一个新的script标签，并且编写一些js代码 -->  <script type="text/javascript">  //对象.方法，修复选择器中的内容，写div，则修复div标签中的png  DD\_belatedPNG.fix("\*"); 全部  DD\_belatedPNG.fix("div，img");一个一个写。  </script> |

## hack

有一些情况，有一些特殊的代码我们只需要在某些特殊的浏览器中执行，而在其他的浏览器中不需要执行，这时就可以使用CSS Hack来解决该问题。CSS Hack实际上指的是一个特殊的代码，这段代码只在某些浏览器中可以识别，而其他浏览器不能识别，通过这种方式，来为一些浏览器设置特殊的代码。

### 条件hack

条件Hack，条件hack只对IE浏览器有效，其他的浏览器都会将它识别为注释，IE10及以上的浏览器已经不支持这种方式。

<！--以下内容只会出现在IE6中，其他浏览器中不会显示-->

<！--[if IE6]>

<p>为了您和家人的健康，请远离IE6！！</p>

<！[endif]-->

<！--[if IE8]>

<p>当前浏览器是IE8！</p>

<！[endif]-->

在Ie9以下的浏览器显示。 lt 小于 gt大于 lte小于等于 gte大于等于

1、项目汇总的做法

单独给浏览器设置样式，创建两个css文件。

<link red=”stylesheet” type=”text/css” href=”css/style.css”/>

<!—[if IE8]>

<link red=”stylesheet” type=”text/css” href=”css/style-ie8.css”/>

<!—[endif]>

### 属性级hack

假设在IE6中需要将背景颜色设置为黄色才能达到和其他浏览器同样的效果。

在样式前面添加下划线，则样式只有IE6级以下的浏览器才能识别。

\_background-color:yellow;

在样式前面添加下\*，则样式只有IE7级以下的浏览器才能识别。

\*background-color:yellow;

在样式最后添加一个\0只有IE8及以上的浏览器才能识别。

\*background-color:yellow\0;

Css Hack不到万不得已的情况尽量不使用，维护较难。

## 在网页中使用多媒体

在将多媒体包含到网站之前，要记住不是所有用户都具有能够播放你的媒体类型的设备，也不是所有用户都有快速传输大文件的宽带连接。始终警告你的访问者：他们所点击的链接将带给他们多媒体文件，并且提供给他们查看或者聆听该内容的选择一—不要强迫他们使用这些文件。

### 链接多媒体文件

格式：<a href=”movie.mov”>click video</a>

当用户单击“click video”时,视频文件hockey.mov将传输到其计算机上。他安装的助手应用程序或插件将在文件下载完毕时自动启动。如果没有找到与AVI兼容的助手程序或插件，网页浏览器将为用户提供一个下载适当插件或将视频保存到硬盘中供以后浏览的机会。

1. 嵌入多媒体文件

<object>标签网页中嵌入各种多媒体的首选方式。用于代替一些Html源代码中人能看到的<embed/>标签。

将多媒体文件嵌入到页面中产生一组软件控制，使文件能够直接播放——不需要辅助窗口，也没有必要离开所在的页面。下面是使用<object>标签嵌入前面看到的池塘视频的代码：

<object classid="CLSID:6BF52A52-394A-11d3-B153-00C04F79FAA6"

width="320"height="305">

<param name="typ"value"videox-ms-wmv"/>

<param name="URL"value="pond.wmv"/>

<param name="uiMode"value="full"/>

<param name="autoStar"value="false"/>

</object>

属性：classid：最复杂的部分是<object>标签的classid属性，它设置为一个很长的字母数字编号。这个编号是Microsoft Windows 媒体播放器的全局ID（global ID），它告诉<object>标签将Windows媒体播放器嵌入网页中以播放视频剪辑。你可以将这段代码复制并粘贴到自己的网页中。

width和height：<object>标签的width和height属性决定嵌入的Windows Media Player窗口的大小。如果没有设置这两个属性，有些浏览器将自动调整窗口的大小以适应内容，而有些将不显示内容一因此要安全地播放，应将这两个属性设置为多媒体内容播放时的大小。

<param>:负责指定关于如何播放剪辑的更多细节

属性：name和vlaue 负责将数据（value）同特定设置（name）关联起来。 type参数：指定要播放的媒体对象的标识符

MIME类型是唯一标识Internet上不同类型的媒体对象的标识符。MIME是Multipurpose

Internet Mail Extensions(多用途Internet邮件扩展)的缩写，之所以取这样的名字，是因为MIME类型原来是用于表示电子邮件附件的。这些MIME类型应在<object>标签的type属性中使用，用于标识data属性引用的多媒体对象类型。下列常见的声音和视屏格式。

WAV音频：audio/x-wav。

AU音频：audio/basic。

MP3音频：audio/mpeg。

MIDI 音频：audio/midi。

WMA音频：audio/x-ms-wma。

RealAudio:audio/x-pn-realaudio-plugin。

AVI：video/x-msvideo。

WMV:video/x-ms-wmv。

MPEG视频：video/mpeg。

QuickTime:video/quicktime。

1. 使用多媒体的技巧
2. 不要在页面中包含多媒体并且将它设置为页面装入时自动播放。始终让用户选择开始（和结束）音频或者视频。
3. 只要可能，让用户选择多媒体播放器。不要将自己限制在仅能用一种类型的播放器或者在一种操作系统上播放的多媒体内容。
4. 多媒体文件比典型的图像和文本文件大，这就意味着你必须在Web服务器上有存储它们的空间，请求它们也就需要传输带宽的分配。
5. 如果你的网站完全是音频或者视频，提供很少的文本或者图像，要理解你的一些目标用户因为他们的系统或者带宽的限制不能看到或者听到你所介绍的内容，为这些用户提供其他得到你的信息的选项。
6. 利用免费的在线视频宿主服务，比如You Tube(http://www.youtoube.com/)。You Tube不仅提供视频剪辑的存储，还将提供给你将视频插入你的网页所需的代码。图12.4所示为YouTube的“世界上最可爱的宠物”页面。如果你从图中所示的“Embed”区域复制并且粘贴文本，你将得到如下代码：

## 9、框架

## 1、建立框架集

框架集文档是一个HTML页面，指示Web浏览器将窗口分为几个框架，并指定每一框

架应显示哪个网页。框架集文档实际上并不包含内容，它只是告诉浏览器要加载哪些网页以及如何在浏览器窗口中排列这些网页。

1框架标签<frameset>

属性：rows框架下上相邻

Cols:框架左右相邻

属性值：像素，“\*“代表占据剩下的全部

当以像素为单位指定框架的大小，最好在相同框架集中包含至少一个可变\*宽度的框架。

例如<frameset rows=”50,\*”>

不能知道用户浏览网页时窗口的大小，一般使用百分比指定，

例如：<frameset cols=“20%,80%”>

2、单独框架<frame>

位置：包含在<frameset>中

属性：src:框架中加载的网页地址。如果是一个图像，也可以是图像的链接。

name:框架的名称，用来标识框架。也可用id。

scrolling：是否显示滚动条。

属性值：auto：需要时显示

yes：始终显示(不需要)

frameborder：0

framespacing：

1. 框架和窗口之间的链接：

框架代码：<frameset>

<frame src=”top.html” name=”top”/>

<frame src=”home.html” name=”main”/>

<frameset/>

网页代码

<a style=”color:#ffffff” href=”services.html” target=”main”>service</a>

## 2、内联框

# JAVASCRIPT

## JavaScript

### js

JavaScript ：基于对象和事件驱动的脚本语言，应用于客户端。

基于对象：提供了很多对象，可以直接使用。

事件驱动：html的网站静态效果。Javasript动态效果

客户端：专门治的是浏览器

### Js的特点

交互性（信息的动态交互）

安全性（js不能访问本地磁盘的文件）

跨平台性（只有能够致辞js的浏览器都可以运行）

### Javasript和java的关系（没有任何关系）

1.java：sun公司开发，现在为orcle 公司。Js是网景公司

2.javasript是基于对象的，java是面向对象的。

3.java是强类型语言，js是弱类型语言

例如：java 中 int I =10；

Js中 int i=10； int i=“10” int i= Boolean；

4.javaSript 只需解析就可以执行，而java需要县编译成字节码文件，在执行 。

### js的组成（三部分）

ECMAscript：由ECMA组织制定的js的语法，语句。

ECMA：欧洲计算机协会

BOM browser object modele 浏览器对象模型

DOM document object model 文档对象模型

### js和html的结合方式（两种）

1.使用标签

<script type =”text/javascript”>

js的代码

</script>

2.使用script标签映入一个外部的js代码

<script type=”text/javascript” src=“js文件的相对路径”>

引入js代码：<script type=”text/javascript” src=“js文件的路径”>

特点：可以是实现js代码的重复使用，避免了代码的冗余。

注：

1.使用第二种方式的时候，就不要再script标签里写js的代码了，不会执行。

2.引文js在html文档中是一门单独的语言，所以可以声名在文档中的任意位置，一般情况下声名在head标签中。页面先显示出来，再执行js代码，可以将js代码放到body标签后

### Js中的注释

1.单行注释 //注释内容

2.多行注释 /\* 注释内容 \*/

## js的原始类型和声明变量

### Java的基本数据类型

Js的原始类型

String 字符串类型 var str=“abc”；

Number 数字类型 var num=123；

Boolean 布尔类型 var flag = true；

var str ="123";

alert(str);

Object类型

Date 对象类型 Var date =new Date();

alert（typeof date）// Object类型

Null：k空对象将var声名为null，指定这个var只能引用对象。

获取对象的引用，null表示对象的引用为空，所有对象的引用都object。

Undifined:变量声名不赋值的情况下，默认值时Undifined。

var a;

alert(a);// undifined

typeof

提供了一个运算符 typeof 查看当签变量的数据类型

例如： alert（typeof（num）） 打印出 number

注：

1.js中的代码所有变量声明只有var关键字。

2.js中的变量名严格区分大小写的。

3.js中的字符串可以使用双引号和的单引号。

4.js中可以声名同名变量，不会报错，后声名的变量会将前面的覆盖。

5.java中变量的类型决定变量的使用方式，而js变量只有在赋值时才能决定变量的使用方式。两种的都时数据决定数据运算方式。

### Js变量的强换

1.使用Number(其他类型的数据);

string 类型的数字字符:对应的number数字

string 非数字字符:NaN(number类型)

boolean true: 1

boolean false:0

object 有具体的值:数字

object null: 0

2.其他类型转换为boolean 类型,使用Boolean(其他类型的数据);

number 正数或者负数：true

number 0:false

string 非空字符串 : true

string 空字符串 "":false

object 具体的对象 : true

object null : false

声明不赋值的变量: false

## 变量的运算

### 算术运算符

加法：+;

例如：var a = 123; var b = 234; alert(a+b);//357

减法：-

乘法：\*

除法：/

余数：%

number类型的和number类型的

number类型和boolean类型(true--1,false--0)

number类型和string类型(\*,/,%,-)

string类型和string类型的数字(\*,/,%,-)

string类型数字和boolean类型(\*,/,%,-)

在算术运算中如果两边的数据类型不是number的话，会使用Number()强转后在进行运算，字符串的加法会作为字符连接符，不会运算。在字符串中”+”符合代表的是字符串的连接符，不会参与运算。

### 逻辑运算

! 、 & 、&&、 | 、|| (与java中的一致);

var a=true;

var b=true;//false

alert(a&b);

if(a&b){//

alert("haha");

}

### 关系运算符

关系运算符：!=: >=:<=:>:<:

### 自增运算符

自增运算符：++，--，+=，-=

### ==和===运算符

等值运算符:==

先判断类型，类型一致则直接比较，类型不一致，则先使用Number()进行强转后再进行比较。

等同运算符:===

先判断类型，类型一致则再比较内容，内容一致则返回true，内容不一致则返回false。

注意：

1.null和undefined在做==判断时候返回true

2.当用==去比较基础类型和对象类型时，对象会调用valueOf()方法或者是toString方法。对象和对象类型比较比较的是地址符。

## Js的逻辑结构

### if

格式： if结构

单分支结构：

if(判断条件){执行体}

双分支：

if(判断条件){执行体}else{执行体}

多分支：

if(判断条件){执行体}else if(判断条件){执行体}……else{执行体}

2 swicth选择结构

格式： switch (a){

case "1":

alert("第一项选择");

break;

case "2":

alert("第二项选择");

break;

default:

alert("没有对应的选项");

break;

}

注：判断的变量可以是number类型也可以是string类型，但是不要混用。

### 循环结构

1、for(变量;条件;迭代条件){循环体}循环

例：九九乘法表

for(var i=1;i<=9;i++){

for(var j=1;j<=i;j++){

document.write(i+"\*"+j+"="+i\*j+"&nbsp;&nbsp;&nbsp;&nbsp;");

}

document.write("<br />");

}

2、while(循环条件){循环体}

3、do{循环体}while(循环条件)

## 数组：

### 数组的声明

var arr=new Array();//声明一个空数组对象

var arr=new Array(length)//声明一个指定长度的数组

var arr=[元素]//声明数组(最常用);<br />

注意：js中的数组声明不用指定长度，js的数组长度是不固定的，会随着元素的数量改变而改变。

### 数组的赋值和取值

数组可以存储任意类型的数据

数组名[角标]=值;//角标可以是任意的正整数或者是0

数组的取出：

数组名[角标]//返回当前角标对应存储的值

如果角标不存在，返回undefined;

### 数组的length属性

作用1：数组名.length//返回当前数组的长度。

作用2：数组名.length=新的值//动态的改变数组的长度

注意：length>原有长度，则使用空进行填充。

length<原有长度，则从后面进行截取，最后的数据会被删除。

4、数组的遍历

普通for循环：

for(var i=0;i<arr.length;i++){

alert(arr[i]);

}

for-in：

for(var i in arr){

alert(i);//获取的是角标

}

## Js的函数

作用:功能代码块的封装。减少代码的冗余。

### 函数的声明和调用：

方式一:function 函数名(形参名1,形参名2……){执行体}

function test1(a1,a2)

{

alert("函数声明一");

}

方式二:var 函数名=new Function("形参1","形参2"……,"函数执行体");

var test2=new Function("a1","a2","alert('函数声明二'+a1)");

注意：此声明表明在js中函数是作为对象存在的。

方式三:var 变量名=function(形参名1,形参名2……){函数执行体}

var test3=function(a1,a2){

alert("我是函数声明三");

}

调用函数：test2(1,2);

### 函数的参数

js中的函数在调用时，形参可以不赋值，不会报错

js中的函数在调用时，形参赋值可以不全部赋值，不会报错，但是实参会依次进行赋值。

### 函数的返回值

在js中如果函数有返回值则直接返回，没有返回值默认返回undefined

注意：

js的代码声明区域和执行区域是一起的，都是在js代码的代码域中。

### 函数的执行符:

在js中函数的执行符值()

没有小括号则函数名其实就是一个变量

加上小括号则会函数会被执行。

### 函数作为实参传递

在js中函数是可以作为实参进行传递的。

function testobj(a){

alert(a());

}

var testParam=function(){

alert("我是函数参数");

}

testobj(testParam());

## Js的类

### 类的声明

格式：function 类名(形参1，形参2……){

this.属性名1=形参1;

this.属性名2=形参2;

……

this.属性名=fn

}

### 类的调用

var 对象名=new 类名(实参1，实参2……);

注意：

js中类的内容只是对象的公共部分，每个对象还可以自定义的进行扩充。

### 类的继承：prototype关键字

同过prototype关键字实现了不同对象之间的数据共享。

作用1:实现某个类的所有子对象的方法区对象的共享，节省内存

## Js的对象

### js创建自定义对象

对象的作用：用来存储整体数据。

原因：原因：很多时候我们没有办法预先知道一个对象应该有哪些属性，所以只能临时的创建一个对象来自定义属性存储数据。来保证数据的完成性。

应用：Ajax中会使用。

创建自定义对象：

var 对象名=new Object();

对象名.属性名1=属性值1;

对象名.属性名2=属性值2;

对象名.属性名3=属性值3;

var 对象名={};

对象名.属性名1=属性值1;

对象名.属性名2=属性值2;

对象名.属性名3=属性值3;

2、一般用来存储数据，不会再自定义对象中存储函数对象。

3、js中的对象属性和内容是可以自定义的扩充的，不是依赖于类的声明的，类只是对象公共部分的一种声明，是为了节省代码的冗余的。

### 常用的对象和方法

String、Date、Math、Global

### String对象

1. 声明String对象存储字符串

var str=new String("abcdefg");//声明String对象存储字符串

var str2="MNP";//简写形式

2.字符串大小写的转换

alert(str.toUpperCase());//将字符串转换为大写

alert(str2.toLowerCase());//将字符串转换为小写

3.字符串的切割

var s="哈哈,嘿嘿,呵呵";

var s1=s.split(",");//按照指定的字符切割字符串，返回数组。

alert(s1.length);

4. 字符串的截取

var s="abcdef";

alert(s.substr(1,3));//从指定的开始位置截取指定长度的子字符串

alert(s.substring(1,3));//从指定的开始位置和指定的结束位置截取子字符串，含头不含尾。

5. 查找子字符串第一次出现的角标

var s="abcdefg";

alert(s.indexOf("dd"));//返回指定子字符串第一次出现的角标，没有则返回-1;

### Date对象

1、创建Date对象

var d=new Date();

2.获取当前年份

alert(d.getYear());//返回的是1900年开始距今的年分数

alert(d.getFullYear());//返回的是当前的年份

alert(d.getMonth()+1);//返回的当前月份的角标值，需要+1

alert(d.getDate());//返回的是当前的日期数

alert(d.getDay());//返回的是当前的星期数，但是周天会返回值为

alert(d.getHours());//返回当前时间的小时数

alert(d.getMinutes());//返回当前时间的分钟数

alert(d.getSeconds());//返回当前时间的秒数

### Math对象

Math在使用的时候不能new，使用Math.方法名调用即可

1.创建随机数字

alert("Math.random():"+Math.random());//返回0-1之间的随机数字，含0不含1。

alert(1000+Math.random()\*9000);

2.向下取整

alert(Math.floor(1000+Math.random()\*9000));

3.向上取整

alert(Math.ceil("12.34"));

4.四舍五入

alert(Math.round(12.12));

alert(Math.round(12.65));

5.数值比较:求取最小值，求取最大值

alert(Math.min(12,13,5,78));//获取最小值

alert(Math.max(12,3,4,56));//获取最大值

Global对象学习

该对象从不直接使用并且不能new，也是就直接写方法名调用即可。

1.使用eval将字符串转换为可执行的js代码

var str="var a=123";

eval(str);

alert(a);

2.使用isNaN判断是否值NaN

alert(isNaN("123"));

3.获取字符中的浮点数

alert(parseFloat("12.34a34a"));

## js的事件机制

解释：一般情况下当我们的行为动作满足了一定的条件后，会触发某类事务的执行。

作用：主要是结合js的函数来使用，通过事件机制调用js函数。

### 单双击事件

只要是单击或者可以触发js中的函数。

单击：onclick 当鼠标单击的时候会触发

<input type="button" name="" id="" value="测试单机" onclick="testOnclick()" />

双击：ondblclick 当鼠标双击的时候会被触发

<input type="button" name="" id="" value="测试单机" ondblclick="testOnclick()" />

### 鼠标事件

onmouseover 当鼠标悬停在某个HTML元素上的时候触发

<div id="showdiv" onmouseover="textOnmouse()">鼠标悬停事件</div>

onmousemove 当鼠标在某个HTML元素上移动的时候触发，游戏中鼠标的移动。

<div id="showdiv" onmousemove="textOnmouse()">鼠标悬停事件</div>

onmouseout 当鼠标在某个HTML元素上移出的时候触发

<div id="showdiv" onmouseout="textOnmouse()">鼠标悬停事件</div>

### 键盘事件

onkeyup 当键盘在某个HTML元素上弹起的时候触发，游戏中技能操作，wasd。

<input type="text" name="" id="" value="" onkeyup="testOnkeyup();"/>

onkeydown 当键盘在某个HTML元素上下压的时候触发,

<input type="text" name="" id="" value="" onkeydown="testOnkeydown()"/>

### 焦点事件

onfocus 当某个HTML元素获取焦点的时候触发

<input type="text" name="" id="" value="" onfocus="testOnfocus();"/>

onblur 当某个HTML元素失去焦点的时候触发

<input type="text" name="" id="" value="" onblur="testOnblur();"/>

### 页面加载事件

作用：

onload 当页面加载成功后触发。

### 注意:

js中添加事件的第一种方式：

在HTML上直接使用事件属性进行添加，属性值为所监听执行的函数。

js中添加事件的第一种方式：document 对象来执行

## windows对象

### BOM(Browser Object Model)

BOM浏览器对象模型：是规范浏览器对js语言的支持(js调用浏览器本身的功能)，提供了独立于内容而与浏览器窗口进行交互的对象。BOM的具体实现是window对象。

Window 对象表一个浏览器窗口或一个框架。在客户端JavaScript中，Window 对象是全局对象，要引用当前窗口根本不需要特殊的语法，可以把那个窗口的属性作为全局变量来用。在使用window对象的方法和属性的时候，可以不带window，比如window.alert(“韩顺平！”），可以直接写成alert（“韩顺平”）；

### windon对象

1、window对象不用new，直接进行使用即可，类似Math的使用方式，window关键字可以省略不写。

### 框体方法

1. alert:警告框，提示一个警告信息，没有返回。调用的是浏览器封装的方法。

调用：window.alert("我是警告框");

2 confirm:确认框，提示用户选择一项操作（确定/取消）

点击确定 返回true

点击取消 返回false

代码window.confirm("你确定要删除吗?"); 弹出提示框是否删除

3. prompt:提示框， 提示用某个信息的录入或者说收集。

点击确定，返回当前用书录入的数据，默认返回空字符串。

点击取消,返回null。

代码：window.prompt("请输入昵称：")；例如游戏中输入游戏名称等。

4.定时执行：setTimeout(code,millisec)

参数1：函数对象，参数2：时间，单位毫秒。返回值：返回当前间隔器的id

作用：方法用于在指定的毫秒数后调用函数或计算表达式。

代码：function testSetTimeout(){ //alert(会立即执行。)

window.setTimeout(function testdelete(){

alert(window.confirm("您确定要删除吗"));

},3000);

}

5间歇调用：setInterval (code,millisec)

参数1：函数对象，参数2：时间，单位毫秒。返回值：返回当前间隔器的id

作用：每间隔指定的时间执行指定的函数。

代码：function test(){

alert("间隔两秒执行");

}

function testInterval(){

window.setInterval(function test(){

alert("间隔两秒执行");

},2000);

}

<input type="button" name="" id="" value="2s之后执行" onclick="testInterval()"/>

6.clearTimeout(number id):

参数：定时器的id，此id可以在js中创建一个全局变量，变量接收来自setTimeout返回的id值。

作用：在还没执行前，将该定时器结束，

代码：function testClearTimeout(){

window.clearTimeout(idi);

}

7.clearInterval:用来停止指定的间隔器

参数：间隔器的id，需要创建全局变量来接收来自定时器的id。

作用：用来停止指定的间隔器

代码：function testClearInterval(){

window.clearInterval(ids);

}

### 子窗口方法

1、子页面方法

格式：window.open(pageURL,name,parameters)

参数pageURL为子窗口路径，name为子窗口句柄，parameters为窗口参数(各参数用逗号分隔)。

作用：打开一个新的窗口

function testOpen(){

window.open('son.html','newwindow','height=400, width=600,top=100px,left=320px,toolbar=yes,menubar=yes,scrollbars=yes,resizable=yes,location=no, status=yes');}、

2、window.close()

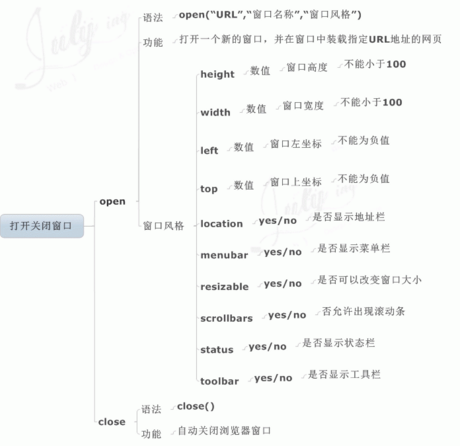
作用：关闭子页面，

注：关闭子页面的方法window.close(),但是此方法只能关闭open方法打开的子页面。

案例：倒计时功能

代码：

|  |
| --- |
| <script type="text/javascript">  /\* 倒计时功能\*/  function testTime(){  window.setInterval(function(){  var span = document.getElementById("timeSpan");  span.innerHTML = span.innerHTML-1;  if(span.innerHTML==0){  window.close();//实际是资源的跳转。  }  },1000);  }  </script>  <body onload="testTime()"> <!--当页面加载成功后触发该子页面。-->  <br />  <b>欢迎来到子页面，<span id="timeSpan">5</span>秒后自动关闭</b>  </body> |



3、子页面调用父页面的函数

格式：window.opener.父页面的函数

作用：当子页面添加完信息之后，点击确认，重新加载父页面的资源，显示更新的内容。

代码：

4、js的window对象的常用属性

windows对象还有其他对象，其他的属性。

地址栏相当于一个文本框，

1.地址栏属性:location对象

1、window.location.href="新的资源路径（相对路径/URL）"

让当前页面的资源跳转。

修改当前资源路径的三种方式

(1)直接修改浏览器地址。

(2)超链接

(3)通过js来动态的改变资源地址。

2、window.location.reload()重新加载页面资源

功能：刷新此页面

3、历史记录属性

window.history.forward() 页面资源前进，历史记录的前进。

作用：跳转到历史记录的下个页面

window.history.back() 页面资源后退，历史记录后退

作用：跳转到历史记录的上个页面

window.history.go(index)

作用：跳转到指定的历史记录资源

注意window.history.go(0)相当于刷新。

4、屏幕属性

window.srceen.width;//获取屏幕的宽度分辨率

window.screen.height;//获取屏幕的高度分辨率

作用：根据浏览器的大小设置屏幕的大小。

function test(){

var widhth = window.screen.width;

var height = window.screen.width;

alert(widhth+":"+height);

}

5、浏览器配置属性

window.navigator.userAgent

获取浏览器的属性

function testNa(){

alert(window.navigator.userAgent);

}

显示内容：Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.12 Safari/537.36

6、主体面板属性(document)

对象里面分装对象，将html中的所有标签都变为对象，父节点封装子节点。

## document对象

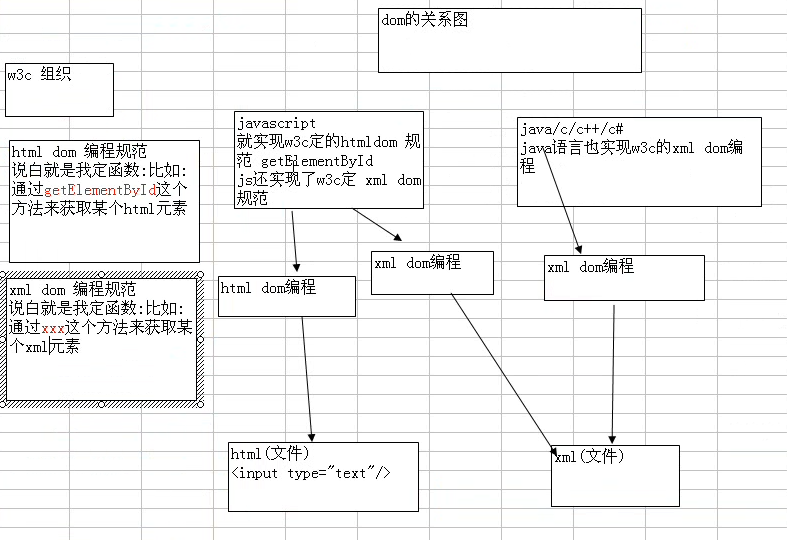
### document

需求：

DOM：Document Object Model(文本对象模型），根据W3C DOM规范，DOM是HTML与XML的应用编程接口（APl），DOM将整个页面映射为一个由层次节点组成的文件。

js把浏览器，网页文档和网页文档中的html元素都用相应的内置对象来表示，这些对象与对象间的层次关系称为dom，针对网页（htm1，jsp，php，aspx.net)的dom就是htmldom。我们这讲的就是html dom。

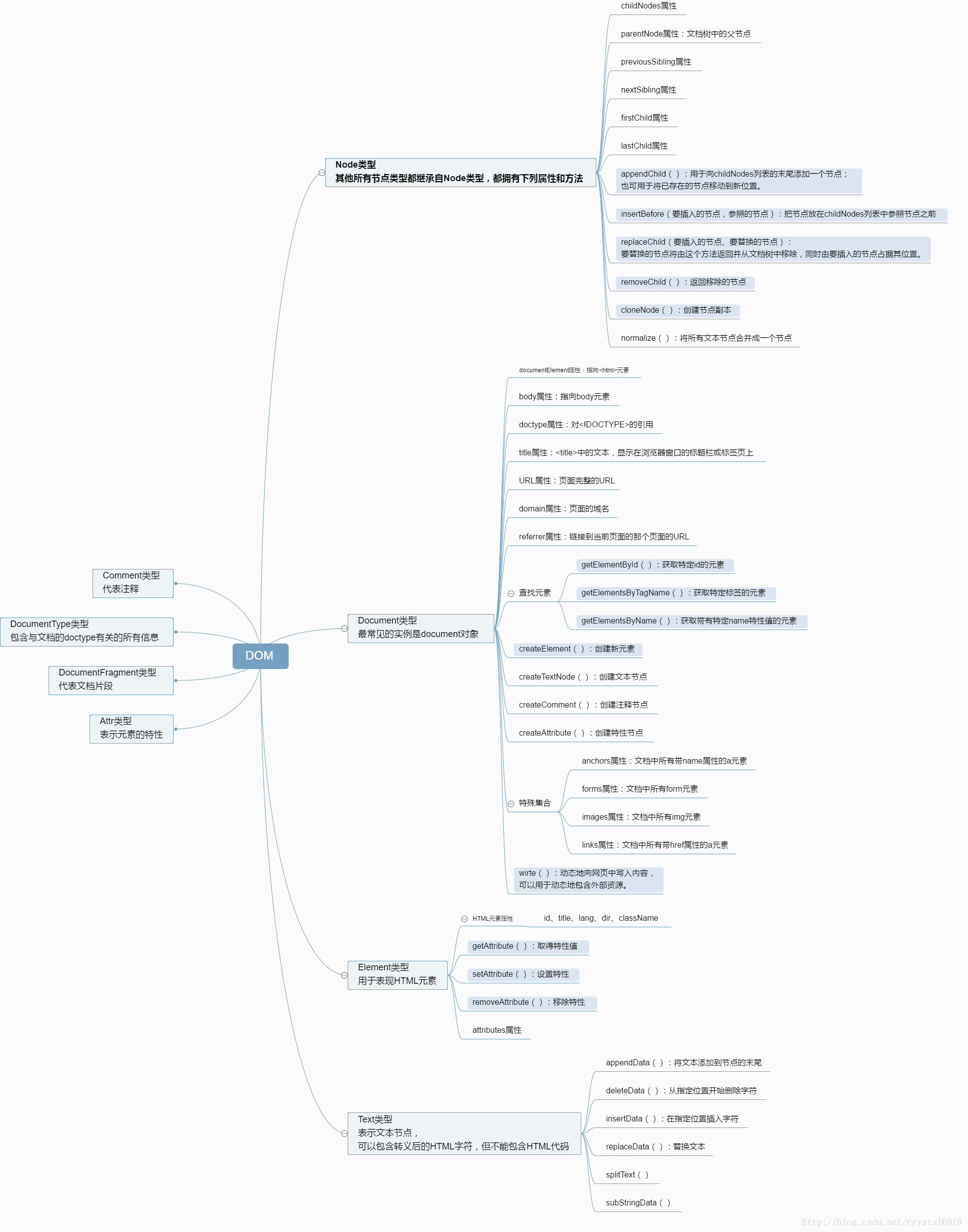
document属于window对象，是window对象的属性。



由于javascript ，xml 和 html是由不同的人开发的，javascript不能直接操作 html和xml文档，不能直接操作怎么办呢？W3C就出来制定了一个规范，也就是dom规范，W3c规定了一些函数，可以通过这些函数获取某个html元素，也可以获取某个xml元素，然后由javascript开发人实现这些函数，那么也就实现了由javascript来操作html和xml文件。由于html和xml是不一样的，所以也就出现了 html dom编程和 xml dom编程两种规范。

作用：js把浏览器，网页文档和网页文档中的html元素都用相应的内置对象来表示，这些内置对象就是dom对象，编程人员通过访问dom对象，就可以实现对浏览器本身、网页文档、网页文档中元素的操作，从而控制浏览器和网页元素的行为和外观。

### dom对象层次图



### dom对象的加载过程

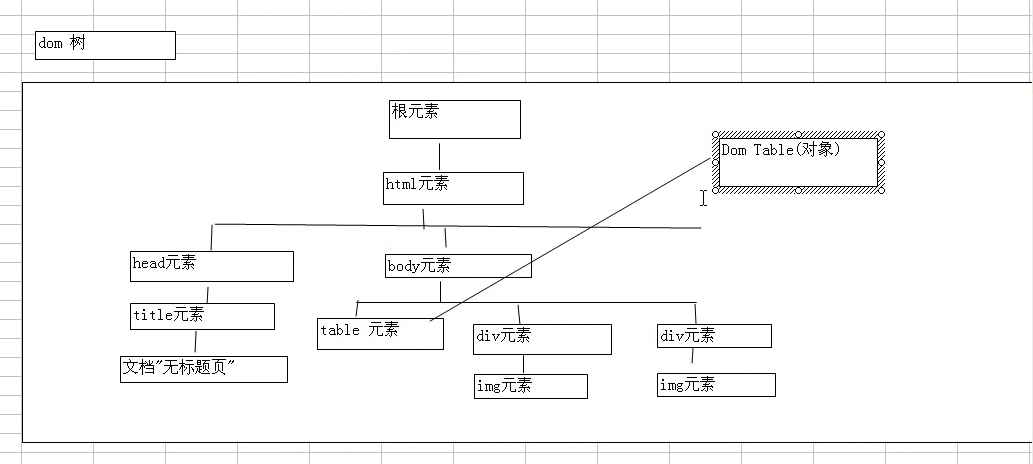
浏览器加载html页面，会在浏览器内部创建document对象，使用对象进行操作，操作的是内部创建的document对象封存的信息，最后它会更新源码。 不存在使用document修改html源码。

## document树

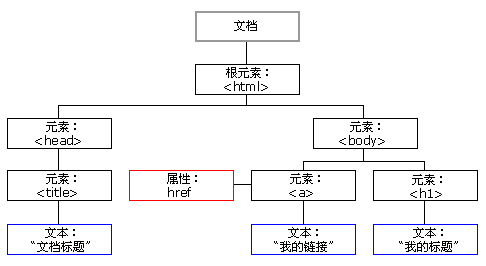
### 代码

|  |
| --- |
| <html>  <head>  </head>  <body onkeydown="move(event)">  <table border="1">  <tr>  <td></td>  <td><input type="button" value="向上走" onclick ="move(this)"/></td>  <td></td>  </tr>    <tr>  <td><input type="button" value="向左走" onclick ="move(this)"/></td>  <td></td>  <td><input type="button" value="向右走" onclick ="move(this)"/></td>  </tr>    <tr>  <td></td>  <td><input type="button" value="向下走" onclick ="move(this)"/></td>  <td></td>  </tr>  </table>  <div style="width:500px;height:400px;background-color:pink">  <img id="wugui" src="wugui.png" border="1" style="position:absolute;width:50px;height:40px;left:100px;top:120px"/>  <img id="cock" src="ji.png" border="1" style="position:absolute;width:50px;height:40px;left:200px;top:200px"/>  </div>  </body>  </html> |

### 上述diamante的dom树



### html的dom层次图



dom编程时，html文档里的table，img，title等等就是内置对象，这个可以使用window.alert()；函数输出。这些内置的对象提供了很多方法，这样就很方便的使用了。

## Dom对象的方法

### 获取对象的方式

直接获取方式：

通过id

通过name属性值

通过标签名

通过class属性值

间接获取方式：

父子关系

子父关系

兄弟关系

### 通过id方式获取

方法：getElementById(id);

注：1.id的一般只有一个，但是可以定义多个相同的id，通过这个方法获取id的标签内容时，即使有多个相同id，也只会获取一个，并且只会获取相同id的第一个id。

2. id号不要用数字开头

|  |
| --- |
| function test(){  var myhref=document.getElementById("a1");//其实它只找一个，只找第一个，找到第一个后就返回了，不往后执行了  window.alert(myhref);  }  <!-- 故意定义了三个id号相同的超链接 -->  <a id="a1" href="http://www.baidu.com">连接到baidu</a>  <a id="a1" href="http://www.sina.com">连接到sohu</a>  <a id="a1" href="http://www.163.com">连接到163</a> |

### 通过name方式获取

方法：getElementsByName(name);

注：1注意是Elements 不是 Element。

|  |
| --- |
| function test2() { //id不能重复，但是name可以重复 var hobbies=document.getElementByName("hobby"); //window.alert(hobbies.length); for(var i=0;i<hobbies.length;i++) { //如何判断是否选择 if(hobbies[i].checked){ window.alert("你的爱好是"+hobbies[i].value); } } }  <input type="checkbox" name="hobby" value="足球"/> 足球  <input type="checkbox" name="hobby" value="旅游"/> 旅游  <input type="checkbox" name="hobby" value="音乐"/> 音乐  <input type="button" value="testing" onclick="test2()"/> |

### 通过标签名方式获取

方法：getElementsByTagName(tagName);

|  |
| --- |
| //通过标签名来获取对象（元素） function test3() { var myObjs=document.getElementsByTagName("input"); for(var i=0;i<myObjs.length;i++) { window.alert("input:"+myObjs[i].value); } }  <input type="checkbox" name="hobby" value="足球"/> 足球  <input type="checkbox" name="hobby" value="旅游"/> 旅游  <input type="checkbox" name="hobby" value="音乐"/> 音乐  <input type="button" value="获取所有input" onclick="test3()"/> |

### 通过class属性方式获取

方法：getElementsByClassName(class)

function testGetEleByClassName(){

var inps=document.getElementsByClassName("common");

alert(inps.length);

}

### dom对象的节点属性和方法

在dom编程中，一个html文档会被当作 dom 树来对待，dom会把所有的 html 元素映射成Node节点，于是你就可以使用Node节点（对象）的属性和方法。

## 操作对象

### 操作html 固有元素属性（查看和修改）

可以获取和id和name值，但一般一会修改这两个值，修改id可能会导致css样式的变化，修改name值，后台通过name获取值可能找不到。

通过这种方式不能查找和修改， 用户自定义的属性。

获取属性值：元素对象.属性

|  |
| --- |
| //  function testField(){  //获取元素对象  var inp=document.getElementById("uname");  //获取元素属性值  alert(inp.type+":"+inp.name+":"+inp.id+":"+inp.value);  } |

修改元素属性值：元素对象.属性 = 属性值。

|  |
| --- |
| function testField2(){  //获取元素对象  var inp=document.getElementById("uname");  //修改元素属性  inp.value="哈哈";  inp.type="button";  } |

2自定义属性

获取自定义属性：对象. getAttribute(自定义的属性)；

|  |
| --- |
| //获取  function testOwnField(){  //获取元素对象  var inp=document.getElementById("uname");  //获取自定义属性的值  alert(inp.getAttribute("abc"));//获取到“嘿嘿”    用户名 : <input type="text" name="uname" id="uname" value="" abc="嘿嘿"/> |

修改自定义属性对象. setAttribute(自定义的属性)；

|  |
| --- |
| function testOwnField2(){  //获取元素对象  var inp=document.getElementById("uname");  //修改自定义属性的值  inp.setAttribute("abc","呵呵");  }  用户名 : <input type="text" name="uname" id="uname" value="" abc="呵呵"/> |

使用自定义方式操作固有属性

|  |
| --- |
| function testOper(){  //获取元素对象  var inp=document.getElementById("uname");  //操作对象属性  alert(inp.getAttribute("type"));  alert(inp.getAttribute("value"));  } |

### 操作元素内容

获取元素对象

元素对象名.innerHTML//返回当前元素对象的所有内容，包括HTML标签

元素对象名.innerText//返回当前元素对象的文本内容，不包括HTML标签

修改

元素对象名.innerHTML="新的值"//会将原有内容覆盖，并HTML标签会被解析

元素对象名.innerHTML=元素对象名.innerHTML+"新的值"//追加效果

元素对象名.innerText="新的值"//会将原有内容覆盖，但HTML标签不会被解析，会作为普通文本显示。

### 通过js修改css的样式

需求：在浏览网页的过程中修改背景颜色或者更换某个网站的背景。

js操作元素样式：

1.获取元素对象

2.通过style属性

元素对象名.style.样式名="样式值"//添加或者修改

元素对象名.style.样式名=""//删除样式

注意:

以上操作，操作的是HTML的style属性声明中的样式。而不是其他css代码域中的样式。如果style没有了，他会从其他的样式去寻找，style的级别最高。

3通过className

元素对象名.className="新的值"//添加类选择器样式或者修改类选择器样式

元素对象名.className=""//删除类样式。

|  |
| --- |
| <script type="text/javascript">  //js操作元素样式  //js给元素操作样式---style  function testOperCss(){  //获取元素对象  var showdiv=document.getElementById("showdiv");  //添加元素样式  showdiv.style.backgroundColor="#FFA500";    //js修改元素样式  showdiv.style.border="solid 2px red";  //js删除样式  showdiv.style.border="";  删除样式是删除的style属性添加的样式，而css中的样式可能还存在。依然会有样式。  }  //js操作样式--className  function testOperCss2(){  //获取元素对象  var div01=document.getElementById("div01");  //获取  alert(div01.className);  //添加或者修改  div01.className="common2";  //删除  div01.className="";    }  </script> |

## 操作文档结构

第一种方式：使用innerHTML

div.innerHTML=div.innerHTML+"内容"//增加节点

div.innerHTML=""//删除所有子节点

父节点.removeChild(子节点对象)//删除指定的子节点。

1、增加节点

|  |
| --- |
| function testAdd(){  //获取元素对象  var showdiv=document.getElementById("showdiv");  //给div追加上传按钮  showdiv.innerHTML=showdiv.innerHTML+  "<div><input type='file' value='' /><input type='button' value='删除' onclick='delInp(this)'/></div>";  } |

2、删除节点（父节点对象删除子节点对象）

|  |
| --- |
| function delInp(btn){  //获取父级div  var showdiv=document.getElementById("showdiv");  //获取要删除的子div  var cdiv=btn.parentNode;  //父div删除子div  showdiv.removeChild(cdiv);  } |

# jquery

## jquery

### 什么jquery

jQuery(javaScript Query)是一个优秀的JavaScript库，是一个凭借简洁的语法和跨平台的兼容性，极大地简化了JavaScript开发人员遍历HTML文档，操作DOM，执行动画和开发Ajax的操作。jQuery封装了很多预定义的对象和函数。其理念：write less,do more.

### 常见的javascript库

Prototype:是最早成型的JS库之一，对于JS的内置对象做了大量的扩展。

Do jo：提供了很多奇特JS库没有提供的功能。例如：离线存储的API，生成图标的组件等等。

YUI：是由Yahoo公司开发的一套完备的，扩展性良好的富交互网页程序工作集。

Ext JS：原本是对YUI的一个扩展，主要用于创建前端用户界面。

Moo Tools：是一套轻量、简洁、模拟化和面向对象的JS框架。

jQuery：同样是一个轻量级的库，拥有强大的选择器等更多优点，吸引了更多开发者去学习使用它。

### jQuery的功能

HTML元素选取，HTML元素操作，CSS操作，HTML事件函数，JavaScript特效和动画， HTML DOM遍历和修改，AJAX， Utilities。

### jQuery函数

jQuery库只提供了一个叫jQuery的函数，该函数中以及该元素的原型中定义了大量的方法。jQuery函数具有四种参数：

1）选择器（字符串）

jQuery函数通过该选择器获取对应的DOM，然后将这些DOM封装到一个人jQuery对象中并返回。

2）DOM对象（即Node实例）

jQuery函数将该DOM封装成jQuery对象并返回。

3）HTML文本字符串

jQuery函数根据传入的文本创建好HTML元素并封装成jQuery对象并返回。

$("<div class="one">one</div>");

4）一个匿名函数

$(function(){});

当文当加载完毕之后jQuery函数调用匿名函数。

### jQuery对象

jQuery对象是jQuery函数的一个实例，是一个类数组对象，数组中存放的是DOM对象，而DOM对象是Node的实例。

对jQuery对象的操作实际上是对jQuery数组中的DOM对象的批量操作。jQuery对象和DOM对象可以相互转化。

jQuery对象的获取通常是使用选择器来获取的。比如，获取所有clss为one元素:$(".one");