# Html+CSS

## 前端基础知识介绍

### 1）HTML5与XHTML

HTML5是一个比较新的标准，这个标准新加了很多可以用的标签和属性，各大浏览器按这个标准去实现很多这些新加的标签和属性。

XHTML是结合HTML和XML和HTML的一种标准，对语法要求比较严格，并且为了兼容XML，在语法上与HTML有一些不同。

### 2）Web Server 和Web Services

后端相关知识。Web server生成HTTP响应，确定服务器需要返回的html代码等。为了保证客户端与服务器的交互，需要指明协议（http/https）和一种数据封装格式（html/xml），web server提供的web service，指的就是这种协议+格式的交流体系。不过web service的生态系统和HTML的标准不一样，用户可以选择的协议和数据封装格式更多。普通的网站访问采用的HTTP+HTML只是其中一种，一些封闭系统内的交流还可以自己定义一个协议和格式来用（比如QQ）。

Web Service传输的数据再经本地客户端分析渲染，就能够以普通人能够理解的形式展现出来。除了Web Service,Web Server还能兼顾很多功能，包括提供缓存、平衡负载等。

### 3）一个普通网站的访问过程

* 用户操作浏览器访问，浏览器向服务器发出一个 HTTP 请求；
* 服务器接收到 HTTP 请求，Web Server 进行相应的初步处理，使用服务器脚本生成页面；
* 服务器脚本（利用Web Framework）调用本地和客户端传来的数据，生成页面；
* Web Server 将生成的页面作为 HTTP 响应的 body，根据不同的处理结果生成 HTTP header，发回给客户端；
* 客户端（浏览器）接收到 HTTP 响应，通常第一个请求得到的 HTTP 响应的 body 里是 HTML 代码，于是对 HTML 代码开始解析；
* 解析过程中遇到引用的服务器上的资源（额外的 CSS、JS代码，图片、音视频，附件等），再向 Web Server 发送请求，Web Server 找到对应的文件，发送回来；
* 浏览器解析 HTML 包含的内容，用得到的 CSS 代码进行外观上的进一步渲染，JS 代码也可能会对外观进行一定的处理；
* 用户与页面交互（点击，悬停等等）时，JS 代码对此作出一定的反应，添加特效与动画；
* 交互的过程中可能需要向服务器索取或提交额外的数据（局部的刷新，类似微博的新消息通知），一般不是跳转就是通过 JS 代码（响应某个动作或者定时）向 Web Server 发送请求，Web Server 再用服务器脚本进行处理（生成资源or写入数据之类的），把资源返回给客户端客户端用得到的资源来实现动态效果或其他改变。

## 基本标签

现在一般使用xhtml1.0版本，这种版本比较规范。

### 1）<p>标签，段落

添加段落，默认段前段后有空白。

### 2）<hx>标签，标题

添加标题。标题标签共有6个，h1、h2、h3、h4、h5、h6，分别为一级标题、二级标题、三级标题、四级标题、五级标题、六级标题，依据重要性递减。<h1>是最高等级。

### 3）<strong>、<em>，加强语气

<em>表示强调，在浏览器中默认用斜体表示，<strong>表示更强烈的强调，默认用粗体表示。目前国内前端程序员比较喜欢用<strong>。

### 4）<span>，为文字设定单独样式

<span>标签没有语义，作用是设置单独样式。

### 5）<q>、<blockquote>，文本引用

<q>短文本引用，引用的文本不用加双引号，浏览器会自动添加双引号。<q>标签的真正关键点不是它的默认样式双引号，而是它的语义：引用别人的话。

<blockquote>长文本引用，引入大段文字。浏览器对<blockquote>标签的解析式缩进样式。

### 6）<br>，分行显示文本

在需要回车换行的地方加入<br />，<br />标签作用相当于word文档中的回车。<br />是一个空标签，即没有内容的标签。类似的有<br />、<hr />、<img />。Html忽略回车和空格。

### 7）&nbsp; 空格

### 8）<hr>，添加水平横线

<hr>是一个空标签。<hr />在浏览器中默认样式线条比较粗，颜色为灰色。

### 9）<address>，添加地址信息

<address>标签在浏览器显示样式为斜体

### 10）<code>、<pre>，加入代码

<code>标签可插入一行代码，插入多行代码使用<pre>标签。<pre>标签的作用：预格式化的文本。被包围在pre元素中的文本通常会保留空格和换行符。<pre>标签不只是为显示计算机的源代码时用的，若需要在网页中预显示格式时都可以使用它。

### 11）<ul>、<ol>标签，添加列表

ul-li是没有前后顺序的信息列表。在网页显示中，默认样式为每个li前面自带一个圆点。

<ul>

<li>信息</li>

<li>信息</li>

……

</ul>

<ol>可用于制作有序列表，语法同<ul>。<ul>标签在网页中的默认显示样式为每项<li>前自带一个序号，序号默认从1开始。

### 12）<div>，排版

在网页制作过程中，可以将一些独立的逻辑部分划出来，放在一个<div>标签中，该标签的作用相当于一个容器。为使逻辑更清楚，可以用id属性来为<div>提供唯一的名称，id是唯一的。

### 13）table标签，网页上的表格

<table></table>:整个表格以<table>标记开始、</table>标记结束。

<tbody></tbody>：表格内容较多时，表格会下载一点显示一点，若加上<tbody>标签，表格要等表格内容全部下载完才会显示。

<tr></tr>：表格的一行，有几对tr表格就有几行。

<td></td>：表格的一个单元格，一行中包含几对td说明一行有几列。

<th></th>：表格的头部的一个单元格，表格表头。

table表格在没有添加css样式，在浏览器中显示是没有表格线的。

<th>表头，文本默认为粗体并且剧中显示。

用css样式为表格加入边框。eg:

<style type=”text/css”>

table tr td,th{border:1px solid:#000;}

</style>

上述代码为th，td单元格添加粗细为一个像素的黑色边框。

<caption>标签，为表格添加标题和摘要。

摘要的内容是不会显示出来的，作用是增加表格的可读性（语义化），语法：<table summary=”表格简介文本”>

标题用以描述表格内容，标题显示在表格上方。语法<caption>标题文本</caption>

### 14）<a>标签，链接到另一个页面

<a>标签可以实现超链接。

语法:<a href=”目标网址” title=”鼠标滑过显示的文本”>链接显示的文本</a>

title属性的作用，鼠标滑过链接文字时会显示这个属性的文本内容。

只要为文本加入a标签后，文字的颜色就会自动变为蓝色（被点击过的文本颜色为紫色）

<a>标签默认在当前浏览器窗口打开链接的网页，若要在新的浏览器窗口打开，添加target属性，target=”\_blank”。



注意：若mailto后面同时有多个参数的话，第一个参数必须以“？”开头，后面的参数每一个以”&”分隔。

### 15）<img>标签，为网页插入图片

语法：<img src=”图片地址” alt=”下载失败时的替换文本” title=”提示文本” />

src:标识图像的位置。

alt：指定图像的描述性文本，当图像不可见时（下载不成功时），可看到该属性指定的文本。

title:提供在图像可见时对图像的描述（鼠标滑过图片时显示的文本）。

图像可以是GIF,PNG,JPEG格式的图像文件。

## 表单标签，与用户交互

表单是可以把浏览者输入的数据传送到服务端，这样服务器端程序就可以处理表单传过来的数据。

语法：<form method=”传送方式” action=”服务器文件”>

<form>标签是成对出现的，以<form>开始，以</form>结束。

action：浏览者输入的数据被传送到的地方，比如一个PHP页面（save.php）。

method：数据传送的方式(get/post)。

注意：表单的所有控件（文本框、文本域、按钮、单选框、复选框等）都必须放在<form></form>标签之间，不然用户输入信息无法提交到服务器。

1. **文本输入框、密码输入框**

语法:<form>

<input type=”text/password” name=”名称” value=“文本” />

</form>

当type=”text”时，输入框为文本输入框；当type=”password”时，输入框为密码输入框。

name：为文本框命名，以备后台程序ASP、PHP使用。

value：为文本框设置默认值，一般起到提示作用。

1. **文本域，支持多行文本输入**

语法：<textarea rows=”行数” cols=”列数”>文本</textarea>

rows和cols可用css样式的width和height代替。col->width。

1. **单选框、复选框，让用户选择**

单选框中的选项只能选择一项，复选框中的选项可以任意选择多项。

语法：<input type=”radio/checkbox” value=”值” name=”名称” checked=”checked” />

type：radio为单选框，checkbox为复选框。

value：提交数据到服务器的值。

name：为控件命名，以备后台php,asp使用。

checked：当设置checked=”checked”时，该选项被默认选中。

注意：同一组的单选按钮中，name取值一定要一致，这样才能起到单选的作用。

1. **下拉列表，节省空间**

<select>

<option value=””>内容</option>

</select>

selected=”selected”属性，则该选项被默认选中。

在<select>标签中设置multiple=”multiple”属性，可实现多选功能，ctr+单击（mac下command+单击）可以选择多个选项。

1. **提交按钮，提交数据**

<input type=”submit” value=”提交”>

只有当type值设置为submit时，按钮才有提交作用，value为按钮上显示的文字。

1. **重置按钮，重置表单信息**

语法：<input type=”reset” value=”重置”>

1. **<label>标签**

该标签不会向用户呈现任何特殊效果，它的作用是为鼠标用户改进了可能性。当用户单击选中该标签时，浏览器会自动将焦点转到和标签相关的表单控件上。

语法：<label for=”控件id名称”>

注意：标签的for属性值一定要与相关控件的id属性值相同。

eg:

<form>

<label for="male">男</label>

<input type="radio" name="gender" id="male" />

<br />

<label for="female">女</label>

<input type="radio" name="gender" id="female" />

<label for="email">输入你的邮箱地址</label>

<input type="email" id="email" placeholder="Enter email">

</form>

## CSS基本知识

CSS全称为“层叠样式表（Cascading Style Sheets）”，主要是用于定义HTML内容在浏览器内的显示样式，如文字大小、颜色、字体加粗等。

CSS样式由选择符和声明组成，而声明又由属性和值组成。

选择符：又称选择器，指明网页中要应用样式规则的元素。

声明：英文大括号中的部分就是声明，属性和值之间用英文冒号；多条声明之间用英文符“;”隔开。

CSS中用/\*注释语句\*/来注释代码，html用<!---注释语句->

Eg:

p{

font-size:12px;

color:red;

font-weight:bold;

}

## CSS样式基本知识

**1）内联式css样式，直接写在现有的html标签中。**

注意:要写在元素的开始标签中，并且css样式代码要写在style=””双引号中，多条用分号隔开。

<p style="color:red">这里文字是红色。</p>

**2）嵌入式css样式，写在当前文件中**

将css样式代码写在<style type=”text/css”></style>标签之间。一般情况下嵌入式css代码写在<head></head>标签之间。

<style type="text/css">

span{

color:red;

}

</style>

**3）外部式css样式，写在单独的一个文件中**

也称为外联式，该文件以.css为扩展名，在<head>内使用<link>标签将css样式文件链接到HTML文件内。

注意：css样式文件名以有意义的英文字母命名，如main.css；rel=”stylesheet” type=”text/css”是固定写法不可修改；<link>标签位置一般写在<head>标签之内。

<link href="**base.css**" rel="stylesheet" type="text/css" />

**4）css样式优先级**

三种样式的优先级:内联式>嵌入式>外部式（就近原则），该优先级前提是css样式在相同的权值情况下。

嵌入式大于外部式有一个前提:嵌入式css样式的位置一定在外部式的后面。即<link href=”xx.css”>代码在<style type=”text/css”>...</style>代码的前面。

## CSS选择器

选择器{

样式；

}

选择器指明了{}中的样式的作用对象，也就是样式作用于网页中的哪些元素。

**1）标签选择器**

标签选择器其实就是html代码中的标签，如<html>、<body>、<p>、<img>。

p{font-size:12px;line-height:1.6em;}

**2）类选择器**

类选择器在css样式编码中是最常用到的。其中类是为标签设置的类名。

语法：.类选择器名称{css样式代码;}

注意：英文圆点开头；类选择器名称可以任意起名（但不要起中文）。

.stress{color:red;}

**3）ID选择器**

ID选择器类似于类选择符，但有一些重要的区别：

1. 为标签设置id=“ID名称”，而不是class=“类名称”；
2. ID选择符的前面是井号(#)，而不是英文圆点（.）。

**4）类和ID选择器的区别（谨记id的唯一性）**

相同：可以用于任何元素。

不同：a)ID选择器只能在文档中使用一次（id是唯一的），类选择器可以使用多次；b)可以使用类选择器词列表方法为一个元素同时设置多个样式。

.stress{

    color:red;

}

.bigsize{

    font-size:25px;

}

<p>到了<span **class="stress bigsize"**>三年级</span>下学期时，我们班上了一节公开课...</p>

**5）子选择器**

即大于符号（>），用于选择指定标签元素的第一代子元素。

.food>li{border:1px solid red;}

该代码会是class名为food下的子元素li加入红色实现边框。

**6）包含（后代）选择器**

即加入空格，用于选择指定标签元素下的后辈元素。

.first span{color:red;}

注意：子选择器仅指它的直接后代，而后代选择器是作用于所有子后代元素。

**7）通用选择器**

通用选择器是功能最强大的选择器，使用一个\*号指定，作用是匹配html中的所有标签元素。

**8）伪类选择器**

允许给html不存在的标签（标签的某种状态）设置样式，如给html中一个标签元素的鼠标滑过状态来设置字体颜色：

a:hover{color:red;}

目前，可以兼容所有浏览器的“伪类选择符”就是a标签上使用：hover了。

**9）分组选择符**

分组选择符（，）。

h1,span{color:red;}

相当于下面代码

h1{color:red;}

span{color:red;}

## CSS继承、层叠和特殊性

**1）继承**

css的某些样式具有继承性。继承是一种规则，它允许样式不仅应用于某个特定html标签元素，而且应用于其后代。

p{color:red;}

上述代码，颜色设置不仅应用于P标签，还应用与p标签中的所有子元素文本。

但有一些css样式不具备继承性，如border：1px solid red;该段代码只是给p标签设置了边框为1的像素、红色、实心边框线，而对于子元素标签没起到作用。

**2）特殊性**

p{color:red;}

.first{color:green;}

<p class="first">三年级时，我还是一个<span>胆小如鼠</span>的小女孩。</p>

上段代码的两个css样式都匹配到p这个标签上，但是green才是正确颜色。因为浏览器根据权值来判断使用哪种css样式，使用权值高的。

权值规则：

标签的权值为1，类选择法的权值为10，ID选择符的权值最高为100。如下所示：

p{color:red;} **/\*权值为1\*/**

p span{color:green;} **/\*权值为1+1=2\*/**

.warning{color:white;} **/\*权值为10\*/**

p span.warning{color:purple;} **/\*权值为1+1+10=12\*/**

#footer .note p{color:yellow;} **/\*权值为100+10+1=111\*/**

注意：继承也有权值，但很低，有文献指出它只有0.1，所以可以理解为继承的权值最低。

**3）层叠**

层叠就是在html文件中对于同一个元素可以有多个css样式存在，当有相同权重的样式存在时，会根据这些css样式的前后顺序来决定，处于最后面的css样式会被应用。

**4）重要性**

使用!important为某些样式设

置最有最高权值。

注意：!important要写在分号前。

当网页制作者不设置css样式时，浏览器会按照自己的一套样式来显示网页。并且用户也可以在浏览器中设置自己习惯的样式。样式的优先级：浏览器样式<网页制作者样式<用户自己设置的样式，但是!important优先级样式是个例外，权值高于用户自己设置的样式。

## CSS格式化排版

**1）文字排版**

#### a)字体

可以使用css样式为网页中的文字设置字体、字号、颜色等样式属性。

body{font-family:"宋体";}

注意：不要设置不常用的字体，因为如果用户在本地电脑上没有所安装设置的字体，就会显示浏览器默认字体。现在一般网页喜欢使用“微软雅黑”。

body{font-family:"Microsoft Yahei";}

或

body{font-family:"微软雅黑";}

注意：第一种方法比第二种方法兼容性更好一些。

#### b）字号、颜色

body{font-size:12px;color:#666}

#### c）粗体

可以使用css样式来改变文字的样式：粗体、斜体、下划线、删除线。如下代码可以实现粗体：

p span{font-weight:bold;}

#### d）斜体

p a{font-style:italic;}

#### e）下划线

p a{text-decoration:underline;}

#### f）删除线

.oldPrice{text-decoration:line-through;}

**2）段落排版**

#### a）缩进

p{text-indent:2em;}

2em的意思就是文字的2倍大小。

#### b）行间距（行高）

p{line-height:1.5em;}

#### c）中文字间距、字母间距

使用letter-spacing设置文字间隔或字母间距。

h1{

letter-spacing:50px;

}

如果要设置英文单词之间的间距，采用word-spacing来实现。

h1{

word-spacing:50px;

}

#### d）对齐

使用text-align样式代码设置文本、图片位置。

h1{

text-align:center;//center居中 left 居左 right居右

}

## CSS盒模型

**1）元素分类**

在css中，html中的标签元素大体被分为三种不同类型：块状元素、内联元素（又叫行内元素）和内联块状元素。

常用的块状元素有：<div>、<p>、<h1>...<h6>、<ol>、<ul>、<dl>、<table>、<address>、<blockquote> 、<form>。

常用的内联元素有：<a>、<span>、<br>、<i>、<em>、<strong>、<label>、<q>、<var>、<cite>、<code>。

常用的内联块状元素有：<img>、<input>。

#### a）块级元素

在html中，<div>、<p>、<h1>、<form>、<ul>、<li>就是块级元素。设置display:block就是将元素显示为块级元素。如下代码将内联元素a转换为块状元素，从而使a元素具有块状元素特点。

a{display:block;}

块级元素特点：

每个块级元素都从新的一行开始，并且其后的元素也另起一行；

元素的高度、宽度、行高以及顶和底边距都可设置；

元素宽度在不设置的情况下，是它父容器的100%（和父元素的宽度一致），除非设定一个宽度。

#### b）内联元素

<span>、<a>、<label>、<strong>和<em>是典型的内联元素（行内元素）（inline）。块状元素可以通过display:inline将元素设置为内联元素。

内联元素特点：

和其他元素都在一行上；

元素的高度、宽度及顶部和底部边距不可设置；

元素的宽度就是它包含的文字或图片的宽度，不可改变。

注意：当行内元素之间有”回车”、“tab”、“空格”时就会出现间隙。解决办法：写在一行、之间不要有空格之类的符号或者使用font-size:0

#### c）内联块状元素

内联块状元素就是同时具备内联元素、块状元素的特点，代码display:inline-block就是将元素设置为内联块状元素。<input>、<img>就是内联块状元素。

inline-block元素特点：

和其他元素在同一行；

元素的高度、宽度、行高以及顶和底边距都可设置。

**2）盒模型**

块级标签都具备盒子模型的特征。

1. 边框

盒子模型的边框就是围绕着内容及补白的线，可以设置这条线的粗细、样式、颜色（边框的三个属性）。

div{

border:2px solid red;

}

上面的代码可以拆分为：

div{

border-width:2px;

border-style:solid;

border-color:red;

}

注意：border-style常见有dashed（虚线）、dotted（点线）、solid（实线）。

border-color可以设置为十六进制颜色。

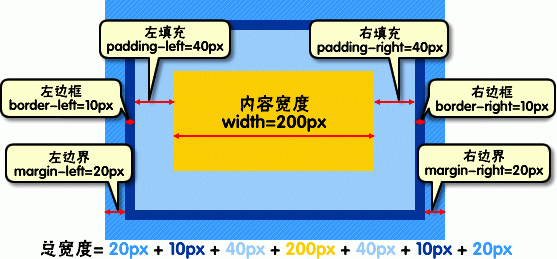
boder-width可以设置为thin|medium|thick（不常用），通常用像素（px）。

css样式允许只为一个方向的边框设置样式：

div{**border-bottom**:1px solid red;}

1. 宽度和高度

盒模型的宽度和高度和我们平常所说的物体的宽度和高度理解是不一样的，css内定义的宽（width）和高（height），指的是填充以里的内容范围。因此，一个元素实际宽度=左边界+左边框+左填充+内容宽度+右填充+右边框+右边界。



1. 填充

元素的内容和边框之间可以设置距离，称指为填充。填充分为上、下、左、右（顺时针），可以分开写。

div{padding:20px 10px 15px 30px;}

如果上、右、下、左填充都为10px，可以写为：

div{padding:10px;}

如果上下填充都一样为10px，可以写为：

div{padding:10px 20px;}

1. 边界

元素与其他元素之间的距离可以使用边界（margin）来设置，与填充类似。边界使用margin标识。

margin与padding区别：padding在边框里，margin在边框外。

## CSS布局模型

布局模型是建立在盒模型基础之上，又不同于我们常说的CSS布局样式或CSS布局模板。CSS包括3中布局模型，即流动模型（Flow）、层模型（Layer）和浮动模型（Float）。

**1）流动模型**

流动（Flow）是默认的网页布局模式。流动模型典型特征：第一点，块状元素都会在所处的包含元素内自上而下按顺序垂直延伸分布。因为默认状态下，块状元素的宽度都为100%。实际上，块状元素都会以行的形式占据位置。第二点，在流动模型下，内联元素都会在所处的包含元素内从左到右水平分布显示。

**2）浮动模型**

设置元素浮动可以让两个块状元素并排显示。任何元素在默认情况下是不能浮动的，但可以用css定义为浮动。

div{

width:200px;

height:200px;

border:2px red solid;

float:left;

}

<div id="div1"></div>

<div id="div2"></div>

浮动存在的问题：a）最典型问题是一个父元素包含了多个浮动的子元素。页面的内容设置了一个宽度，子元素的浮动确定了他们的位置，但浮动元素不会影响父元素的宽度。这样做会让父元素塌陷，从而使父元素的高度为“0”，以及忽略其他的属性。b）嵌套的元素不会正确的排列，也会有错误的样式出现。

如何清除浮动:

方法一：在容器的结束标签前添加一个空标签，在空标签上直接设置演示“clear:both”。这种方式大多数情况下有效，但是不太适合语义化。

方法二：overflow技巧。在具有浮动元素的父容器中设置“overflow”的属性值为“auto”。建议，最好使用overflow:hidden;来清楚子标签浮动对父容器的影响。

方法三：clearfix技巧。根据上下文，清楚浮动更好的方法是clearfix技巧。该技巧是基于在父元素上使用“：before”和“:after”两个伪类。使用这些伪类，我们可以在浮动元素的父容器前面和后面创建隐藏元素。“：before”伪类是用来防止子元素顶部的外边距塌陷。“：before”伪类是用来防止子元素的底部的外边距塌陷，以及用来清除元素的浮动。

**3）层模型**

层布局模型就像图像软件photoShop中非常流行的图层编辑功能一样，每个图层都能够精确定位操作。css定义了一组定位属性来支持层布局模型。层模型有五种形式：默认定位position:static；绝对定位position：absolute；相对定位position：relative；固定定位position：fixed；粘性定位position:sticky。

1. 默认定位

Static定位意味着他们没有也不接受位置属性设置。如果 元素设置了position属性，将会覆盖元素的默认值：static。

1. 绝对定位

设置position:absolute表示绝对定位，作用是将元素从文档流中拖出来，然后使用left、right、top、bottom属性相对于其最接近的一个具体属性的父包含块进行绝对定位。如果不存在这样的包含块，则相对于body元素，即相对于浏览器窗口。如下代码可以实现div元素相对于浏览器窗口向右移动100px，向下移动50px。

div{

width:200px;

height:200px;

border:2px red solid;

position:absolute;

left:100px;

top:50px;

}

<div id="div1"></div>

1. 相对定位

Position:relative表示相对定位，通过left、right、top、bottom属性确定元素在正常文档流中的偏移位置。相对定位完成的过程是首先按static（float）方式生成一个元素（并且元素像层一样浮动起来），然后相对于以前的位置移动，移动幅度有left、right、top、bottom属性确定，偏移前的位置保留不对。以下代码实现相对于以前位置向下移动50px，向右移动100px。

#div1{

width:200px;

height:200px;

border:2px red solid;

position:relative;

left:100px;

top:50px;

}

<div id="div1"></div>

1. 固定定位

fixed表示固定定位，与absolute定位类型类似。但它的相对移动的坐标是视图本身。由于视图本身是固定的，不会随窗口的滚动条滚动而变化，除非你在屏幕中移动窗口屏幕的位置，或者改变浏览器窗口的显示大小，因此固定定位的元素会始终位于浏览器窗口内视图的某个位置，不会受文档流动的印象。这与background-attachment:fixed;属性功能相同。

1. Relative和Absolute组合使用

使用relative帮忙可以实现相对于其他元素进行定位，但必须遵守以下规范：参照定位的元素必须是相对定位元素的前辈元素；参照定位的元素必须加入position:relative;定位元素加入position：absolute；

1. 粘性定位sticky

粘性定位是相对定位和固定定位的混合。元素在跨越特定阈值前为相对定位，之后为固定定位。

#one{position:sticky;top:10px;}

在窗口中，滚动到元素top距离小于10px之前，元素为相对定位。之后元素将固定在与顶部距离10px的位置，知道viewport视口回滚到阈值一下。

粘性定位常用于定位字母列表的头部元素。标示 B 部分开始的头部元素在滚动 A 部分时，始终处于 A 的下方。而在开始滚动 B 部分时，B 的头部会固定在屏幕顶部，直到所有 B 的项均完成滚动后，才被 C 的头部替代。需要指定top，right，bottom或left四个阈值其中之一，才可使粘性定位生效否则其定位与相对定位相同。

## 其他

**1）字体缩写**

body{

font-style:italic;

font-variant:small-caps;

font-weight:bold;

font-size:12px;

line-height:1.5em;

font-family:"宋体",sans-serif;

}

可以缩写为：

body{

font:italic small-caps bold 12px/1.5em "宋体",sans-serif;

}

注意：使用这一简写方式至少要指定font-size和font-family属性，其他的属性（如font-weight,font-style,font-variant,line-height）如未指定将自动使用默认值。

在缩写font-size与line-height中间要加“/”斜杠。

**2）颜色值**

字体颜色color，背景颜色background-color，边框颜色border。设置颜色放法有很多种

a）英文命令颜色

p{color:red;}

b）RGB颜色

p{color:rgb(133,45,200);}

c）十六进制颜色

这种颜色设置方法是现在比较普遍使用的方法

**3）长度值**

a）像素

像素指的是显示器上的小点（css规范中假设90像素=1英寸）。实际情况是浏览器会使用显示器的实际像素值有关，目前大多数设计者都倾向于使用像素px作为单位。

b）em

em是本元素给定字体的font-size值。若font-size=18px，则1em=18px。

注意：当给font-size设置单位为em时，此时计算的标准以标签的父元素的font-size为基础。

c）百分比

和em类似，以标签的font-size为标准

## CSS样式设置小技巧

**1）水平居中设置-行内元素**

如果被设置元素为文本、图片等行内元素，水平居中是通过给父元素设置text-align:center;来实现。

**2）水平居中设置-定宽块状元素**

通过设置左右margin值为auto来实现居中。

**3）水平居中总结-不定宽块状元素方法**

不定宽块状元素有三种方法居中：加入table标签；设置display:inline方法，与第一种方法类似，显示类型为行内元素，进行不定宽元素的属性设置；设置position：relative和left:50%，利用相对定位的方式，将元素向左偏移50%，即达到居中的目的。

a）加入table标签

利用table标签的长度自适应性----即不定义其长度也不默认父元素body的长度。（table其长度根据其内文本长度决定），因此可以看作一个定宽度块元素，然后再利用定宽度块居中的margin方法，使其水平居中。

b）设置display：inline

改变块级元素的display为inline类型（设置为行内元素显示），然后使用text-align：center来实现居中效果。

与加入table标签相比优势是不用增加无语义标签，但也存在着一些问题，它将块状元素的display类型改为inline，变成了行内元素，所以少了一些功能，比如设定长度值。

c）设置浮动和相对定位

通过给父元素设置float，然后给父元素设置position：relative和left:50%，子元素设置position：relative和left:-50%来实现水平居中。

理解：假想ul层的父层（即下面例子中的div层）中间有条平分线将ul层的父层（div层）平局分成两份，ul层的css代码是将ul层的最左端与ul的父层的（div层）的平分线对齐；而li层的css代码则是将li层的平分线与ul层的最左端（也就是div层的平分线）对齐，从而实现li层的居中。

<body>

<div class="container">

<ul>

<li><a href="#">1</a></li>

<li><a href="#">2</a></li>

<li><a href="#">3</a></li>

</ul>

</div>

</body>

<style>

.container{

float:left;

position:relative;

left:50%

}

.container ul{

list-style:none;

margin:0;

padding:0;

position:relative;

left:-50%;

}

.container li{float:left;display:inline;margin-right:8px;}

</style>

**4）垂直居中**

垂直居中分为两种情况：父元素高度确定的单行文本以及父元素高度确定的多行文本。

1. 父元素高度确定的单行文本

通过设置父元素的height和line-height高度一致来实现。（height：该元素的高度，line-height:行高，指在文本中，行与行之间的基线间的距离。）

line-height和font-size的计算值之差，在css中成为行间距。分为两半，分别加到一个文本行内容的顶部和底部。

这种文字行高与块高一致带来了一个弊端：当文字内容的长度大于块的宽度时，就有内容脱离了块。

<div class="container">

hi,imooc!

</div>

<style>

.container{

height:100px;

line-height:100px;

background:#999;

}

</style>

1. 父元素高度确定的多行文本

方法一：

使用插入table（包括tbody、tr、td）标签，同时设置vertical-align：middle。在父元素中设置vertical-align会对inline-block类型的子元素都有用。

<body>

<table><tbody><tr><td class="wrap">

<div>

<p>看我是否可以居中。</p>

</div>

</td></tr></tbody></table>

</body>

table td{height:500px;background:#ccc}

td标签默认情况下就默认设置了vertical-align为middle，所以此处不需要显示设置。

方法二：

该方法兼容性较差，在chrome、firefox及IE8以上的浏览器下可以设置块级元素的display为table-cell（设置为表格单元显示），激活vertical-align属性。

好处是不用添加多余的无意义标签，缺点是兼容性不好，不兼容IE6、7而且这样修改display的block变成了table-cell，破坏了原有的块状元素性质。

**5）隐性改变display类型**

只要html代码中出现position：absolute或者float:left/right二者之一，元素的display显示类型就会自动变为一display:inline-bloack的方式显示。

# 网页布局

网页布局又称版式布局，是网页UI设计师将有限的视觉元素进行有机的排列组合。

网页设计的特点：网页可以自适应宽度；网页的长度理论上可以无限延长。

分栏又叫分列，常见的布局分为一列布局、两列布局、三列布局、混合布局。

1. **一列布局**

通常作为网站的首页，适合放置简单的内容，一列布局经常是固定宽度。

1. **两列布局**
2. **三列布局**
3. 设置百分比，可以利用float实现

.left{

width:200px;

height:500px;

float:left;

background:#ccc;

}

.main{

height:500px;

float:left;

background:#999;

}

.right{

width:300px;

height:500px;

float:right;

background:#ddd;

}

1. 要求左右两列是指定像素宽度，float失效，采用position

.left{

width:200px;

height:500px;

position:absolute;

left:0;

top:0;

background:#ccc;

}

.main{

height:500px;

margin:0 300px 0 200px;

background:#999;

}

.right{

width:300px;

height:500px;

position:absolute;

right:0;

top:0;

background:#ddd;

}

1. **混合布局**
2. **自动居中-列布局**
3. 标准文档

W3C标准：是由万维网联盟指定的一系列标准，包括：

结构化标准语言（HTML和XML）；

表现标准语言（CSS）；

行为标准语言（DOM和ECMAScript）。

倡导结果、样式、行为三者分离。

1. CSS中的定位机制

3种定位机制：标准文档流（Nomal flow），浮动（Floats），绝对定位（Absolute positioning）。

1. 标准文档流

特点：从上到下，从左到右，输出文档内容。由块级元素和行级元素组成。

块级元素：从左到右撑满页面，独占一行，触碰到页面边缘时，会自动换行。

行级元素：可以在同一行显示，不会破坏文档结构。

块级元素和行级元素都是盒子模型。

1. 盒子模型

盒子模型=网页布局的基石。由4部分组成：边框、外边距、内边距、内容。

盒子的3D模型：border第一层，content+padding第二层，background-img第三层，background-color第四层，margin第五层。

1. **横向两列布局**

1）浮动布局及float属性

能够实现横向多列布局，通过设置float属性实现。3个属性值：left左浮动，right右浮动，none不浮动。特点：元素左移或右移，直至触碰到容器为止。设置了浮动的元素，仍旧处于标准文档流中。

设置浮动后的标签会对紧邻该标签之后的元素产生影响。

2）清除浮动的常用方法

对受到浮动影响的元素设置如下属性：

a）clear属性——常用clear:both;

clear:left;或者clear:right;（要求清楚受到了左浮动还是右浮动的影响）

b）同时设置width:100%（或固定宽度）+ overflow:hidden;

空标签也能清除浮动，但是对元素是无意义的，不建议使用。

3）横向两列布局

网页布局最常见的方式之一。

主要应用技能：float属性——使纵向排列的块级元素，横向排列；margin属性——设置两列之间的间距。

父包含块若未设置浮动，子包含块设置了浮动，则父标签的宽度扩展将会受到影响。

当父包含块缩成一条时，用clear:both;清除浮动无效，它一般用于紧邻后面的元素清除浮动。此时应该使用溢出属性overflow:hidden;

Ps:为了保证子元素正常显示，不要设置父包含块的高度，从而使其可以自适应子元素高度。

1. **绝对定位布局**

1）绝对定位布局简介

绝对定位布局是使用position属性实现的网页布局，是css中规定的第三种定位机制，能够实现横向多列布局及较为复杂的定位。比如带有遮罩层效果的提示框、固定层效果、全屏广告等。

Position属性拥有三种定位形式：静态定位、相对定位、绝对定位。

可以设置4个属性值：static（静态定位，默认状态），relative（相对定位），absolute（绝对定位），fixed（固定定位）。

2）相对定位

特点：相对于自身原有位置进行偏移；仍处于标准文档流中；拥有偏移属性和z-index属性。

3）绝对定位

特点：建立了以包含块为基准的定位；完全脱离了标准文档流；拥有偏移属性和z-index属性。

绝对定位与相对定位相比在选择参照基准的时候要复杂的多。

未设置偏移量：无论是否存在已经定位祖先元素，都保持在初始位置；脱离标准文档流。

当一个元素设置绝对定位，没有设置宽度时，则元素宽度根据内容进行调节。

设置偏移量：偏移参照基准：若无已定位祖先元素，以html为偏移参照基准；有已经定位祖先元素，以距其最近的已定位祖先元素为基准进行偏移。

4）使用绝对定位实现横向两列布局

常用于一列宽度固定，另一列宽度自适应。

主要应用技能：relative——父元素相对定位；absolute——自适应宽度元素绝对定位。

注意：固定宽度列的高度>自适应宽度的列！！！（因为设置绝对定位的列脱离文档流，无法撑起整个页面。）

# JS基础

## 1.JavaScript基本知识

**1）为什么学习JS**

* 所有主流浏览器都支持JavaScript；
* 当前全世界大部分网页都使用JavaScript；
* 它可以让网页呈现各种动态效果；
* JavaScript可以提供漂亮的网页、令用户满意的上网体验。
* 易学性，只要文本编辑器就可以编写JavaScript程序，使用简单的命令，完成基本操作。

**2）如何插入JS**

JavaScript代码写在<script></script>标签之间 type=”text/javascript”。

1. **引用JS外部文件**

Js文件中，不需要<script>标签，直接编写javascript代码即可。

JS文件不能直接运行，需要嵌入到HTML文件中执行，需要在HTML代码中添加如下代码：<script src=”script.js”></script>。

1. **js在页面中的位置**

可以将javascript代码放在html文件中任何位置，但一般放在网页的head或者body部分。若放在<head>部分，浏览器解析head部分就会执行这个代码，然后才解析页面的其余部分。若放在<body>部分，JavaScript代码在网页读取到该语句的时候就会执行。

注意：如果是通过实践调用执行function，那么Js执行与位置无关，其他情况通常是按位置执行。

1. **JavaScript语句与符号**

JavaScript语句是发给浏览器的命令，这些命令的作用是告诉浏览器要做的事情。一行的结束就被认定为语句的结束，通常在结尾加上一个分号表示语句结束。

注意：“;”分号要在英文状态下输入，分号可以不写，但为了养成良好的编程习惯，建议还是写上分号。

1. **注释**

注释的作用是提高代码的可读性，帮助自己和别人阅读和理解代码，与java编写规则一致。

1. **变量**

变量是用于存储某种/某些数值的存储器。

定义变量的关键字var，语法为：var 变量名；

变量名命名规则：变量必须使用字母、下划线或者美元符号（$）开始；然后可以使用任意多个英文字母、数字、下划线或者美元符组成；不能使用JavaScript关键词与JavaScript保留字。

变量必须先声明再赋值，可以重复赋值。

注意：JavaScript中区分大小写。

1. **判断语句**

If…else语句在指定条件成立时执行代码。

1. **函数**

函数是完成某个特定功能的一组语句。基本语法：function 函数名(){

函数代码;

}

说明：function定义函数的关键字。

函数的执行与java类似。

## 2.常用互动方法

**1）输出内容（document.write）**

document.write()可用于直接向HTML输出流写内容，也就是直接在网页输出内容。

第一种：输出内容用“”括起，直接输出“”内的内容；

第二种：通过变量，输出内容；

第三种：输出多项内容，内容之间用+号连接；

第四种：输出HTML标签，并起作用，标签使用""括起来。

如何输出空格方法：使用输出html标签&nbsp;来解决；使用css样式来解决，document.write("<span style='white-space:pre;'>"+"  1        2    3    "+"</span>");

在输出时添加white-space:pre;样式属性，该样式表明空白会被浏览器保留。

1. **JavaScript-警告（alert消息对话框）**

alert(字符串或变量);

注意：在点击对话框“确定按钮前，不能进行其他任何操作”；消息对话框通常可以用于调试程序；alert输出内容，可以是字符串或变量，与document.write相似。

1. **确认（confirm消息对话框）**

Confirm消息对话框通常用于允许用户做选择的动作。弹出对话框，一般包括一个确定按钮和一个取消按钮。

语法：confirm(str)，返回值：Boolean，确定返回true，取消返回false。

1. **提问（prompt消息对话框）**

prompt弹出消息对话框，通常用于询问一些需要与用户交互的消息。弹出对话框包含一个确定按钮、取消按钮与一个文本输入框。

语法：prompt(str1,str2)；str1：要显示在消息对话框中的文本，不可修改。Str2：文本框中的内容，可以修改。点击确认按钮，文本框中的内容作为函数返回值；点击取消按钮，将返回null。

1. **打开新窗口（window.open）**

open()方法可以查找一个已经存在或者新建的浏览器窗口。

语法：window.open(url,窗口名称,参数字符串)

参数说明：url可选参数，在窗口中要显示网页的网址或路径。如果省略这个参数，或者它的值是空字符串，那么窗口就不显示任何文档；窗口名称，可选参数，被打开窗口的名称，该名称由字母、数字和下划线字符组成。"\_top"、"\_blank"、"\_self"具有特殊意义的名称。 \_blank：在新窗口显示目标网页，\_self：在当前窗口显示目标网页， \_top：框架网页中在上部窗口中显示目标网页。相同name的窗口只能创建一个，name不包含空格。参数字符串，可选参数，设置窗口参数，各参数用逗号隔开。

1. **关闭窗口（window.close）**

语法：window.close()；或者<窗口对象>.close();

## 3.DOM操作

**1）认识DOM**

文档对象模型DOM（Document Object Model）定义访问和处理HTML文档的标准方法。Dom将HTML文档呈现为带有元素、属性和文本的树结构（节点树）。

HTML文档可以说是由节点构成的集合，三种常见的DOM节点：

元素节点：<html>、<body>、<p>等都是元素节点。

文本节点：向用户展示的内容，如<li>…</li>中的JavaScript、DOM、CSS等文本。

属性节点 :元素属性，如<a>便签的链接属性。

**2）通过ID获取元素**

语法：ducument.getElementById(“id”);

**3）innerHTML属性**

innerHTML属性用于获取或替换HTML元素的内容。

语法：Object.innerHTML;Object是获取的元素对象，innerHTML区分大小写。

**4）改变HTML样式**

语法：Object.style.property = new style;

<p id="pcon">Hello World!</p>

<script>

var mychar = document.getElementById("pcon");

mychar.**style.color**="red";

mychar.**style.fontSize**="20";

mychar.**style.backgroundColor** ="blue";

</script>

**5）显示和隐藏（display属性）**

语法：Object.style.display = value;value取值：none此元素不会被显示，block此元素将显示为块级元素。

**6）控制类名（className属性）**

className属性设置或返回元素的class属性。

语法：object.className = classname;

作用：获取元素的class属性，为网页内的某个元素指定一个css样式来更改该元素外观。

# JavaScript进阶篇

了解JS变量、数组、函数、语法、对象、事件、DOM操作。

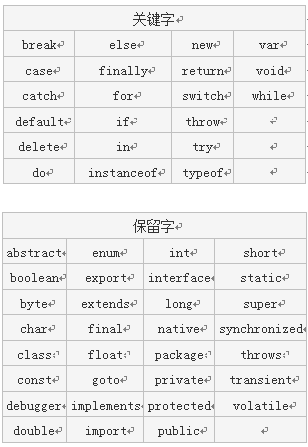
## JS基础语法

**1）变量**

变量是存储某种/某些数值的存储器。可以把变量看做一个盒子。

**2）变量命名**

变量命名规则：必须以字母、下划线或美元符号开头，后面可以跟字母、下划线、美元符号和数字；变量命名区分大小写；不允许使用JavaScript关键字和保留字做变量名。



**3）变量声明**

语法：var 变量名;

Var相当于找盒子的动作，是JavaScript中的关键字（即保留字），作用是声明变量 ，并为变量准备位置（即内存）。

var可以一次声明多个变量，变量间用“，”逗号隔开。

**4）变量赋值**

使用“=”号给变量赋值。

**5）表达式**

表达式和数学中的定义相似，表达式是指具有一定的值、用操作符把常量和变量连接起来的代数式。一个表达式可以包含常数或变量。

**6）+号操作符**

操作符是用于在JavaScript中指定一定动作的符号。

sum = numa +numb;其中的=号和+号都是操作符。

”=”操作符是赋值，不是等于。“+”不只代表加法，还可以连接两个字符串。

**7）自加一，自减一（++，--）**

**8）比较操作符**

**9）逻辑与操作符**

&&是并且的意思。

**10）逻辑或操作符（||）**

**11）逻辑非操作符（!）**

**12）操作符优先级**

操作符之间的优先级（高到低）：

算术操作符-》比较操作符-》逻辑操作符》“=”赋值符号；同级运算从左到右进行，多层括号由里向外。

## 数组

**1）什么是数组**

一个数组变量可以存放多个数据。数组是一个值的集合，每个值都有一个索引号，从0开始，每个索引都有一个相应的值，根据需要添加更多数值。

**2）如何创建数组**

var myarray=new Array();

创建数组时还可以为数组指定长度，长度任意指定。

var myarray= new Array(8);

注意：创建的新数组是空数组，没有值，如输出，则显示undefined；虽然创建数组时指定了长度，但实际上数组都是变长的，也就是说即使指定了长度8，仍然可以将元素存储在规定长度以外。

**3）数组赋值**

第一步：创建数组var myarr=new Array();

第二步：给数组赋值

myarr[1]=" 张三";

myarr[2]=" 李四";

注意：每个值都有一个索引号，从0开始。

var myarray = new Array(66,80,90,77,59);//创建数组同时赋值

var myarray = [66,80,90,77,59];//直接输入一个数组（称 “字面量数组”）

数组存储的数据可以使任何类型（数字、字符、布尔值等）。

**4）向数组增加一个新元素**

使用一个新索引，为数组增加一个新元素。

**5）使用数组元素**

引用数组变量并提供一个索引，就能得到一个数组元素的值。如:myarry[0]。

**6）数组属性（length）**

语法：myarray.length;//获得数组的长度。

JavaScript数组的length属性是可变的！！！

**7）二维数组**

二维数组的两个维度的索引值也是从0开始，两个维度的最后一个索引值为长度-1。

二维数组定义方法：

方法一：先声明一维，再声明二维（用for循环）。

方法二：var myarr = [[0,1,2],[1,2,3]];

## 流程控制语句

**1）if语句**

If语句是基于条件成立才执行相应代码时使用的语句。

注意：if要小写！

**2）二选一（if...else语句）**

**3）多重判断（if...else if嵌套）**

**4）多种选择（switch语句）**

语法说明：switch必须赋初始值，值与每个case值匹配。满足执行该case后的所有语句，并用break语句阻止执行下一个case，所有case值都不匹配，则执行default后的语句。

**5）重复（for循环）**

**6）while循环**

**7）do...while循环**

do...while结构的基本原理和while结构是基本相同的，但是它保证循环体至少被执行一次。先执行，后判断条件。

**8）退出循环（break）**

**9）继续循环（continue）**

continue的作用是仅仅跳过本次循环，而整个循环继续执行。

## 函数

**1）定义函数**

function 函数名（）{  
 函数体；

}

function定义函数的关键字，“函数体”替换为完成特定功能的代码。

**2）函数调用**

第一种情况：在script标签中调用，直接写函数名。

第二种情况：在html文件中调用，如通过点击按钮，通过设置属性onclick=”add2()”；

**3）有参数的函数**

function 函数名（参数1，参数2）{

函数代码；

}

**4）返回值的函数（return）**

函数中参数和返回值不只是数字，还可以是字符串等其他类型。

## 事件响应

**1）什么是事件**

事件是可以被JavaScript侦测到的行为。主要事件如下：



**2）鼠标单击事件（onclick）**

<input name="button" type="button" value="点击提交" onclick="add2()" />

**3）鼠标经过事件（onmouseover）**

**4）鼠标移开事件（onmouseout）**

**5）光标聚焦事件（onfucus）**

**6）失焦事件（onblur）**

**7）内容选中事件（onselect）**

当文本框或者文本域中的文字被选中时，触发onselect事件，同时调用程序会被执行。

**8）文本框内容改变事件（onchange）**

**9）加载事件（onload）**

事件会在页面加载完成后，立即发生，同时执行被调用的程序。

注意：加载页面时，触发onload事件，事件写在<body>标签内。

**10）卸载事件（onunload）**

当用户退出页面时（页面关闭、页面刷新等），触发onunload事件，同时执行被调用的程序。

注意：不同浏览器对onunload事件支持不同。

window.onunload = onunload\_message;

function onunload\_message(){

alert("您确定离开该网页吗？");

}

## JavaScript内置对象

**1）什么是对象**

JavaScript中所有事物都是对象，如字符串、数值、数组、函数等，每个对象带有属性和方法。

对象的属性：反映该对象某些特定的性质，如字符串长度、图像的长宽等；

对象的方法：能够在对象上执行的动作。例如表单的提交，时间的获取。

**2）Date日期对象**

日期对象可以存储任意一个日期，并且可以精确到毫秒数。

定义一个时间对象：var Udate = new Date();

Udate成为日期对象，并且已有初始值：当前时间（当前电脑系统时间）。

自定义初始值：var d = new Date(2017,3,2);

var d = new Date(‘Oct 1, 2012’);



注意：结果格式依次为：星期、月、日、年、时、分、秒、时区（火狐浏览器）；不同浏览器，时间格式有差异。

**3）返回星期的方法getDay()**

getDay()返回星期，返回的是0-6数字，0表示星期天。

**4）返回/设置时间方法get/setTime()**

单位毫秒数，计算从1970年1月1日零时到日期对象所指的日期的毫秒数。

时间推迟1小时，可表示为：x.setTime(x.getTime()+60\*60\*1000);

**5）String字符串对象**

Var mystr = “Hello World!”

Var mynum = mystr.toUpperCase();//将小写字母转换成大写字母。

**6）返回指定位置的字符**

charAt()方法可返回指定位置的字符，返回的字符是长度为1的字符串。

注意：如果index不在0与string.length-1之间，该方法将返回一个空字符串。

**7）返回指定的字符串首次出现的位置indexOf()**

该方法区分大小写；如果要检索的字符串值没有出现，则该方法返回-1；

**8）字符串分割split()**

若将空字符串“”作为参数，则每个字符之间都会被分割。

**9）提取字符串substring（）**

注意：如果start比top大，那么该方法在提取子串之前会先交换这两个参数。

**10）提取指定数目的字符substr()**

该方法从字符串中提取从startPos位置开始的指定数目的字符串。

stringObject.substr(startPos,length)

注意：如果startPos是负数，从字符串的尾部开始算起的位置，也就是，-1值字符串中最后一个字符，-2值倒数第二个字符，以此类推。如果startPos为负数且绝对值大于字符串长度，startPos为0。

**11）Math对象**

提供对数据的数学计算。

Math对象是一个固有的对象，无需创建它，直接把Math作为对象使用就可以调用其所有的属性和方法。这是它与Date,String对象的区别。





**12）向上取整ceil()**

可对一个数进行向上取整。

Math.ceil(x),返回的是大于或等于x,并且与x最接近的整数。

**13）向下取整floor（）**

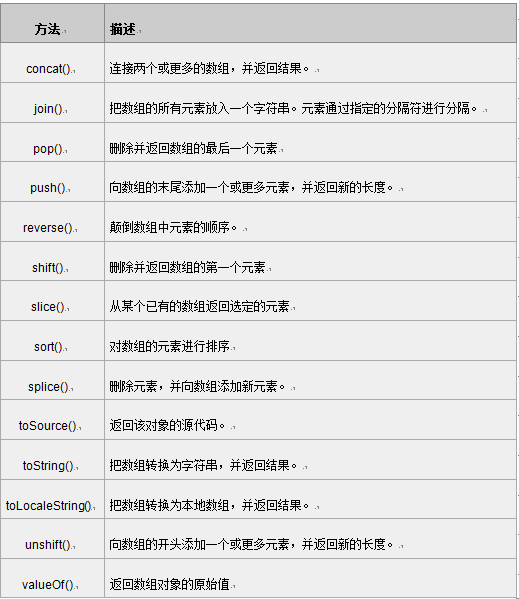
**14）四舍五入round()**

**15）随机数random()**

返回介于0~1（大于或等于0但小于1）之间的一个随机数。

**16）Array数组对象**

数组对象是一个对象的集合，里面的对象可以是不同类型的。数组的每一个成员对象都有一个下标，用来表示它在数组中的位置，从0开始。



**17）数组连接concat（）**

concat（）方法用于连接两个或多个数组，此方法返回一个新数组，不改变原来的数组。

arrayObject.concat(array1,array2,...,arrayN)

该方法不会改变现有的数组，而仅仅会返回被连接数组的一个副本。

**18）指定分隔符连接数组元素join（）**

用于把数组中的所有元素放入一个字符串。元素是通过指定的分隔符进行分割的。

arrayObject.join(分隔符)

注意：返回一个字符串，该字符串把数组中的各个元素串起来，用分隔符置于元素与元素之间。这个方法不影响数组原本的内容。

**19）颠倒数组元素顺序reverse（）**

该方法会改变原来的数组，而不会创建新的数组。

1. 选定元素slice()

从已有的数组中返回选定的元素。

arrayObject.slice(start,end)

返回一个新的数组，包含从Start到end(不包含该元素)的arrayObject中的元素；该方法不会修改数组，而是返回一个子数组。

String.slice（）和Array.slice（）类似。

**20）数组排序sort()**

arrayObject.sort(方法函数)。

使数组中的元素按照一定的顺序排列。

如果不指定<方法函数>，则按unicode码顺序排列。

myArray.sort(sortMethod);

注意: 该函数要比较两个值，然后返回一个用于说明这两个值的相对顺序的数字。比较函数应该具有两个参数 a 和 b，其返回值如下：

若返回值<=-1，则表示 A 在排序后的序列中出现在 B 之前。  
  若返回值>-1 && <1，则表示 A 和 B 具有相同的排序顺序。  
  若返回值>=1，则表示 A 在排序后的序列中出现在 B 之后。

## 浏览器对象

**1）window对象**

Window对象是BOM对象的核心，window对象指当前的浏览器窗口。



1. **JavaScript计时器**

在JavaScript中，可以在设定的时间间隔之后来执行代码，而不是在函数被调用后立即执行。

计时器类型：一次性计数器，仅在指定的延迟时间之后触发一次；间歇性触发计时器，每隔一定的时间间隔就触发一次。



1. **计时器setInterval()**

在执行时，从载入页面后每隔指定的时间执行代码。

语法：setInterval(代码，交互时间);

代码：要调用的函数或要执行的代码串。

交互时间：周期性执行或调用表达式之间的时间间隔，以毫秒计。

返回值：一个可以传递给clearInterval()从而取消对“代码”的周期性执行的值。

函数调用格式（假设有一个clock（）函数）

setInterval(“clock()”,1000);或setInterval(clock,1000);

1. **取消计时器clearInterval()**

clearInterval()方法可取消由setInterval()设置的交互时间。

语法：clearInterval(id\_of\_setInterval);

1. **计时器setTimeOut()**

setTimeOut()计时器，在载入后延迟指定时间后，去执行一次表达式，仅执行一次。

setTimeOut(代码，延迟时间);

1. **取消计时器clearTimeOut()**

setTimeOut()和clearTimeOut()一起使用，停止计时器。

语法：clearTimeouo(id\_of\_setInterval);

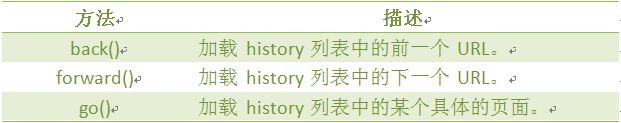
1. **History对象**

History对象记录用户曾经浏览过的页面（URL），并可以实现浏览器前进与后退相似导航的功能。

注意：从窗口被打开的那一刻开始记录，每个浏览器窗口、每个标签页乃至每个框架，都有自己的history对象与特定的window对象关联。

语法：window.history.[属性|方法]。Window可以省略





1. **返回前一个浏览的页面back()**

等同于点击浏览器的倒退按钮， go(-1)；

1. **返回下一个浏览的页面forward()**

等价点击前进按钮，go(1)；

**10）返回浏览历史中的其他页面go()**

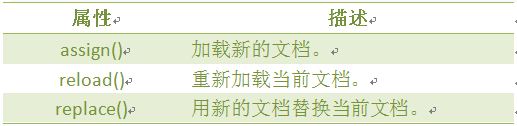
**11）Location对象**

Location用于获取或设置窗体的URL，并且可以用于解析URL。

location.[属性|方法]

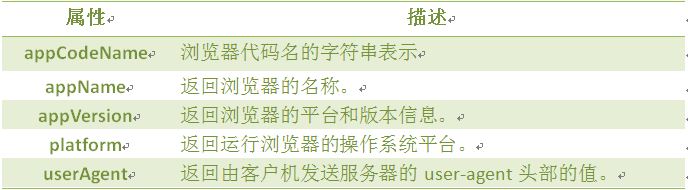






**12）Navigator对象**

Navigator对象包含有关浏览器的信息，通常用于检测浏览器与操作系统版本。



**13）userAgent**

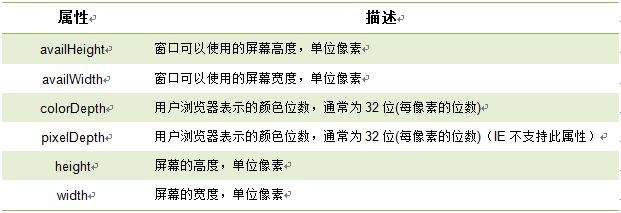
返回用户代理头的字符串表示（就是包括浏览器版本信息等的字符串）。

语法：navigator.userAgent

**14）screen对象**

Screen对象用户获取用户的屏幕信息。

语法：window.screen.属性



**15）屏幕分辨率的高和宽**

Window.screen对象包含有关用户屏幕的信息。

**16）屏幕可用高和宽度**

Screen.availWidth属性返回访问者屏幕的宽度，以像素计，减去界面特性，比如任务栏。

注意：不同系统的任务栏默认高度不一样，及任务栏的位置可在屏幕上下左右任何位置。

## Dom对象，控制HTML标签

**1）认识Dom**

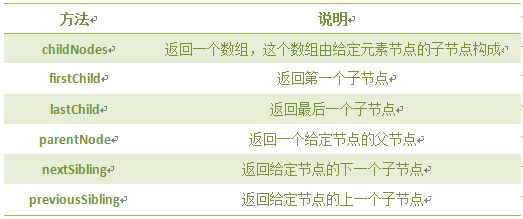
文档对象模型DOM(document object module)定义访问和处理HTML文档的标准方法。DOM将HTML文档呈现为带有元素、属性和文本的树结构（节点树）。



节点属性：



遍历节点树：



DOM操作：



**2）getElementsByName()方法**

返回带有指定名称的节点对象的集合。

document.getElementsByName(name)

1. **getElementsByTAGName()方法**

document.getElementsByTagName(Tagname)

1. **getElementById()，getElementsByName()，getElementsByTagName()区别**

ID是唯一的，name可以重复，TagName可以看作某类，如小孩类。

1. **getAttribute()方法**

通过元素节点的属性名称获取属性的值。

elementNode.getAttribute(name)

**6）setAttribute()方法**

setAttribute()方法增加一个指定名称和值的新属性，或者把一个现有的属性设定为指定的值。

elementNode.setAttribute(name,value)

注意：把指定的属性设置为指定的值，如果不存在指定名称的属性，该方法将创建一个新属性。

**7）节点属性**

在文档对象模型（DOM）中，每个节点都是一个对象。DOM节点有三个重要的属性：nodeName节点名称，nodeValue节点值，nodeType节点类型。

nodeName属性：节点的名称，是只读的。

* 元素节点的nodeName与标签名相同；
* 属性节点的nodeName是属性的名称；
* 文本节点的nodeName永远是#text;
* 文档节点的nodeName永远是#document。

nodeValue属性：节点的值。

* 元素节点的nodeValue是undefined或null；
* 文本节点的nodeValue是文本本身；
* 属性节点的nodeValue是属性的值。

nodeType属性：节点的类型，是只读的。常用节点类型如下：

元素类型 节点类型

元素 1

属性 2

文本 3

注释 8

文档 9

**8）访问子节点的childNodes**

访问指定元素节点下的所有子节点列表，返回的值可以看作是一个数组，具有length属性。

elementNode.childNodes

注意：如果选定的节点没有子节点，则该属性返回不包含节点的nodeList。

**9）访问子节点的第一和最后项**

Node.firstChild node.lastChild

Internet Explorer会忽略节点之间生成的空白文本节点，而其他浏览器不会。可以通过检测节点类型，过滤子节点。

**10）访问父节点parentNode**

elementNode.parentNode

父节点只能有一个。

**11）访问兄弟节点nextSibling，previousSibling**

NextSibling属性可返回某个节点之后紧跟的节点（处于同一树层级中）。

NodeObject.nextSibling，如果无此节点，则该属性返回null。

**12）插入节点appendChild**

在指定节点的最后一个子节点列表之后添加一个新的子节点。

appendChild(newNode)

**13）插入节点insertBefore()**

insertBefore()方法可在已有的子节点前插入一个新的子节点。

insertBefore(newnode,node).

**14）删除节点removeChild()**

removeChild()方法从子节点列表中删除某个节点。如删除成功，此方法可返回被删除节点，如失败，则返回null。

Var x=NodeObject.removeChild(node).

该方法删除的节点在内存中，可通过x进行操作，若要完全删除，给x赋值null。

**15）替换元素节点replaceChild()**

该方法实现子节点的替换，返回被替换对象的引用。

Node.replaceChild（newNode,oldNode）;

**16）创建元素节点createElement**

返回一个Element对象。

document.createElement(tagName)

**17）创建文本节点createTextNode**

document.createTextNode(data)

**18）浏览器窗口可视区域大小**

获得浏览器窗口的尺寸。

一、对于IE9+、Chrome、Firefox、Opera 以及 Safari：

•  window.innerHeight - 浏览器窗口的内部高度

•  window.innerWidth - 浏览器窗口的内部宽度

二、对于 Internet Explorer 8、7、6、5：

•  document.documentElement.clientHeight表示HTML文档所在窗口的当前高度。

•  document.documentElement.clientWidth表示HTML文档所在窗口的当前宽度。

或者

Document对象的body属性对应HTML文档的<body>标签

•  document.body.clientHeight

•  document.body.clientWidth

在不同浏览器都实用的 JavaScript 方案：

var w= document.documentElement.clientWidth

|| document.body.clientWidth;

var h= document.documentElement.clientHeight

|| document.body.clientHeight;

**19）网页尺寸scrollHeight**

scrollHeight和scrollWidth，获取网页内容高度和宽度。

一、针对IE、Opera:

scrollHeight 是网页内容实际高度，可以小于 clientHeight。

二、针对NS、FF:

scrollHeight 是网页内容高度，不过最小值是 clientHeight。也就是说网页内容实际高度小于 clientHeight 时，scrollHeight 返回 clientHeight 。

三、浏览器兼容性

var w=document.documentElement.scrollWidth

|| document.body.scrollWidth;

var h=document.documentElement.scrollHeight

|| document.body.scrollHeight;

注意:区分大小写

scrollHeight和scrollWidth还可获取Dom元素中内容实际占用的高度和宽度。

**20）网页尺寸offsetHight**

offsetHeight和offsetWidth，获取网页内容高度和宽度(包括滚动条等边线，会随窗口的显示大小改变)。

一、值

offsetHeight = clientHeight + 滚动条 + 边框。

二、浏览器兼容性

var w= document.documentElement.offsetWidth

|| document.body.offsetWidth;

var h= document.documentElement.offsetHeight

|| document.body.offsetHeight;