html（HyperText Mark-up Language）超文本标记语言

超文本指的是超链接，标记指的是标签

<!DOCTYPE html> #文档声明

<html lang="en"> #“<html>”和“</html>”定义html的整体

<head> #“<head>”和“<body>”是第一层子元素

<meta charset="UTF-8">

<title>网页标题</title>

</head>

<body>

网页显示内容

</body>

</html>

‘lang=“en”定义网页的语言为英文，定义成中文是'lang="zh-CN"',不定义也没什么影响

“<head>”标签里面负责对网页进行一些设置以及定义标题，设置包括定义网页的编码格式，外链css样式文件和javascript文件等，设置的内容不会显示在网页上，标题的内容会显示在标题栏，“<body>”内编写网页上显示的内容。

HTML文档类型：目前常用的两种文档类型是xhtml 1.0和html5

html注释

html文档代码中可以插入注释，注释是对代码的说明和解释，注释的内容不会显示在页面上，html代码中插入注释的方法是：

<!-- 这是一段注释 -->

html标签特点：

html的标签大部分是成对出现的,少量是单个出现的，特定标签之间可以相互嵌套，包含的标签和父标签可以是同类型的，也可以是不同类型的

html标题

通过 <h1>、<h2>、<h3>、<h4>、<h5>、<h6>,标签可以在网页上定义6种级别的标题。6种级别的标题表示文档的6级目录层级关系

段落

<p>标签定义一个文本段落，一个段落含有默认的上下间距，段落之间会用这种默认间距隔开

换行

代码中成段的文字，在代码的段落中插入<br />来强制换行

字符实体

代码中成段的文字，如果文字间想空多个空格，在代码中空多个空格，在渲染成网页时只会显示一个空格，如果想显示多个空格，可以使用空格的字符实体：&nbsp;

在网页上显示 “<” 和 “>” 会误认为是标签，想在网页上显示“<”和“>”可以使用它们的字符实体，“<” 和 “>” 的字符实体为 &lt;和 &gt;

html块

1、div标签 块元素，表示一块内容，没有具体的语义。

2、span标签 行内元素，表示一行中的一小段内容，没有具体的语义。

含样式和语义的标签

1、em标签 行内元素，表示语气中的强调词

2、i标签 行内元素，原本没有语义，w3c强加了语义，表示专业词汇

3、b标签 行内元素，原本没有语义，w3c强加了语义，表示文档中的关键字或者产品名

4、strong标签 行内元素，表示非常重要的内容

语义化的标签

语义化的标签，就是在布局的时候多使用语义化的标签，搜索引擎在爬网的时候能认识这些标签，理解文档的结构，方便网站的收录。比如：h1标签是表示标题，p标签是表示段落，ul、li标签是表示列表，a标签表示链接，dl、dt、dd表示定义列表等，语义化的标签不多。

html图像

<img>标签可以在网页上插入一张图片，它是独立使用的标签，通过“src”属性定义图片的地址，通过“alt”属性定义图片加载失败时显示的文字，以及对搜索引擎和盲人读屏软件的支持

html链接

<a>标签可以在网页上定义一个链接地址，通过src属性定义跳转的地址，通过title属性定义鼠标悬停时弹出的提示文字框。

<a href="#"></a> <!-- # 表示链接到页面顶部 -->

页面内定义了“id”或者“name”的元素，可以通过a标签链接到它的页面滚动位置，前提是页面要足够高，有滚动条，且元素不能在页面顶部，否则页面不会滚动。

html列表

有序列表 网页上生成的列表，每条项目上会按1、2、3编号

<ol>

<li>列表文字一</li>

<li>列表文字二</li>

</ol>

无序列表 生成无编号符号

<ul>

<li>列表文字一</li>

<li>列表文字二</li>

</ul>

定义列表：用于术语的定义。

<h3>前端</h3>

<dl> <dl>标签表示列表的整体。

<dt>html</dt> <dt>标签定义术语的题目。

<dd>负责页面的结构</dd> <dd>标签是术语的解释。

</dl> 一个<dl>中可以有多个题目和解释

HTML表格

table常用标签

1、table标签：声明一个表格

2、tr标签：定义表格中的一行

3、td和th标签：定义一行中的一个单元格，td代表普通单元格，th表示表头单元格

html表单

表单用于搜集不同类型的用户输入，表单由不同类型的标签组成，实现一个特定功能的表单区域（比如：注册），首先应该用<form>标签来定义表单区域整体，在此标签中再使用不同的表单控件来实现不同类型的信息输入

html内嵌框架

<iframe>标签会创建包含另外一个html文件的内联框架（即行内框架），src属性来定义另一个html文件的引用地址，frameborder属性定义边框，scrolling属性定义是否有滚动条

<iframe src="http://www..." frameborder="0" scrolling="no"></iframe>

a标签的target属性可以将链接到的页面直接显示在当前页面的iframe中

<a href="01.html" target="myframe">页面一</a>

<iframe src="01.html" frameborder="0" scrolling="no" name="myframe"></iframe>

CSS（Cascading Style Sheets，层叠样式表）

html只负责文档的结构和内容，表现形式完全交给CSS

css的定义方法：选择器 { 属性:值; 属性:值; 属性:值;}

选择器是将样式和页面元素关联起来的名称，属性是希望设置的样式属性每个属性有一个或多个值。div{ width:100px; height:100px; color:red }

css页面引入方法：

1、外联式：通过link标签，链接到外部样式表到页面。

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

2、嵌入式：通过style标签，在网页上创建嵌入的样式表。

<style type="text/css">

div{ width:100px; height:100px; color:red }

......

</style>

3、内联式：通过标签的style属性，在标签上直接写样式。

<div style="width:100px; height:100px; color:red ">......</div>

css选择器

标签选择器

标签选择器，此种选择器影响范围大，建议尽量应用在层级选择器中。

id选择器

通过id名来选择元素，元素的id名称不能重复，所以一个样式设置项只能对应于页面上一个元素，不能复用，id名一般给程序使用，所以不推荐使用id作为选择器。

类选择器

通过类名来选择元素，一个类可应用于多个元素，一个元素上也可以使用多个类，应用灵活，可复用，是css中应用最多的一种选择器。

.red{color:red} //定义到head里

.big{font-size:20px}

.mt10{margin-top:10px}

<div class="red">....</div> // body中引用

<h1 class="red big mt10">....</h1>

<p class="red mt10">....</p>

层级选择器

主要应用在选择父元素下的子元素，或者子元素下面的子元素，可与标签元素结合使用，减少命名，同时也可以通过层级，防止命名冲突。

.box span{color:red}

.box .red{color:pink}

.red{color:red}

<div class="box">

<span>....</span>

<a href="#" class="red">....</a>

</div>

<h3 class="red">....</h3>

组选择器

多个选择器，如果有同样的样式设置，可以使用组选择器。

.box1,.box2 {width:100px;height:100px}

.box1{background:red}

.box2{background:pink}

<div class="box1">....</div>

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

伪类及伪元素选择器

常用的伪类选择器有hover，表示鼠标悬浮在元素上时的状态，伪元素选择器有before和after,它们可以通过样式在元素中插入内容。

.box1:hover{color:red}

.box2:before{content:'行首文字';}

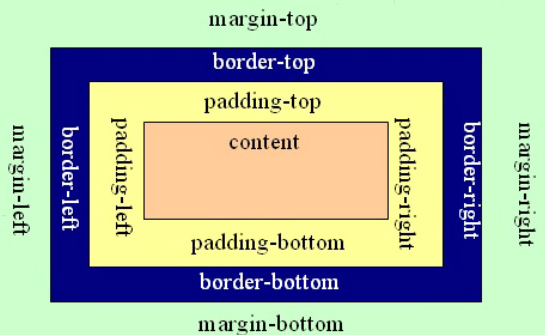
.box3:after{content:'行尾文字';}

<div class="box1">....</div>

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

<div class="box3">....</div>

css盒子模型



设置对应的样式分别为：盒子的边框(border)、盒子内的内容和边框之间的间距(padding)、盒子与盒子之间的间距(margin)。

设置边框

border-top:10px solid red;

四个边如果设置一样，可以将四个边的设置合并成一句：border:10px solid red;

设置内间距padding

padding：20px 40px 50px 30px; /\* 四个值按照顺时针方向，分别设置的是 上 右 下 左

padding：20px 40px 50px; /\* 顶部内边距为20px，左右内边距为40px，底部内边距为50px

padding：20px 40px; /\* 设置上下内边距为20px，左右内边距为40px

padding：20px; /\* 设置四边内边距为20px

设置外间距margin 和padding的设置方法相同

盒子模型的尺寸

盒子宽度 = width + padding左右 + border左右

盒子高度 = height + padding上下 + border上下

块元素

元素就是标签，块元素也称为行元素，如：div、p、ul、li、h1~h6、dl、dt、dd等等。

支持全部的样式

如果没有设置宽度，默认的宽度为父级宽度100%

盒子占据一行、即使设置了宽度

内联元素

也可以称为行内元素，如：a、span、em、b、strong、i等等

支持部分样式（不支持宽、高、margin上下、padding上下）

宽高由内容决定

盒子并在一行

代码换行，盒子之间会产生间距

子元素是内联元素，父元素可以用text-align属性设置子元素水平对齐方式，用line-height属性值设置垂直对齐方式

解决内联元素间隙的方法

1、去掉内联元素之间的换行

2、将内联元素的父级设置font-size为0，内联元素自身再设置font-size

内联块元素

也叫行内块元素，是新增的元素类型，img和input元素的行为类似这种元素，但是也归类于内联元素，可以用display属性将块元素或者内联元素转化成这种元素

支持全部样式

如果没有设置宽高，宽高由内容决定

盒子并在一行

代码换行，盒子会产生间距

子元素是内联块元素，父元素可以用text-align属性设置子元素水平对齐方式，用line-height属性值设置子元素垂直对齐方式

display属性

display属性是用来设置元素的类型及隐藏的，常用的属性有：

1、none 元素隐藏且不占位置

2、block 元素以块元素显示

3、inline 元素以内联元素显示

4、inline-block 元素以内联块元素显示

浮动

浮动元素有左浮动(float:left)和右浮动(float:right)两种

background属性

给盒子设置背景图片和背景颜色

background-color 设置背景颜色

background-image 设置背景图片地址

background-repeat 设置背景图片如何重复平铺

background-position 设置背景图片的位置

background-attachment 设置背景图片是固定还是随着页面滚动条滚动

background: #00FF00 url(bgimage.gif) no-repeat left center fixed

**JavaScript**

JavaScript主要解决的是前端与用户交互的问题，包括使用交互与数据交互。 JavaScript是浏览器解释执行的

前端三大块

1、HTML：页面结构

2、CSS：页面表现：元素大小、颜色、位置、隐藏或显示、部分动画效果

3、JavaScript：页面行为：部分动画效果、页面与用户的交互、页面功能

JavaScript嵌入页面的方式

1、行间事件（主要用于事件）

<input type="button" name="" onclick="alert('ok！');">

2、页面script标签嵌入

<script type="text/javascript">

var a = '你好！';

alert(a);

</script>

3、外部引入

<script type="text/javascript" src="js/index.js"></script>

javascript语句与注释

1. 一条javascript语句应该以“;”结尾
2. javascript注释

// 单行注释

var a = 123;

/\*

多行注释

1、...

2、...

\*/

变量

定义变量需要用关键字 'var'。同时定义多个变量可以用","隔开，公用一个‘var’关键字

var c = 45,d='qwe',f='68';

变量类型

5种基本数据类型：number、string、boolean、undefined、null

1种复合类型：object

变量、函数、属性、函数参数命名规范

1、区分大小写

2、第一个字符必须是字母、下划线（\_）或者$

3、其他字符可以是字母、下划线、$或数字

获取元素

可以使用内置对象document上的getElementById方法来获取页面上设置了id属性的元素，获取到的是一个html对象，然后将它赋值给一个变量

....

<div id="div1">这是一个div元素</div>

....

<script type="text/javascript">

var oDiv = document.getElementById('div1');

</script>

</body>

如果把javascript写在元素的上面，就会出错，因为页面上从上往下加载执行的，javascript去页面上获取元素div1的时候，元素div1还没有加载，解决方法有两种：

第一种方法：将javascript放到页面最下边

第二种方法：将javascript语句放到window.onload触发的函数里面,获取元素的语句会在页面加载完后才执行，就不会出错了。

window.onload = function(){

var oDiv = document.getElementById('div1');

}

....

<div id="div1">这是一个div元素</div>

操作元素属性

操作属性的方法

1、“.” 操作

2、“[ ]”操作

属性写法

1、html的属性和js里面属性写法一样

2、“class” 属性写成 “className”

3、“style” 属性里面的属性，有横杠的改成驼峰式，比如：“font-size”，改成”style.fontSize”

通过“.”操作属性：

<script type="text/javascript">

window.onload = function(){

var oInput = document.getElementById('input1');

var oA = document.getElementById('link1');

// 读取属性值

var val = oInput.value;

var typ = oInput.type;

var nam = oInput.name;

var links = oA.href;

// 写(设置)属性

oA.style.color = 'red';

oA.style.fontSize = val;

}

</script>

<input type="text" name="setsize" id="input1" value="20px">

<a href="http://www.itcast.cn" id="link1">传智播客</a>

通过“[ ]”操作属性：

<script type="text/javascript">

window.onload = function(){

var oInput1 = document.getElementById('input1');

var oInput2 = document.getElementById('input2');

var oA = document.getElementById('link1');

// 读取属性

var val1 = oInput1.value;

var val2 = oInput2.value;

// 写(设置)属性

// oA.style.val1 = val2; 没反应

oA.style[val1] = val2;

}

</script>

......

<input type="text" name="setattr" id="input1" value="fontSize">

<input type="text" name="setnum" id="input2" value="30px">

<a href="http://www.itcast.cn" id="link1">传智播客</a>

innerHTML

innerHTML可以读取或者写入标签包裹的内容

<script type="text/javascript">

window.onload = function(){

var oDiv = document.getElementById('div1');

//读取

var txt = oDiv.innerHTML;

alert(txt);

//写入

oDiv.innerHTML = '<a href="http://www.itcast.cn">传智播客<a/>';

}

</script>

......

<div id="div1">这是一个div元素</div>

函数：重复执行的代码片