**HTML（Hyper Text Markup Language 超文本标记语言） 概述**

**Hyper Text 超文本: 具备特殊功能的文本。**

普通文本: a，小写字母a

超文本: a，功能为超链接

**Markup 标签: 规定了超文本的组成形式。**

超文本a: <a></a>

**Language 语言: 具备特定的语法规范，**

HTML是网页设计结构的语言，由HTML语言编写的文件以 '.html' 或 '.htm' 结尾，并且由浏览器负责运行。

**HTML开发工具：**

**记事本、超级记事本(Notepad++、Editplus)等。**

建议初期使用，手动敲一个一个代码，有助于加深对代码的记忆和理解

**WebStorm(web神器，特别是对js的提醒功能受到广大开发者的欢迎)、sublime text 3、HBuilder等。**

建议后期使用，有了一定的代码寄出之后，这些工具可大大提升我们的开发速度，还可避免编写错误。

**运行环境：各款浏览器。**

**个人学习过程中总结HTML一些特别之处**

**HTML的标签规范：**

编写HTML的标签语法时，请一定遵守w3c规范，建议编写标签时一律用小写字母，封闭类型标签是成对出现的(例如：<div></div>)。非封闭类型又称为空标签或单标签，要记得添加结束标签(例如：<img />)，虽然你不加结束标签浏览器也能够读取处理，这是因为浏览器内核自动帮你添加的标签，从而浏览器能够识别出你编写的未封闭标签，从而加大了浏览器的负担。在HTML语言的所有标签中，除了<!Doctype>标签之外，所有的标签都要有结束标签。

**HTML标签之间的嵌套：**

HTML标签之间允许嵌套标签，形成更为复杂的语法结构，避免页面布局标签的单一化，但是有一个前提：行内、行内块标签不能嵌套块级标签，块级标签可以嵌套行内、行内块标签和块级标签。还有就是标签的嵌套格式一定要正确，正确的格式为：一个块级标签可以在标签内部嵌套一个或多个行内、行内块标签货块级标签（例如：<div><p><span></span></p></div>）。何为标签内部？即开始标签和结束标签之间的部分(例如：<div>标签内部</div>)。注意：这里有一个奇葩的块级标签，p></p><标签，它不能嵌套块级标签，只能嵌套行内、行内块标签。

**<!Doctype>标签的作用：**

要说这个标签的作用，还是得诉说一下历史。在HTML5还未出来之前，虽然说那个时候w3c也制定了一定的标准，但是各个浏览器产商对这个标签的使用都有自家风格，程序员书写代码的风格也各具特彩，这可是个展示个人风格的时代，各浏览器产商和各程序员之间碰撞出强烈的火花，谁都不服谁。直到HTML5出来之后，这下好了，这可是个大福音啊，各浏览器产商被w3c新出炉的HTML5标准收拾的服服帖帖的，大家都开始实行HTML5标准，谁不向HTML5标准靠齐，可能意味着被市场抛弃，就连N多年前牛逼哄哄的IE都被折服了，微软开始兼容w3c的标准，IE每升级一次版本，就意味着在之前版本的基础上要多加一个内核，N多年前做的孽终究是要还的，风光过后，只能默默地低头舔伤口，最后实在是受不了了，通过升级win10，创建了一个Edge浏览器，彻底抛弃了IE，拥抱w3c，免费为用户升级到win10系统，就是为了让用户使用Edge浏览器，避免因为以前的IE浏览器而被用户抛弃，从而免去被淘汰的风险，虽然win10确实是挺好用的，当然这是外话了。

下面来说一下HTML5的新标准。说是新标准吧，也说的过去，但是更准确的说是对旧有标准的极大包容和添加了一些新标准并去掉了一些旧有的标准，因为w3c实行这么多年的标准都无法普遍化，所以w3c干脆就在包容各浏览器产商就标准的前提上提出了一些新标准并去掉了一些旧有的标准，俗话说得好，我给你一个台阶了，你也得还回来吧，所以个浏览器产商就想HTML5标准看齐了。不过也是HTML5的新标准确实好使，在很大程度上简化了程序员的开发难度，所以也受到广大程序员的喜爱，自然而然各浏览器产商也就向HTML5的新标准靠齐了，具体HTML5新标准会在HTML5这一栏讲到。

好了，说了那么多的历史，也是时候该上点正菜了，现在说说<!Doctype>标签的作用。该标签指定了文档的类型和版本，放置在html页面的顶端，该标签有两种模式，分别是标准模式和混杂模式。

何为严格模式？又称标准模式，就是根据w3c制定的标准去解析代码。

如何触发标准模式？在<!Doctype>标签中添加正确的文档声明。

何为混杂模式？又称怪异模式或兼容模式，就是各个浏览器根据自己内核的特性去解析代码。

如何触发混杂模式？在<不使用!Doctype>标签或在!Doctype>标签中使用格式不正确的文档声明。

若使用了标准模式，那么相同网页，在各浏览器上显示都是同样的页面效果。

若使用了混杂模式，那么相同网页，在各浏览器上显示可能是不同的页面页面。

所以，强烈建议程序员在编写代码时，要在<!Doctype>标签中添加文档声明，毕竟现在个浏览器产商基本上都实现了HTML5的新标准，这是整合的趋势，不管是浏览器产商亦或是程序员，谁若还是依然使用以前的混杂模式，那么将意味着会面临被淘汰的风险。

这两种模式的具体说明，请看前辈的文章 <http://www.cnblogs.com/shenxiaolin/p/5425249.html>

**元数据meta标签必写：**

meta元素可提供有关页面的元信息，比如页面编码、针对搜索引擎的描述和关键词等。它位于文档的头部，是空标记，使用属性来定义与文档相关联的名称/值对。说到这个meta标签呢，还得说一下搜索引擎。何为搜索引擎呢，其实就是我们平常在浏览器上想搜索东西时在那个输入框中输入东西，搜索引擎就会根据我们输入的内容去查找包含输入内容的标题或描述信息等，说白了就是帮我们查找我们想找的内容的一个工具,常用的搜索引擎有谷歌、百度、必应等。说道搜索引擎，就不得不谈一下SEO(Search Engine Optimization 搜索引擎优化)技术了，SEO是指通过站内优化比如网站结构调整、网站内容建设、网站代码优化等以及站外优化，比如网站站外推广、网站品牌建设等，使网站满足搜索引擎收录排名需求，在搜索引擎中提高关键词排名，从而吸引精准用户进入网站，获得免费流量，产生直接销售或品牌推广。然而，我们实现SEO技术的前提是要在meta标签中写详细的内容，怎么写这个内容？这就要求你对本网站主题有很深的了解，从而判断用户要查找这一类型的主题可能会输入的短关键字或长段文字，所有我们能够想到的这些信息都得写上个，meta标签中有两个属性是专门针对SEO的，即<meta name="keywords" content="关键词1,关键词2,关键词3,关键词4,关键词5">: 定义网页关键字 和 <meta name="description" content="描述信息">: 定义网页描述信息。不过在meta标签里面写的内容也有字数限制，关键词数量最好保持在5个以内，关键词要和页面内容相关。Description用来告诉搜索引擎你的网站或该页面的主要内容，描述内容要和页面内容相关，字数在150个字以内。而我们在meta标签里面写的内容能够帮助我们更好的被搜索引擎查找到，从而增大被用户查询点击的次数，次数越大，反而会被搜索引擎更容易推荐给用户，虽然说搜索引擎排名是要用钱去买的，但是我们在meta标签内写的内容要趋向于SEO的标准，还是有点效果的。

还得说一下meta标签中设置编码需要注意的地方:

meta标签是通过charset属性来设置编码的，这个属性的作用是为了让浏览器根据编码来翻译网页的语言，让用户能够看到自己熟悉的语言，便于阅读。例如：下面说一下可能经常要设置的一些编码：

1. 国际通用编码：utf-8
2. 简体中文: gb2312、gbk
3. 繁体中文: gbk
4. 中国少数名族语言：gb18030

具体meta标签详解，请查看前辈的文章 <http://blog.csdn.net/neubuffer/article/details/16900749>

**设置title标签要慎重,一经设置请勿轻易修改，因为它就是定站神针。**

title标签是网页的眼睛，是用户进入网页的入口，极其重要。虽然一个完整的网站可能会有成百上千个页面，伴随着每个网页的诞生，自然而然每张网页都有自己的标题。在你的网站还没有名气的时候，如何推销自己，让用户能够在未记住你网站域名的情况下记住你的网站名称，这时候就看你怎么发挥title标签的威力了。

title威力炮就应该发在首页上，首页是一个网站的门户，众多程序员和美工倾尽所能的去美化首页，就是希望能够给用户留下一个好的印象，用户一旦对你这个网页有了好感，自然而然的会去看浏览器的标题，记住你的网站名称，甚至会把你的网页添加到收藏夹，那么，你就成功了一步。你能不能深深的牵住用户的心，就看你网站的东西能不能一直吸引着用户。这就是每过一段时间，很多网站都会改版的原因所在，一些网站会通过改变网页的新鲜感来吸引用户，这也不失为一条妙计。

title标签应该怎样设置？

大家日常搜索应该也会有这样子的经验，一般搜索结果相关性强且关键词匹配结构接近吻合的，往往是关键词靠前的排名会比较靠前。为什么会是这样？因为百度发布搜索引擎2.0提到的关键词应该尽可能靠左靠前，这样比较容易提升你的排名。

**link VS @import**

两者的共同点：都是导入外部的css文件。

两者的不同点：

1. link标签是HTML标签，除了加载css文件之外，还可以定义rel等其它事务；

@import是CSS提供的，@import就只能加载CSS。

1. link基本无兼容性问题；

@import是CSS2.1提出的，低版本浏览器不支持，只在IE5.5以上才能被识别。

1. link引用css时，在页面载入时同时加载；

@import需要等待页面被加载完再加载。

1. link方式的样式的权重 高于@import的权重
2. 补充一点：

外联的CSS加载顺序是最早的，外联的话，你的HTML文件会相当整洁，结构清晰，方便编码。 内联的CSS加载顺序较慢，定在内联里面的CSS代码会使得HTML页面相当乱，违背了单一责任原则，不便于维护。

1. 最后，若无特别需求，强烈建议使用外联的link标签嵌入css文件。

**header中的script VS body中的script**

两者的共同点：都可导入外部的js文件和直接在HTML文件内部中编写js文件。

两者的不同点：

1. 加载速度：

外部导入的js文件比内部编写的js文件加载速度快的多；

1. 从页面代码的整洁程度上说：

外联的话，你的HTML文件会相当整洁，结构清晰，方便编码 定在内联里面的js代码会使得HTML页面相当乱，违背了单一责任原则，不便于维护。

1. 网页的加载顺序：

外联js文件先于HTML文件主体加载，然后是header中的js文件，最后才是body中的js文件。

1. 对页面结构的影响：

有的时候，我们需要执行一些DOM操作，这就得操作页面上的结构; 而header中的js文件是先于HTML文件主体加载，势必会造成页面布局混乱  
而放在body中最底部的js文件是随着HTML文件加载完成才加载js文件的，这样子就不会早餐布局混乱。

1. 最后，若无特别需求，强烈建议在body中最底部嵌入外联的js文件。

**行内元素、行内块元素和块级元素的区别:**

* 行内元素：不会换行，可以和其它行内元素在一行内共存，在浏览器中显示时不会换行,无法设置宽高，大部分的行内元素允许被修改宽高（例如：span标签、img标签等）。
* 行内块元素：拥有行内元素的不换行功能和块级元素可设置宽高功能，行内块元素时行内元素的一个特殊的子集，一些行内块元素无法修改宽高度（例如：表单控件元素、单选按钮、复选框等）。
* 块级元素：块级元素前后都会换行，默认独占一行，所有的块级标签均可设置宽高（例如：div标签、p标签等）。

我们可以这样理解：如果元素是块级元素的话，则总是在新行上显示，就像是书中的一个新段落；而如果元素是行内元素，则只能在当前行内显示，就像是段落中的一个词组。

因此，块级元素常被用来构建网页上的大体结构，用于包含其它行内元素和文本；而行内块元素常被用来构建网页上的小体结构，只能包含行内元素和文本。

**行内元素、行内块元素和块级元素如何互转？**

给行内元素设置宽高是无效的，只能靠里面的内容撑开，内容的宽高即使行内元素的宽高；而给块级元素设置宽高是有效的。如何能够有效的设置行内元素的宽高呢？只有两个办法，把行内元素升级为行内块元素或块级元素。

1. 块级元素转行内元素 --> display: inline;
2. 行内元素转行内块元素 --> display: inline-block;
3. 行内元素转块级元素 --> display: block;
4. 如果行内元素的父类元素是块级元素，可设置 --> display:inherit;继承父类的块级属性，等同于display: block;

**访问资源文件路径需要注意的一些地方:**

访问资源文件的路径有三种方式：

* 绝对路径：从文件所在的最高级目录下开始的完整路径，分为两类：
  1. 访问互联网上的资源：(ex: https://www.baidu.com/img/logo.png); https:协议名，www.baidu.com:主机名，img:文件目录，logo.png:文件名。协议名同主机名之间用" :// "分隔，从主机名到后续的内容使用" / "分离。
  2. 访问本级资源：即从盘符开始直到你当前的文件所经过的路径。(ex: d:\project\img\1.jpg); d:盘符，project:目录1，img:目录1.1 ，1.jpg:当前文件；除了盘符于目录1之间用" :\或:/ "分隔之外，从目录1到后续的内容都使用" \或/ "分隔。

* 相对路径：指从当前文件所在位置开始查找资源文件所在位置所经过的路径，分三类：
  1. 资源文件夹和当前文件夹处于同目录下，直接通过资源文件夹名称进行引用；
  2. 资源文件夹位于当前文件夹的子目录中，先进入当前文件夹，再引用；，
  3. 资源文件夹在当前文件夹的父级目录中，先返回父级目录，再引用。

这里补充一点：当前文件夹找当前文件夹子目录（使用" / "）；当前文件夹找上一级目录(使用" ../ ")。

* 根相对路径：是指相对于我们自定义的根目录，如果说我们网站的根目录是web，不管现在在哪个目录下，都从web目录开始访问资源文件，根目录就表示为" ../ "(ex: ../image/flower.png)。
* 总结：绝对路径和相对路径一般在客户端使用，根相对路径一般在服务器端使用；最后，强烈建议把以上路径中的" / "均转为" \ ",因为Linux系统不支持" / ",但是window两者都支持。

**一些标签需要注意的地方:**

1. img标签中有四个必写属性，分别为src、alt、width、height，不过width和height一般在css中定义，src表示请求图片的路径，alt表示请求图片失败时所显示的描述性文本，alt属性对用户体验效果是挺好的，而且对于盲人来说，alt属性是它们了解图片内容的唯一方式。alt属性的值是一个最多可以包含1024个字符的字符串，其中包括空格和标点。字符串必须包含在引号内。alt文本中还可以包含对特殊字符的实体引用，但是不允许包含其它类别的标记，尤其是不允许有任何标签和样式。

1. a标签href属性的常见表达形式：
   * 普通超链接：<a href='链接地址' >文本</a>
   * 下载链接：<a href='xxx.zip' >文本</a>
   * 邮件链接：<a href='mail:xxx@126.com'>文本</a>
   * 空链接：<a href='#' >文本</a>
   * 链接到JavaScript：<a href='javascript:...;'>文本</a>

当我们需要一个空链接的a标签时，很多人可能都会想到使用" # "，但是有个问题，使用" # "点击a标签时会跳转到页面头部，而且会在地址栏后面添加" # "，地址栏添加的" # "会使我们传递数据到服务器时获取数据出错等等，而且这也会使用户感到很奇怪或愤怒，我想这都不是我们的本意，所以我说一下解决办法,在href中使用javascript:void(0);可避免这一状况。

注意：ol标签和ul标签的子标签只能是li标签，li标签可以嵌套包括ol或ul标签之外的其它标签。

1. ol标签中li标签的两个属性：

start属性：指定起始编码，取值为数字。

type属性：指定有序列表的列表类型，取值为：

* + 1: 默认类型为 数字
  + a: 指定类型为 小写字母
  + A: 指定类型为 大写字母
  + i: 指定类型为 小写罗马字母
  + I: 指定类型为 大写罗马字母
  + none: 没有，不显示列表项

1. ul标签中li标签的type属性：指定列表类型，取值为：
   * disc: 实心圆(默认值)
   * circle: 空心圆
   * square: 实心矩形
   * none: 没有，不显示列表项