# HTML

## 注意事项

标签的属性不写属性值默认为“true”

* 1. 对于cdn引用的标签，添加crossorigin="anonymous"属性，在想url发送请求时将不携带当前域名的cookie
  2. 在script标签内出现字符串</script>时，会被视为结束符，需要在/前添加反斜杠\进行转义

## 综述

* 1. Web标准的构成：结构Structure（网页元素，HTML）、表现Presentation（外观样式，CSS）、行为Behavior（交互，JavaScript）
  2. URL：相对路径、绝对路径、网络URL路径
  3. src与href的区别：
     1. src表示对资源的引用，会将指向的资源下载并应用到文档中，被解析到时，会暂停页面的加载和处理直到该资源加载处理完毕
     2. href表示超文本引用，在当前元素和引用资源之间创建联系，浏览器识别到指向的文件时，会并行下载资源，不停止对当前文档的处理
  4. 文档模式：
     1. 标准模式（严格模式）：按照W3C标准解析代码，若文档中有DTD声明，则进入标准模式
     2. 混杂模式（兼容模式、怪异模式）：浏览器用自己的方式解析代码，通常模拟老浏览器的行为，若文档中无DTD声明，进入兼容模式
  5. 渐进增强与优雅降级：
     1. 渐进增强：先确保低版本浏览器能实现基本功能，再针对高级浏览器进行效果、交互等方面的追加
     2. 优雅降级：先构建完整的功能，再针对低版本的浏览器进行兼容

网站favicon图标：

* + 1. png转换为ico格式图标：<http://www.bitbug.net/> 、 在head标签中放入代码：<link rel=”shortcut icon” href=”图标路径” />

## 标签

* + 1. 标签是由尖括号包围的关键词，一般成对出现（双标签，开始标签和结束标签）
    2. 标签关系：包含关系（head与title）和并列关系（head和body）
    3. 标签骨架：
       1. <html></html>：HTML标签。根标签。
          1. <head></head>：文档的头部。head标签中必须要设置title。

<title>标题</title>：文档的标题。网页标题。

* + - * 1. <body>文本</body>：文档的主体。页面内容。
    1. 标签类型：
       1. 块级元素：div、ul、ol、li、dl、dt、dd、h1-6、p。。。
       2. 行内元素：a、span、img、input、select、strong
       3. 空元素（在开始标签中关闭，即没有闭合标签）：<br>、<hr>、<img>、<input>、<link>、<meta>

### 声明标签

* + - 1. <!DOCTYPE html> 文档类型声明标签，位于文档开头，向浏览器声明解析文档所用的规范（HTML/XHTML）
      2. <html lang=”en”> 当前文档的显示语言，英文模式可以写中文，给浏览器翻译作参考，中文为”zh-CN”
      3. <meta /> 位于head元素内部，提供有关页面的元信息，如SEO优化等
         1. charset=”UTF-8”【指定描述HTML文档的编码类型】、name=”description” content=”页面描述内容”【页面描述】、
         2. name=”keywords” content=”关键词”【页面关键词】、http-equiv="refresh" content=“5;url=http://...”【在5秒后重定向至指定页面】
         3. name="robots" content="xx"【搜索引擎索引方式】

content可用参数：多个参数用逗号分隔

all【文件将被检索，且页面上的链接可以被查询】、none【文件将被检索，且页面上的链接不可以被查询】、

index【文件将被检索】、follow【页面上的链接可以被查询】、noindex【文件不被检索】、nofollow【链接不被查询】

* + - * 1. name="viewport" content="xx"【移动端设置视口】

content可用参数：多个参数用逗号分隔

width=数值/device-width【设置视口宽度为指定数值或设备屏幕可见宽度，height同width】

initial-scale=1【设置初始缩放比例，maximum-scale/minimum-scale同左】

user-scalable=no/yes【是否允许用户缩放】

### 常用文本标签

* + - 1. <h1>文本</h1> 标题标签，由h1-h6逐渐变小，标题自动加粗
      2. <p>文本</p> 段落标签，文本会根据浏览器大小自动换行，段落之间有一定间距
      3. <br /> 换行标签，相当于\n
      4. <strong>文本</strong> | <b>文本</b> 加粗文本
      5. <em>文本</em> | <i>文本</i> 倾斜
      6. <del>文本</del> | <s>文本</s> 删除线
      7. <ins>文本</ins> | <u>文本</u> 下划线

### 布局标签

* + - 1. <div>内容</div> 区隔标记，浏览器通常在div元素前后放置一个换行符
      2. <span>内容</span> 行内标签，不会自动换行
      3. <iframe></iframe> 创建包含另一个文档的内联框架
         1. 优点：用来加载速度较慢的内容（广告）、实现脚本并行下载、实现跨子域通信
         2. 缺点：会阻塞主页面的onload事件、无法被一些搜索引擎识别、产生很多页面不容易管理

### 图像标签

* + - 1. <img src=”图像URL” alt=”文本”/> 图像标签，alt为替换文本，图像无法显示时显示的文字
         1. 可选属性：
         2. title=”文本” 提示文本，鼠标悬浮在图像上时显示的文字
         3. width=”像素” 设置图像的宽度 height=”像素” 高度
         4. srcset=“xxx”

值： 图像URL2 2x【在屏幕密度为2x时加载该图像】或 图像URL2 256w【在可视区域宽度小于该值时，使用该图片】

### 链接标签

* + 1. <link rel=“xxx” href=“url”> 规定文档与外部资源的关系
       1. rel=“可选值”【relationship】：stylesheet、shortcut icon【能被多数浏览器识别】、icon、preload
       2. sizes=“64x64”【定义相应icons的大小，数字单位为px，也可指定属性为any，表示可伸缩为任意尺寸】
       3. as=“可选值”【仅在rel特性设置为preload或prefetch时可用，规定加载内容的类型】
          1. audio、document【iframe元素】、image、style、script、video
    2. <a href=”跳转目标URL” target=”目标窗口的弹出方式”>网页元素</a> 超链接(anchor)标签，点击文本或图像可进行跳转
       1. target： \_self（默认）：当前页面打开 \_blank：新窗口打开
       2. 内部链接：直接链接内部页面名称即可，如”index.html” 跳转到当前页面：”#”
       3. 下载链接：URL若为文件或压缩包，则会下载文件 锚点链接：URL为”#id” 点击会跳转到页面的指定位置
       4. 跳转需要：在网页元素处设置id属性
    3. <script src=“xx”></script> 定义客户端脚本（如JavaScript）
       1. 特性：
          1. type=“可选值”【指定script元素包含或src引用的脚本的语言，值为MIME类型，若为js建议不写type】

text/javascript、text/ecmascript、application/ecmascript、application/javascript、module【视为js模块执行】

* + - * 1. src=”资源URL”【从外部引用文件】 charset=”UTF-8”【指定src引用文件的字符编码】
        2. async【HTML5，异步加载脚本，即脚本与HTML并行加载，脚本加载完后暂停HTML加载，执行脚本，然后继续加载HTML】
        3. defer【仅对外部脚本有效，延迟加载脚本，脚本与HTML并行加载，脚本延迟到HTML加载完成后执行】

defer会在执行完脚本后触发DOMContentLoaded事件，async与DOMContentLoaded事件无关

* + - * 1. integrity=“sha384-xxxxxx”【指定哈希值生成算法和实际哈希值，当获取资源时会进行校验，若哈希不一致，则不执行】
      1. <noscript></noscript> 可包含任何可出现在body中的元素，除了script，当浏览器不支持脚本时，显示该标签内的内容

### 注释标签和字符实体

* + - 1. <!--注释--> 注释，可跨行
      2. 字符实体：
         1. &nbsp; 空格 &lt; 小于号 &gt; 大于号

### 表格标签

* + - 1. <table><tr><td>单元格内的文字</td>...</tr>...</table> 基本表格单元
         1. tr：定义表格中的行 td（table date）：定义行中的单元格
      2. <table><tr><th>表头单元格内容</th></tr></table> 定义表头单元格(table head)的内容（加粗居中）
      3. table的属性：
         1. align：left、center、right 规定表格相对周围元素的对齐方式
         2. border：1或“” 规定表格是否有边框，默认为“”表示没有边框
         3. cellpadding：像素 规定单元边沿与其内容之间的空白，默认为1
         4. cellspacing：像素 规定单元格之间的空白，默认为2
         5. width / height：像素或百分比 规定表格的宽度
      4. 表格的结构标签：
         1. <table>

<thead><tr><th></th>...</tr>...</thead>

<tbody><tr><td></td>...</tr>...</tbody>

* + - * 1. </table>
      1. 合并单元格：
         1. 跨行合并：rowspan=“合并单元格个数” 写在最上单元格
         2. 跨列合并：colspan=“合并单元格个数” 写在最左单元格

### 列表标签

* + - 1. 无序列表：
         1. <ul>

<li>列表项1</li>...

* + - * 1. </ul> #ul内只能放li，li内可以放任何元素
      1. 有序列表：
         1. <ol>

<li>列表项1</li>...

* + - * 1. </ol> #ol内只能放li
      1. 自定义列表：小标题+列表项
         1. <dl>

<dt>名词1</dt>

<dd>名词1的解释</dd>

* + - * 1. </dl> #dt与dd无个数限制

### 表单标签

* + - 1. 表单域：提交信息至服务器
         1. <form action=”接受地址URL” method=”提交方式post/get” name=”名字”>各种表单控件</form>
      2. 表单控件：用户输入用
         1. input输入表单元素：

<input type=“值” />

type值：button（普通按钮，搭配js）、checkbox（复选框）、file（上传文件）、hidden（隐藏的输入字段）、image、password、

radio（单选框）、reset（重置，初始化表单）、submit（提交表单数据到服务器，初始化表单）、text

input属性：

name：单选框、复选框需要相同的name属性以实现功能

value：表单元素的值（checkbox的value为提交时发送到后台的值）

checked：checked=“checked”，规定打开页面时此元素默认被选中。js中用true和false表示

maxlength：规定最大输入长度

autocomplete="off" 关闭历史记录

* + - * 1. select下拉表单元素：

<select>

<option>元素1</option>

...

</select>

select中至少包含一对option，在option中定义selected=“selected”表示为默认选中项

* + - * 1. textarea文本域元素：多行文本输入

<textarea rows=“显示的行数” cols=“每行的字符数”>默认内容</textarea>

* + - 1. 提示信息：
         1. label标签：绑定表单元素，单击label的文本时，浏览器自动将焦点转到对应的表单元素上（若为checkbox则勾选）

<label for=”xx”>文本</label><input type=”” id=”xx” /> 或 <label><input type="checkbox">文件输入</label>

### 其他标签

* + - 1. 按钮元素：<button></button>相比input中的按钮更强大
      2. 分割线：水平分割线（horizontal rule）：<hr />

## HTML5（兼容IE9+）

### 新增标签

* + - 1. <!DOCTYPE html> 文档类型声明标签，位于文档开头，向浏览器声明解析文档所用的规范（HTML/XHTML）
      2. 新增语义标签（本质是块级容器）：
         1. <header></header>头部、<nav></nav>导航栏、<article></article>内容标签、<section></section>定义文档某个区域（大号div）、<aside></aside>页面内容之外的内容（侧边栏）、<details></details>元素细节、<summary></summary>details元素可见的标题、<figure></figure>与主内容相关，删除影响不大的独立的流内容【用<figcaption></figcaption>为figure添加标题】、<hgroup></hgroup>标题组、<dialog></dialog>对话框或窗口、<footer></footer>尾部<small></small>小号文本
         2. 在IE9中需要转换为块级元素、移动端无兼容问题
      3. 新增其他标签：
         1. <ruby></ruby> 定义ruby注释（音标）

<ruby>夼<rp>(</rp><rt>kuang</rt><rp>)</rp></ruby>

rt内表示注释，rp内表示兼容性不好时显示内部内容，兼容性好时不显示内部内容

* + - * 1. <meter></meter> 定义度量衡（类似磁盘空间指示条）

<meter value="5" min="0" low="2" high="7" max="10">5 out of 10</meter>

当value大于high值时会变色警告

* + - * 1. <progress max="100" value="0"></progress> 进度条
      1. 新增多媒体标签：
         1. <video src=”文件地址” poster=“图片地址” controls></video> 支持mp4、WebM、Ogg格式

video的属性：

autoplay=“autoplay” 视频就绪自动播放（chrome需添加muted才生效） controls=“controls” 向用户显示播放控件

width/height=”x px” 设置宽高 loop=“loop” 循环播放

preload=“auto/none” 预加载/不预加载视频（若设置了autoplay则忽略此属性）

src=“url” 视频地址 poster=“url” 加载等待的图片地址 muted=“muted” 静音播放

指定视频源：不兼容MP4，则播放ogg格式视频

<video width=”320” height=”240” controls>

<source src=”movie.mp4” type=”video/mp4”>

<source src=”movie.ogg” type=”video/ogg”>

</video>

* + - * 1. <audio src=“文件地址” controls></audio> 支持MP3、wav、ogg格式

属性（参考video）：autoplay（chrome自动禁用）、controls、loop、src

### 新增属性

* + - 1. crossorigin=“属性值”【可用于audio、img、link、script、video标签，配置相关请求的CORS设置，默认不使用CORS】
         1. anonymous【CORS请求不会通过cookies、SSL证书或HTTP认证携带用户凭据】、use-credentials【CORS请求携带凭据】
         2. 应用：跨域js脚本无法获取报错信息，当服务端添加Access-Control-Allow-Origin且本地设置crossorigin时，可获取

### 表单

* + - 1. 新增表单类型：type=”xx”
         1. email、url、date、time、month、week、number：限制用户输入必须为某个类型

number可以设置max、min、value【默认值】属性

* + - * 1. tel（手机号码）、search（搜索框）、color（颜色选择）
      1. 新增表单属性：
         1. required=“required” 指定内容不能为空 placeholder=“提示文本” 提示信息 autofocus=“autofocus” 自动聚焦
         2. autocomplete=“off/on” 若表单有name属性并成功提交，下次可快速填写（历史记录） multiple=“multiple” 可以多选文件提交
         3. 注：可以通过input：：placeholder｛color：pink；｝来设置提示信息的字体颜色

### 本地存储

* + - 1. 以键值对的方式存储，value为字符串类型，一般转换为JSON
      2. sessionStorage：会话存储
         1. 生命周期：浏览器打开到关闭的过程。大小：5MB。保存在浏览器端
         2. 方法：setItem（‘key’，value） / getItem（‘key’） / removeItem（‘key’）
      3. localStorage：永久存储
         1. 生命周期：永久，除非人为删除。大小：5MB甚至更大。保存在浏览器端
         2. 方法：setItem（‘key’，value） / getItem（‘key’） / removeItem（‘key’）
      4. cookie：用于浏览器和服务器端进行通信。
         1. 特点：最大4kb；每次发送请求都携带，占用带宽；保存在浏览器端；容易被截获；生命周期：会话cookie、人为设置cookie

### web worker

* + - 1. webworker是运行在后台的js，独立于其他脚本，不会影响页面的性能，通过postMessage将结果传回主线程，详见js笔记

### 其他

* + - 1. <img draggable="true" /> 允许拖放，抓取对象之后可以拖到另一个位置
      2. <canvas id="myCanvas" width="200" height="100"></canvas> canvas使用JS在网页上绘制图像
         1. 特点：依赖分辨率、不支持事件处理器、文本渲染能力若、能以jpg或png格式保存图像

# **XML**

* 1. 可扩展标记语言（Extentsible Markup Language，XML）是一种标记语言，是纯文本，是一种跨平台的，与软硬件无关的，处理与传输信息的工具。可被JSON替代。
  2. 与HTML的区别：
     1. 语法要求不同：
        1. 在HTML中不区分大小写。在XML中严格区分
        2. 在HTML中如果上下文清楚地显示出段落或者列表键在何处结尾，可以省略结束标记。XML是严格的树状结构，不能省略结束标记
        3. 在XML中，拥有单个标记而没有匹配的结束标记的元素必须用一个/ 字符作为结尾。
        4. 在HTML中，属性值的引号是可用可不用的。在XML中，属性值必须分装在引号中。
        5. 在HTML中，可以拥有不带值的属性名。在XML中，所有的属性都必须带有相应的值。
        6. 在HTML中，会自动过滤掉空格。在XML文档中，空白部分不会被解析器自动删除。
     2. 标记不同：
        1. HTML使用固有的标记；XML没有固有的标记。
        2. HTML标签是预定义的；XML标签是~~免费的、~~自定义的、可扩展的。
     3. 作用不同：
        1. HTML是用来显示数据的；XML是用来描述数据、存放数据的，所以可以作为持久化的介质
        2. HTML将数据和显示结合在一起，在页面中把这数据显示出来；XML则将数据和显示分开。
        3. XML被设计用来描述数据，其焦点是数据的内容。HTML被设计用来显示数据，其焦点是数据的外观。
  3. 示例：
     1. <student>
        1. <name>孙悟空</name>
     2. </student>
  4. SVG：可缩放矢量图形（Scalable Vector Graphic），是基于XML描述的2D图形的语言
     1. 在 SVG 中，每个被绘制的图形均被视为对象。如果 SVG 对象的属性发生变化，那么浏览器能够自动重现图形
     2. 特点：不依赖分辨率、支持事件处理器、性能不高

# XHTML

* 1. 可扩展超文本标记语言（XHTML，Extensible HyperText Markup Language），是将HTML作为XML的应用重新包装的结果
  2. 在XHTML中使用js必须指定type特性值为text/javascript
  3. 在XHTML中，script标签内的<会被视为标签的开始，解决方法：
     1. 用字符实体&lt；代替 < 或 在script标签内，使用如下形式：【为防止浏览器不兼容CDATA块，用js注释抵消】
        1. //<！[CDATA[
           1. js代码
        2. //]]>