

# html嵌套

2017年3月15日 14:44

html嵌套，类似与数学公式，都是树形结构，自相似，判断一个html嵌套或数学公式是否合法，可用如下方法。类似栈，只有一个地方进一个地方出。

自相似：整体与部分相似或完全相同，如表达式、代码，绝大多数有自相似特性的事物都是树状结构

例如：

```
<html>
<head>
<meta>
<title></title>
</head>
<body></body>
</html>
```

遍历标签，依次为

```
<html>
<head>
```

<meta>-----此时知道meta标签是自闭合标签，所以直接把该标签删除

继以上结果继续遍历：

```
<html>
<head>
<title>
```

</title>-----此时发现title标签得到对应，则删除该对标签，继续遍历，依次类推，到最后若为空则为合法html

# git

2017年3月16日 10:40

## 分布式版本管理工具

在你需要的目录下右键点击git bash here

git init 在一个文件夹的.git文件夹下初始化一个仓库

git status 查看当前仓库的状态

git add . 添加所有文件到暂存区

git add file.txt 添加file.txt到待提交区，即开始跟踪该文件；若文件修改，则需要重新add才算提交。

git commit -m "提交信息"：增加提交的注释

git log 查看提交日志

git diff 工作区与暂存区差异，查看当前文件内容与上一次的差别，必须是已经add的文件，即被跟踪的文件，修改之后才有返回

git diff --cached 暂存区与最近提交区对比

git rm file.txt 删除文件

git branch 显示分支

git branch dev --在当前分支上分出一个dev分支

git checkout master 切换到主分支上

git merge dev 把dev合并到当前分支

git log --graph --pretty=format:'%Cred%H%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit 显示树状图

git branch -d dev 删除某个分支

git rebase -master 将当前分支剪断放到master分支下，树状显示成直线，让历史记录更好看

注：rebase会修改当前分支的历史记录，所以不要把远程仓库的分支拉下来然后rebase然后再push

原每两次差异都会有一次patch，把patch都放到新分支下

git cherry-pick 遴选，只把某一次拿出来提交

git add -p 部分部分添加，patch

若是第一次提交，若提示 please tell me who you are

需要设置使用者的身份：

git config --global user.name 'dan.hu'

git config --global user.email '\*\*\*\*@qq.com'

配置都被写在本地一个叫.gitconfig的文件里

设置内部可以输入中文：

git config --global core.quotePath "false"

设置短命令,即命令别名

git config --global alias.ci commit

git ci

git init --bare 创建空的git仓库，用来push上远程仓库

1、git init 创建本地仓库(在工程目录下创建)

2、git init --bare 创建远端仓库（在服务器或者工程目录以外路径都可以创建的备份仓库）

工程commit到1中，push到2中；

git remote add name url 添加一个远程仓库，地址为url,名字为name

git push name master 把master分支push到origin这个远程仓库

git push 上推到关联网站

git push -force 强推

git pull 下载到本地，若线上代码为最新，需要先下载再修改代码，以保证版本一致，若pull下拉发生冲突，在本地把冲突代码修改完成再push即可。

git remote -v 查看所有关联仓库

git 短命令设置

```
$ git config --global alias.st status -----》 git st
```

```
$ git config --global alias.ci commit -----》 git ci
```

一份代码可以在多个仓库，一个仓库里不允许有多份代码！！

<自闭合标签>

<自闭合标签 />

空标签, void element

self closing tag

自己创建的标签是没有自闭合标签的。如<mytag> </mytag>

有些特定标签,即使是空标签,也不能自闭合

如script/iframe/canvas, 因为其内容用来fallback

如<script src=""><div>该浏览器不支持js</div></script>,当浏览器不支持js时, 会显示div, 否则不会显示

某些标签在特定的情况下可以省略结束标签

如<p>,即因为<p>标签不允许再嵌套<p>标签, 所以根据语意, 默认会帮你结束。

如<p>1<p>2相当于<p>1</p><p>2</p>,

<p>1 m p<address> 同<p>1 m p</p><address>,可在浏览器中测试

结束标签在以下情形中可以省略。<p>元素后紧跟

[<address>](#), [<article>](#), [<aside>](#), [<blockquote>](#), [<div>](#), [<dl>](#), [<fieldset>](#), [<footer>](#), [<form>](#), [<h1>](#), [<h2>](#), [<h3>](#), [<h4>](#), [<h5>](#), [<h6>](#), [<header>](#), [<hr>](#), [<menu>](#), [<nav>](#), [<ol>](#), [<pre>](#), [<section>](#), [<table>](#), [<ul>](#)或另一个<p>元素; 或者父元素中没有其他内容了, 而且父元素不是<a>元素

来自 <https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/p>

在什么情况下html标签可以不需要闭合?

在浏览器可以通过html标准推导出某些标签已经结束的时候, 可以不需要其闭合标签。

典型的有

<p>1<p>1,

<ul>

<li>123

<li>123

</ul>

元素分类:

块元素: <h1> <h2> ..<h6>、<p>、<blockquote>、<ol>、<ul>、<li>

内联元素: <a> (若<a>中包含了块状元素时, 即为块状元素) <q> <em>

块元素都单独显示, 前后都有换行, 特立独行

内联元素会显示在所在段落中, 随波逐流

Html5之后不再仅仅只有块元素与内联元素, 而是Content categories, 具体看

[https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content\\_categories#Flow\\_content](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content_categories#Flow_content)

属性不一定要有值

如:<input type="text" required />

(当属性的值里没有空格、引号等特殊字符时, 属性值可以不加引号)

全局属性 id name title style class tabindex contenteditable

title属性, 鼠标在上面时显示的tooltip文本

alt属性, 主要用在img标签上, 图片加载失败时才显示 (alternative 替补)

class属性是一个以空格分隔的列表, 如class="a b c d"

style属性, 用于给标签指定内联样式

tabindex属性, 可以用在div元素上以实现div也能获取到焦点事件

contenteditable属性, 可以设置div元素或其他元素, 直接可编辑, <div contenteditable> </div>

data-\*\*自定义属性，保存个性化的数据

具体全局属性可参考 [https://developer.mozilla.org/en-US/docs/Web/HTML/Global\\_attributes](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes)

要在内容中输入标签或特殊符号，需要使用符号的字符实体来表示，否则html结构会被打乱

属性转义：

&name

&#number; ; number是十进制的unicode

&#xHHHH; HHHH是十六进制，x表示十六进制(hexadecimal); binary二进制，decimal十进制，octo八进制

常见html字符实体 html entity：

&nbsp;或&#160; 空格 non-breaking space 非合并空格

&amp; 符号

&lt; 小于号

&gt; 大于号

&copy; 版权符号

&quot; 双引号

&apos; 单引号

&equal; 等号

已经弃用

escape 转义

unescape 反转义

用这两个代替

[encodeURIComponent](#)

[encodeURIComponent](#)

如 <html> 为 &lt;html&gt;; , & 为 &amp; (输入实体时都会用到&，而如果内容中需要一个&，要使用&amp;)

具体可参考[http://www.w3school.com.cn/html/html\\_entities.asp](http://www.w3school.com.cn/html/html_entities.asp)

空格：

默认情况下浏览器忽略文字与文字之间多于一个的空格，换行符会被全部忽略。

如何不用css让空格不被合并？

html实体:&nbsp;

但注意这个不是真的空格（即ascii 32），不是我们手动输入的空格，实际是ascii编码是160

html中用空格键输入的空格是&#32;，而不是&#160;，因此如果数据中有大量的空格，但是显示却只显示了一个，原因可能就是数据中的空格是

&#32;，即是手动输入的空格，会自动合并；要正确显示则改成&#160;即可。

<pre>用来保持原样式

"".charCodeAt(0),查看空格的ascii

2\*\*2 在js中可直接运行，即2^2=4

js

'\u6211' 我

css

\u

\A 回车

# html标签

2017年3月16日 14:44

**html**根标签

**head**:头部标签,放一些与页面相关的“元信息”。比如下面的meta。

**meta** : <meta charset="utf8">, 编码方式

**title** : <title></title> 页面标题, 只支持存文本, 不支持任何标签或样式

**link** : <link rel="icon" href="a.icon">图标, 也可以不用, 浏览器会自动读取网站根目录下的a.icon这个文件, 例如http://www.baidu.com/a.icon, 谷歌的图标: <link href="/images/branding/product/ico/googleg\_lodp.ico" rel="shortcut icon">, shortcut即建快捷方式时的图标  
<style></link rel="stylesheet" href="a.css"> : 样式  
<link rel="icon" href="http://example.com/a.icon">最好写绝对路径, 以防止标签在其他路径页面无法打开, 或者直接不写让浏览器自己去访问根目录。

**body** : 所有要显示的内容放在body中

注:

如果没有head标签或body标签, 浏览器都会自动增加。

如果在body或者html标签的结束标签之后又出现了其他标签, 则浏览器会默认为其结束标签无效, 会自动添加到最后。(F12查看元素)

**h1-h6**:

比较传统的观点认为一个页面不能出现超过一个h1标签, 原因是为了SEO。

**SEO**: Search Engine Optimization 搜索引擎优化, 即如何布置页面才能让页面在搜索结果中尽量靠前

影响因素:

关键字 <meta name="keywords" content="购物">

页面被引用的次数(反链接数量)

使用https

使页面的html更符合语义

提升网站网站的速度

**hgroup** (即header group), html5新增标签, 当不同级别的标题显示在一起的时候可以用这个标签把标签归类。该标签可以替换以前使用div的场景。

如:

```
<hgroup>
<h2>a</h2>
<h3>b</h3>
</hgroup>
```

**a** : anchor, 锚。

<a href="javascript:a='88';void 0;">123</a> 不加void 0 浏览器的默认行为相当于document.write(88), 所以不要以赋值字符串结尾

链接地址写在href中 (hyperlink reference) 属性里, 且如果为网站地址需要为绝对地址, 如<a href="jd.com"></a>无效, .com可以是一个文件后缀, 需要写成<a href="https://jd.com"></a>

页内特定位置跳转地址 : <a href="#top"></a>回到顶部</a>, url中会增加#top;

页面之间特定位置链接 : <a href="index.html#chai">see chai tee</a>, 多用于章节跳转;

相对路径:

c主目录连接到a子目录: 'a/a.html' 注: 对于网页, 只用/, 不用\

a子目录连接上一层到c目录: '../c.html' '..'表示父文件夹, '.'表示当前文件夹

a子目录链接到上两层d目录: '../../d.html'

"/index.html" 以斜杠开头, 表示从根目录取

邮件地址: (需要在电脑上安装邮件客户端)

<a href="mailto:aaa@bbb.com"></a>

<a href="mailto:aaa@bbb.com?title=1&subject=2&content=3"></a>

QQ/Taobao临时会话:

tencent://temp-CHAT?qq=123455

空的href属性: href="" 链接到当前页面, 页面会刷新, 类似的, 一个img标签的src=""也指向当前页面的相对路径, 页面会加载两次, 具体可查看文章

<https://www.nczonline.net/blog/2009/11/30/empty-image-src-can-destroy-your-site/>

**target**:

\_blank 链接在空白的窗口显示, 新打开一个空白页

\_parent 链接在父窗体显示 (iframe时需要)

\_top 链接在顶层窗体显示 (iframe时需要)

\_self 默认

自定义: 不能以\_开始, 如下用法:

<a target='a' href="test.html">test0</a>

<a target='a' href="test.html">test1</a>

<a target='b' href="test.html">test2</a>

给target命名为相同名字时, 该标签不会重复打开, 会覆盖之前打开的为该名字的页面

**download**: 点击这个链接将下载链接对应的文件, 而不是跳转到目标页面, 下载的文件名以download属性的值来命名, 且只能下自己域下的文件

<a href="xxx/a.txt" download="abc.txt">点击下载</a>

**img**:

图片格式:

webp(手机端, 同JPG, 效果比JPG好一些)、bmp(大小跟尺寸成正比, 不会被压缩)、png (无损压缩, 适用于屏幕截图, 纯色区域, 压缩率很低, 文件较大)、jpg/jpeg (压缩率很高)、gif、svg (矢量图)

注: psd格式不支持, 它是photoshop专有的格式

JPG/JPEG: 照片和复杂图像, 有损压缩, 因为缩小文件小时会丢掉图像的一些信息, 与jpg压缩算法类似的格式是mpg格式的视频, 不支持透明色

webp: webp在各方面都胜出jpg, 有损压缩, 与webp压缩算法类似webm

PNG: 单色图像、logo和几何图形, 适用于屏幕截图, 纯色区域, 无损, 允许将颜色设置为“透明”

GIF: 可以表示最多256种不同颜色(非固定, 根据照片而定)的图像, 无损压缩, 只允许一种颜色“透明”, 支持动画

bmp: 不会被压缩, 大小跟尺寸成正比

svg : Scale Vector Graphic 可缩放的矢量图像

一个像素是3个字节

alt:图片加载失败之后替换成alt中的文字, alternative, 同键盘上的Alt

width height : 最好按照等比例设置, 或者只设置其中一个, 浏览器会自动按照比例得出另外一个。

但是图片加载往往需要时间, 而图片在加载完成之前浏览器是不知道其宽高的, 所以就会产生页面抖动的问题

所以一般会在标签上把宽高写出来, 这样图片加载过程中页面就不会抖动了

注: 尽量不要用height和width来设置图片大小, 因为浏览器在缩放图像使之适应页面大小之前, 仍然需要获取整个大页面。

base:

<base href="页面中所有相对路径相对对象" target="页面中所有链接的打开位置">

href属性表示页面中所有相对路径的基础路径

一定要以/ 即目录结尾, 如 <base href="https://www.baidu.com/abc/">

<base href="https://www.baidu.com/abc">相当于 <base href="https://www.baidu.com/">

target属性表示页面中所有链接的打开位置, 当然也可以在页内的a标签中用target属性覆盖用base标签设置的全局打开位置

span : 没有明确语义

br : <br>换行, 自闭合标签

hr : 水平分隔线, 自闭合标签, <hr>,horizontal

font/blink/marquee: 不推荐使用 ( deprecated ), 已废弃 ( obsolete ); 字体/闪字/弹幕

em : emphasis,斜体强调

strong : 加粗强调

b : bold,强调, 只是样式上加粗

i : italic 斜体, 用来表示由于某些原因需要与普通文本区分的文档; 也有用这个标签来表示图标, 如http://fontawesome.io, <i class="fa fa-star"> </i>

<!--123 --> : 注释, ctrl+/ ; python和命令行都是#注释

pre:: pre format ted,预定义格式的文本, 里面的内容的回车跟空格都会被保留,常与code标签连用。如

```
<pre>
<code>code goes here</code>
</pre>
```

ul/ol : 无序/有序列表, 都是块状元素, 可以多层嵌套, li为其直接子元素, li内可以是任意其他标签; md就是编译(compile)成了html的该效果

dl : 描述性标签, 子元素为dt、dd, dt下可以有多个dd, description list, 如

```
<dl>
  <dt>老师</dt>
  <dd>a</dd>
  <dt>学生</dt>
  <dd>b</dd>
  <dd>c</dd>
</dl>
```

表单标签 :

<form action="https://www.baidu.com/s" method="get" target="\_blank">

method不填是默认get

提交文件需要增加配置 enctype="multipart/form-data" 到form

CKEditor 富文本编辑器

codeMirro 代码编辑器

IDE

input :

自闭合标签

type值常见的有: text,password,

number ( 只能输入数字, 但是包括e, e4表示 $10^4$  ),

checkbox(多选), 需要用name属性给其分组, 默认选中的值value是on,value值也可以自己设置

radio(单选), 需要用name属性给其分组, 以保证只能选一个,默认选中的值value是on,value值也可以自己设置

file(上传文件)

type值用法详解:

若输入的值不正确, html默认为text

file : 上传文件 <input type="file" name="file\_upload" >, 没有value值, 选择之后才有值, 即文件名称。

accept属性 : 扩展名限制, 只是减少用户过滤文件的范围, 提高用户体验, 其他类型的文件照样可以上传。

如: <input type="file" name="file\_upload" accept=".jpg, .gif," > 或 <input type="file" name="file\_upload" accept="image/\*" >

扩展知识: MIME Type, 文件类型, 媒体类型, 具体格式:

text/plain 文本

text/html

image/jpeg

image/\* 所有图片

audio/mpeg

audio/\* 所有音频

application/octet-stream 不确定的, 字节流

multiple属性 : 同时选择多个上传文件, 该属性没有值, 如<input type="file" name="file\_upload" multiple>

hidden:用来设置默认字段的值

image: 用图片来模拟按钮, <input type="image" src="a.jpg" alt="图片显示失败">,现在基本不这么用

button : 无缺省行为按钮

submit : 用于表单提交, <input type="submit" value="登录">, 一般直接用<button>比较多, 因为<button>里面文字可以嵌套div, span等。

reset: 用于将表单内容设置为缺省内容

email:邮箱地址, 有验证邮箱地址格式, multiple属性, 没有值, 可以写多个地址, 以逗号分隔

datetime:日期



datetime-local:日期时间

time:时间

range:范围, 0-100, 可修改

color:颜色

tel: 电话, 一般用于移动端会自动出来数字键盘

**name属性**: 提交表单时, 会把name值与输入的value拼接成提交后的url, 如https://www.baidu.com/s?f=8&rsv\_bp=1&rsv\_idx=1&word=a

query string 查询字符串, 例实现谷歌查询:

```
<form action="https://www.google.com/search" target="_blank">
  <input type="text" name="q">
  <button>谷歌查询</button>
</form>
```

**value属性**: 形态为button则设置按钮值

形态为 输入框是设置里面默认的内容

disable属性: 禁用, 不能触发focus、changed事件, 就算js代码中修改该属性, 事件还是无法触发

```
[disabled].disabled{ //因为老浏览器没有伪类, 所以要加上[disabled]属性;
  color:gray;
  cursor:not-allowed
}
```

required属性: 必填, 否则表单不提交

hidden属性: 隐藏

maxlength:最大长度, 表单内容为文本时可以设置文本长度

minlength:最短长度, 一些浏览器不支持

placeholder:占位符, 一旦输入, 里面文字就消失, <input type="text" placeholder="请输入姓名">

autofocus:自动获得焦点, 一些浏览器不支持, 最好用js实现

tabindex:为-1的话会跳过那个元素, 默认为0, 可以改变tab键光标顺序, 可以用在div元素上以实现div也能获取到焦点事件, 切换时先按tabindex,再按dom中顺序

pattern:检查控件值的正则表达式。当类型属性的值为text、search、tel、url或email时, 此属性适用

**button**:

type属性: button, submit, reset, 若设置了type为这些值, 功能与input一样。type属性默认为submit  
与input的区别, input的button只能在按钮上设置纯文字, 而button按钮可以设置图片、文字颜色等。

**label**:

用于扩大可点击区域, 如:

```
<label for="like" >apple</label>
<input type="checkbox" id="like"> 或
<label> <input type="checkbox" id="like">apple</label>, (坑: onclick事件若绑定在label事件上会触发两次, 所以一般绑定在input上)
点击apple时, 输入框被选中, 如<label> apple<input type="text" id="like"></label>
```

**select**: 下拉选项

multiple多选, 按住ctrl键, 显示时为大框而非下拉

size设置宽度

子元素为optgroup或option, disabled属性, 如果设置该属性, 其内的所有选择域都将被禁用

```
<select name="city"> 9bh
  <option hidden>请选择</option> --hidden设置下拉选择的时候不出现
  <option value=1>1</option>
  <option value=2>2</option>
  <optgroup label="浙江"> --optgroup 设置树状结构的下拉选项, 本项不可选
    <option>hangzhou</option>
    <option title="dfjiowefjewf">ningbo</option> --title 当内容过长时, 可以把内容放在title中以示提示
  </optgroup>
</select>
```

option标签主体内容太长影响用户体验, 解决方法: 用title显示所有内容, 在option中显示部分内容

**textarea**: 若要输入html标签, 需要html实体转义。rows=3 cols=20

**fieldset**: 字段/域, 给输入域分组

disabled属性, 如果设置该属性, 其内的所有输入域都将被禁用, 类似optgroup

```
<fieldset disabled>
  <legend>基础信息</legend> --作为fieldset的子元素
  <input type="text">
  <input type="text">
</fieldset>
```

## html标签二

2017年3月20日 9:56

### Table:

**table**: 表格的布局在页面上不会乱, 但用css布局的页面可能会导致效果混乱。

**caption**: 表格标题

**thead**: 表头, 只能有一个, 表头放在thead里面可以保证打印的时候, 如果分页了可以每页都有标题

**tbody**: 表格主体, 可以有多个

**tfoot**: 表尾

**th**: 用在表头单元格, 文字默认加粗

**tr**: 不支持position:relative, 也不支持box-shadow, 包括td,th等

**td**: 可以嵌套任何标签, 包括表格

headers属性, 值为某th的id, 可以为多项, 表示该列的具体意义, 适用于读屏软件

如

```
<table border="1">
  <thead>
    <tr>
      <th id="name">姓名</th>
      <th id="age">年龄</th>
      <th id="sex">性别</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th headers="name" id="name_1">1</th>
      <td headers="name_1 age">a</td>
      <td >nv</td>
    </tr>
  </tbody>
</table>
```

colspan: 跨列

rowspan: 跨行

**col**: table的子标签, 用来批量处理列的样式, 如<col bgcolor="red" width="100">

span属性, 用来表示跨了多列, 如:

<col span="2" bgcolor="red">等同于<colgroup bgcolor="red"><col><col></colgroup>

必须出现在caption后面, thead/tbody前面, 否则无效。

**colgroup**: table的子标签, 用来批量处理多列样式, 如<colgroup bgcolor="red"><col><col></colgroup>  
行的样式优先于列的样式

**map**: 映射标签, name属性, 如果设置id属性, id跟name属性值必须一样

area: 为map的子标签, 定义map里的区域, 属性: href、target、alt同a标签

shape: rect, 矩形, x1,y1,x2,y2, 左上角的坐标, 右下角坐标

circle, 圆形, x,y,r, 圆心坐标, 半径

poly, 多边形, x,y 每个角的坐标, 至少6个值 (六个值为三角形)

coords: 坐标位置, coordinate

如

```

<map name="ab">
  <area shape="circle" coords="0,0,10" href="" >
</map>
```

**iframe**: 嵌套其他页面, 因为安全问题, 某些网站不允许被放到iframe标签里, 如谷歌

src属性, 即用于指定嵌套的网站地址

name属性，当a标签的target名称跟iframe的name值一样，则a标签的页面会在iframe中打开

a标签的target有\_top以及\_parent值，即意味着当iframe多层嵌套时，会根据a的target值在相应的窗口打开。\_top即顶层窗口，\_parent即当前a标签所在窗口的父窗口，多用于导航。

跳转记录保存在浏览器前进后退的记录里。

**frameset**：把页面分成多个区块，每个区块显示一个页面，现很多浏览器已经不支持，具体用法如下

```
<frameset cols="200,*" b>          --分成2列，第一列宽度为200，剩下的为第二列
  <frame src="https://www.baidu.com/" >
    <frameset rows="100,*">        --又嵌套一层frameset,分成2行，第一行宽度为100，剩下的为第二行
      <frame src="http://baidu.com">
      <frame src="http://baidu.com">
    </frameset>
  </frameset>
</frameset>
<noframes>您的浏览器不支持iframe</noframes>
```

扩展：浏览器不支持时，才会显示标签中的内容

`<iframe>您的浏览器不支持iframe</iframe>`

`<noframes>您的浏览器不支持iframe</noframes>`

`<script>alert(1)</script>`

`<noscript>您的浏览器不支持script</noscript>`

`<canvas>您的浏览器不支持canvas</canvas>`

为什么不同类型的标签的fallback，内容要以不同的形式提供？

答：因为不同的标签使用方式不同，有些是内部需要放放置的内容，有些则是内容被支持的浏览器忽略。

如果是前者，则fallback需要写在其外部，如script,frameset

如果是后者，一般是把fallback内容写在标签内部，如canvas,iframe

**html5新增标签：**

article

section：同div

aside

header

footer

nav(igate):导航栏，网站顶部的导航栏

main

template，专门用来存放模版，不会显示在页面上，后期通过js绑定数据，同如下：

```
<script type="text/template" id="temp">  ---script标签里的内容不会显示在页面上，该type不是js所以不会被当作代码执行
<div><a>123</a></div>
</script>
```

time: <time datetime="2016-1-1">昨天</time> 时间语义化标签

**其他常见标签：**

sub:下标

sup:上标

code：一般与pre一起，用来显示代码块,<pre><code>code goes here</code></pre>

script：内联：<script type="text/javascript"></script>

外联：<script src="a.js"></script>

内部不能出现代码，最好拼接 alert("<scri"+"pt")，或者用html实体，否则代码会混乱

style：内联：<style></style>

外联：<link rel="stylesheet" href="style.css">

html5中type属性可省略，即<link rel="stylesheet" href="lounge.css">

rel属性指定HTML文件与所链接文件之间的关系，stylesheet即样式表

**多媒体标签：**

video：视频，支持格式：mp4(h264/h265)、m3u8、webm( )

```
<video src="a.mp4" preload controls>
  <source src="a.mp4">
  <source src="a.m3u8"> --mp4不支持的情况下会获取该文件
  您的浏览器不支持video
</video>
```

autoplay属性，自动播放

loop 属性，循环

preload 属性，预加载

controls属性，控制条

扩展：flv.js，把不支持的格式转换为支持的格式，自己查看

audio：音频，支持格式：mp3 ogg m3u8 wmv wav，同video

object: 东西，对象，引入一个外部资源

type属性：MIME Type

可以在页面中嵌入iframe、pdf、svg、img、flash

embed：同object

canvas:画布

progress：进度条，min属性，max属性，value属性

其他标签：

div：无语义，块级标签

span：无语义，内联标签

使用时一定要带class使用

```
<div class="a">123</div>
```

```
<span class="a">123</span>
```

meta：元数据标签

link：<link href="a.css" rel="stylesheet">

doctype：非html标签，

如果不写doctype声明，页面在IE浏览器下会以quick模式（怪异模式）渲染

最主要的区别是css盒模型计算方面

语义化：语义化好的页面能够方便人与机器的理解

能够让团队后来者易于维护代码（可维护性）

能够让页面在搜索引擎的结果中更靠前（让机器更理解）

合适的内容用合适的标签

合理使用h,p,列表,表格,div,span及html5新增的语义标签

使用合适的嵌套

给元素命名上合适的类名及id名称

role与aria-\*:accessible rich Internet Application

如当用div来模拟某个UI控件时，就失去了语义化，视力障碍者就无法通过读屏软件知道当前控件的意义，这时就可以用role来标记该div的语义，用aria-\*来表示他的状态。

role：表示当前标签的语义

role=textbox

role=listbox -->select

role=option -->option

role=progressbar -->progress

role=button

aria-\*:表示当前标签的状态

aria-disabled=true

aria-checked=true

aria-valuenow=true

aria-valuemax=true

aria-valuemin=true

aria-readonly=true

aria-selected=true

一般来说，role与aria-\*用在非语义标签上，如div,span,li,a等，不用在nav,article,footer,select,input等有明确语义的标签上

具体文章：<https://developer.mozilla.org/en-US/docs/Web/accessibility/aria>

总结：

可以使用target属性的标签：form、a、area、base

可以使用href属性的标签：a、link、area、base

可访问性：accessibility,为的就是让不能以正常途径访问页面的人也能够使用这个页面提供的服务。

如td[headers]，图片的alt属性，fieldset/legend，<div role="" aria-\*=""></div>