

HTML

(超文本标记语言——HyperText Markup Language)

块级元素

相对于其前面的内容它会出现在新的一行，其后的内容也会被挤到下一行展现

块级元素不会被嵌套进内联元素中，但可以嵌套在其它块级元素中

内联元素

内联元素不会导致文本换行

原义字符	等价字符引用
<	<
>	>
"	"
'	'
&	&

规定：

1.无特殊声明：双标签

2.[] 例：<meta >

文本知识

html 标签： 为站点设置语言(单个)

```
<html lang="zh-CN">
```

head 标签

1 title： 标签页标题(书签名)★

2 meta： 元数据(单个)★

(title、meta可搜索引擎匹配)

```
指定你的文档中字符的编码：<meta charset="utf-8">
添加信息name和描述content：<meta name="author" content="zwx">
```

3 **link**: 定义当前文档与外部资源的关系(单个)★

```
链接CSS
<link rel="stylesheet" #rel规定当前文档和被链接文档的关系
      href="my-css-file.css">
```

4 **script**: 链接js文件，最好放在 </body> 最后★

```
<script src="js-file.js"></script>
```

body 标签

1 **h1~h6**: 网页内容标题★

2 **p**: 定义段落★

3 **ul**: 无序列表(Unordered List)★

4 **ol**: 有序列表(Ordered List)★

5 **li**: 把列表元素单独列出来★

```
<ol>
  <li></li>
  <li>
    <ul>
      <li></li>
      <li></li>
    </ul>
  </li>
</ol>
```

6 **dl**: 描述列表(description list)

dt: 所描述的词(description term)

dd: 描述(description description)

```
<dl>
  <dt>内心独白</dt>
  <dd>戏剧中，某个角色对自己的内心活动或感受进行念白表演，这些台词只面向观众，而其他角色不会听到。</dd>
</dl>
```

7 **em**: 斜★

8 **strong**: 加粗(强调)★

9 **a**: 超链接★

1. 链接网页

```
<a href="url" #链接内容
    title="内容" #鼠标放在超链接上显示的内容
    target="_blank(新标签页)/_self(默认值, 不弹新窗口)"
>主页</a>
```

2. 本地内容链接(转跳)

```
<h2 id="Mailing_address">邮寄地址</h2>
<p> 请将信件邮寄至 <a href="contacts.html#Mailing_address">我们的地址</a></p>
```

3. 电子邮箱链接

```
<a href="mailto:2574802744@qq.com">向我发邮件</a>
```

相对URL, 绝对URL

尽可能使用相对路径

10 **blockquote**: 块引用(也可以[/])

blockquote 元素前后添加了换行, 并增加了外边距。

```
<blockquote cite="url"></blockquote>
没有浏览器支持cite
```

11 **q**: 用于不需要分段的短引用

```
<q cite="url"></q>
```

12 **abbr**: 提供缩写的解释

```
<abbr title="超文本标记语言 (Hyper text Markup Language)">HTML</abbr>
```

13 **address**: 标记联系方式

```
<address>
    郑文昕.....
</address>
```

14 **sup**: 方程上标 **X²**

15 **sub**: 方程下标 **H₂O**

16 展示计算机代码

- : 用于标记计算机通用代码。
- code: 用于标记计算机通用代码。
- pre: 用于保留空白字符 (通常用于代码块) —— 如果您在文本中使用缩进或多余的空白, 浏览器将忽略它, 您将不会在呈现的页面上看到它。但是, 如果您将文本包含在 `<pre></pre>` 标签中, 那么空白将会以与您在文本编辑器中看到的相同的方式渲染出来。
- var: 用于标记具体变量名。
- kbd: 用于标记输入电脑的键盘 (或其他类型) 输入。
- samp: 用于标记计算机程序的输出。

17 **time**: 时间标签(为机器识别)

18 文档的基本组成部分★★

header: 页眉

nav: 导航栏

main: 主内容。主内容中还可以有各种子内容区段，可用article、section、div等元素表示。

aside: 侧边栏，经常嵌套在main中。

footer: 页脚



细节: ★

- main 只能用一次，直接放在body里
- article 包含一篇文章
- aside: 包含间接信息
- nav: 主导航，不应包含二级链接
- footer: 页脚

19 **span**: 内联的无语义元素

20 **div**: 块级无语义元素

应仅用于找不到更好的块级元素时，或者不想增加特定的意义时

21 **br**: 换行(单个)

22 **hr**: 水平分割线(单个)

多媒体

一 图片

1 **img**: 图片(单个)

1.1 **src**="url"

装饰性图片CSS的background-images

1.2 **alt**=""(无法找到该图片从而显示备选文本)

1.3 **width**=""(宽度)

1.4 **height**=""(高度)

(改变尺寸用CSS好)

1.5 **title**="内容"(悬停在图片，出现标题)

```

```

2 **figure>img+figcaption**(为图片提供一个语义容器)

figure: 语义容器

figcaption: 文字描述元素的内容(最好和alt内容不一样)

```
<figure>
  <img src="">
  <figcaption>猪</figcaption>
</figure>
```

3 **CSS 背景图片: background-image: url("");**

注: 图像对您的内容里有意义, 则使用HTML图像。 如果图像纯粹是装饰, 则使用CSS背景图片。

二 视频和音频

1 **video**: 视频

1.1 **src**="url" (链接)

1.2 **controls** (控制视频音频回放功能)(单元素)

写法1: `<video src="video.mp4" controls></video>`

写法2(推荐)

```
<video controls>
  <source src="1.mp4" type="video/mp4">
  <source src="1.webm" type="video/webm">
  <p>你的浏览器不支持 HTML5 视频。可点击<a href="1.mp4">此链接</a>观看</p>
</video>
```

1.3 **width**和**height**: 横纵比

1.4 autoplay: 自动播放(单元素)

1.5 loop: 循环播放(单元素)

1.6 muted: 默认关闭声音(单元素)

1.7 poster="url" : 图像(在视频播放前显示)

1.8 preload: 缓冲:

= "none": 不缓冲;

= "auto": 页面加载后缓存媒体文件

= "metadata": 仅缓冲文件的元数据

```
<video controls
  width="400"
  height="400"
  autoplay
  loop
  muted
  preload="auto"
>
<source src="1.mp4" type="video/mp4">
</video>
```

2 audio: 音频

(属性与video相同, 无视觉部件)

2.1 src="url" (链接)

2.2 controls (控制视频音频回放功能)(单元素)

写法1: <audio src="music.mp3" controls></audio>

写法2(推荐)

```
<audio controls>
  <source src="music.mp3" type="video/mp3">
  <source src="music.ogg" type="video/ogg">
  <p>你的浏览器不支持 HTML5 音频。可点击<a href="music.mp3">此链接</a>观看</p>
</audio>
```

2.3 autoplay: 自动播放(单元素)

2.4 loop: 循环播放(单元素)

2.5 muted: 默认关闭声音(单元素)

2.6 preload: 缓冲:

= "none": 不缓冲;

= "auto": 页面加载后缓存媒体文件

= "metadata": 仅缓冲文件的元数据

~~width和height：横纵比（无此元素）~~

~~poster="url"：图像(在视频播放前显示)（无此元素）~~

2.7 重新播放媒体-js

```
const mediaElem = document.getElementById("my-media-element");
mediaElem.load();
```

2.8 音轨增删事件-js

```
const mediaElem = document.querySelector("video");
mediaElem.audioTracks.onaddtrack = function(event) {
  audioTrackAdded(event.track);
}
```

2.9 显示视频音频文本字幕

三 iframe：嵌入

```
<iframe src=""></iframe>
```

3.1 allowfullscreen：全屏API(单元素)

~~3.2 frameborder：-1(有边框)/-0(无边框)~~

3.3 src="url"（链接）

3.4 width和height：宽度和高度

3.5 sandbox：安全设置(单元素)

四 将SVG添加到页面

4.1 img：

优点

- 快速，熟悉的图像语法与 `alt` 属性中提供的内置文本等效。
- 可以通过在 `<a>` 元素嵌套 ``，使图像轻松地成为超链接。

缺点

- 无法使用JavaScript操作图像。
- 如果要使用CSS控制SVG内容，则必须在SVG代码中包含内联CSS样式。（从SVG文件调用的外部样式表不起作用）
- 不能用CSS伪类来重设图像样式（如 `:focus`）。

4.1.1 srcset：新浏览器将加载SVG - 较旧的浏览器将加载PNG

```

```

4.2 CSS插入

像上面描述的 `` 方法一样，使用 CSS 背景图像插入SVG 意味着它不能被 JavaScript 操作，并且也受到相同的 CSS 限制。

4.3 如何在HTML中引入SVG代码：

```
<svg width="300" height="200">
  <rect width="100%" height="100%" fill="green" />
</svg>
```

4.4 iframe 嵌入SVG

五 怎样创建自适应的图片？

表格

1 **table**：创建表

2 **th**：表头(table header)

3 **td**：单元格内容(table data)

3.1 **colspan**：表行占的单元格个数

3.2 **rowspan**：表列占的单元格个数

4 **tr**：换行(table row表行)

```
<table>
  <caption>标题</caption>
  <tr>
    <th rowspan="4">表头</th>
  </tr>

  <tr>
    <td colspan="2">内容1</td>
    <td>内容2</td>
  </tr>

</table>
```

5 **caption**：表的标题

6 **thead**：表头

7 **tfoot**：页脚

8 **tbody**：正文