# Html学习笔记

小甲鱼导航网页：<https://man.ilovefishc.com/html5/>

HTML实体：https://man.ilovefishc.com/page/ISO.html

HTML的注释语法

<!-- 在此处写注释 -->

Css注释语法

<!-- 在此处写注释 -->

JavaScript的注释语法

与c系列一样

# **HTML 块级元素（block element）：**

块级元素总是在新的行上开始，并尽可能地占据本行全部可用的宽度。

<address> <article> <aside> <blockquote> <canvas> <dd> <div> <dl> <dt> <fieldset> <figcaption> <figure> <footer> <form> <h1>~<h6> <header> <hr> <li> <main> <nav> <noscript> <ol> <output> <p> <pre> <section> <table> <tfoot> <ul> <video>

**HTML 行内元素（inline element）**

相反，行内元素不会另起一行，它也只占用必要的宽度

一言以蔽之：块级元素讲究排场，行内元素勤俭节约！（当然，也不是说谁好谁坏，元素无贵贱，存在即合理）

<a> <abbr> <acronym> <b> <bdo> <big> <br> <button> <cite> <code> <dfn> <em> <i> <img> <input> <kbd> <label> <map> <object> <q> <samp> <script> <select> <small> <span> <strong> <sub> <sup> <textarea> <time> <tt> <var>

**行内元素与块级元素对比**

一般情况下，行内元素只能包含数据和其他行内元素。  
  
而块级元素可以包含行内元素和其他块级元素。  
  
这种结构上的包含继承区别可以使块级元素创建比行内元素更”大型“的结构。

HTML 要求浏览器将连续的空白字符合并为一个空格，这种做法其实很有道理，因为这样可以把 HTML 文档的布局和文档内容自身的布局给区别开来

相当于你文档中使用空格换行来达到所谓的美观需求的布局，HTML 是不予理会的，如果你想要内容在网页上以美美哒的形式展现出来，那好，请使用 HTML 的语法来实现

# 1、基础

## 1、!DOCTYPE

<!DOCTYPE html> 是一个声明，表示该文档是由 HTML5 进行编写的。

每个网页第一行必须是这个！

<!DOCTYPE> 声明必须是 HTML 文档的第一行，位于 <html> 标签之前。请始终向 HTML 文档添加 <!DOCTYPE> 声明，这样浏览器才能获知文档类型。

## 2、html

html 元素用于告诉告诉浏览器其自身是一个 HTML 文档。

<html> 与 </html> 标签限定了文档的开始点和结束点，在它们之间是文档的头部和主体。

正如您所了解的那样，文档的头部由 <head> 标签定义，而主体由 [<body>](https://man.ilovefishc.com/page/body.html) 标签定义。

**注意：**即使 html 元素是文档的根元素，它也不包含 doctype 元素。[doctype](https://man.ilovefishc.com/page/!DOCTYPE.html) 元素必须位于 html 元素之前。

**<meta charset="UTF-8">编码格式时utf-8形式**

## 3、head

<head> 标签用于定义文档的头部，它是所有头部元素的容器。

<head> 中的元素可以引用脚本、指示浏览器在哪里找到样式表、提供元信息等等。

**注意：**下面这些标签可用在 head 部分：[<base>](https://man.ilovefishc.com/page/base.html" \t "_blank), [<link>](https://man.ilovefishc.com/page/link.html), [<meta>](https://man.ilovefishc.com/page/meta.html), [<script>](https://man.ilovefishc.com/page/script.html), [<style>](https://man.ilovefishc.com/page/style.html)，[<title>](https://man.ilovefishc.com/page/title.html" \t "_blank)。

[<title>](https://man.ilovefishc.com/page/title.html) 是 head 部分中唯一必需的标签。

**实现网页尺寸 “自适应”：** <meta name="viewport" content="width=device-width, initial-scale=1.0">

## 4、body

<body> 标签定义文档的主体。

body 元素包含文档的所有内容（比如文本、超链接、图像、表格和列表等等）。

## 5、<h1> - <h6>

<h1> - <h6> 标签可定义标题。

<h1> 定义最大的标题。<h6> 定义最小的标题。

由于 h 元素拥有确切的语义，因此请慎重地选择恰当的标签层级来构建文档的结构。因此，**请不要利用标题标签来改变同一行中的字体大小**。

相反，应当使用层叠样式表定义来达到漂亮的显示效果

## 6、P

<p> 标签用于定义一个段落。

p 元素会自动在其前后创建一些空白。浏览器会自动添加这些空间，你也可以通过样式表进行约束。

## 7、meta

<meta> 标签用于描述页面内容，关键词，作者，最新修订时间以及其它元信息

<meta> 标签的内容不会显示在网页中，但经常被机器（比如网页爬虫）解析。

元信息可以被浏览器（指导它如何加载和显示页面）所使用，也可以有利于搜索引擎收录页面（指定关键词）。

HTML5 通过 <meta> 标签引入一个新的方法，让网页尺寸实现自适应。

**注意：**

<meta> 标签永远位于 [<head>](https://man.ilovefishc.com/page/head.html) 标签内部。

content 属性必须和 name 或 http-equiv 属性共存亡。

为了让网页尺寸实现自适应，你应该在 HTML 文档中加入下面代码：

<meta name="viewport" content="width=device-width, initial-scale=1.0">

**属性：**

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| charset | character\_set | 指定 HTML 文档的编码。 |
| content | text | 指定与 http-equiv 或 name 属性相关的值。 |
| http-equiv | content-type、default-style、refresh | content-type：另一种声明 HTML 文档作用字符编码的方法（<meta http-equiv="content-type" content="text/html charset=UTF-8">）。 default-style：指定页面优先使用的样式表。 refresh：以秒为单位指定一个时间间隔，在此时间过去之后将从服务器重新载入当前页面。也可以另行指定一个 URL 让浏览器载入。 |
| name | application-name、author、description、generator、keywords、viewport | application-name：当前页所属 Web 应用系统的名称。 author：当前页的作者名。 description：当前页的内容描述。 generator：用来生成 HTML 的软件名称。 keywords：当前页的关键词。 viewport：网页尺寸自适应 |

## 8、style

<style> 标签用于为 HTML 文档定义样式信息。

style 元素可以出现在 HTML 文档中的各个部分，一个文档可以包含多个 style 元素。

https://fishc.com.cn/thread-108391-1-1.html

**属性：**

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| [media](https://man.ilovefishc.com/page/media.html) | screen、tty、tv、projection、handheld、print、braille、aural、all | 指定样式适用的媒体。 |
| [scoped](https://man.ilovefishc.com/page/scoped.html) | scoped | 指定样式的作用范围（如果使用该属性，则样式仅仅应用到 style 元素的父元素及其子元素）。 |
| [type](https://man.ilovefishc.com/page/type.html) | text/css | 指定样式的类型。 |

## 9、base

<base> 标签用于设置相对 URL 的解析基准。

base 元素可以用来设置一个基准 URL，让 HTML 文档中的相对链接在此基础上进行解析。

**注意：**

<base> 标签必须位于 [<head>](https://man.ilovefishc.com/page/head.html) 标签内部，并尽量靠前，以便随后的元素中的相对 URL 可以用上其设置的基准 URL。

**属性：**

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| href | URL | 指定该 HTML 文档中所有相对链接的基准 URL。 |
| target | \_blank、\_parent、\_self、\_top、framename | 指定在何处打开超链接。 \_blank：在新窗口中打开 \_parent：在当前的父窗口中打开，如果不存在父窗口，此选项的行为方式与 \_self 等同 \_self：当前窗口打开（默认） \_top：在整个窗口中打开 framename：在指定的框架中打开 |

## 10、script

<script> 标签用于在 HTML 文档中加入脚本（例如 JavaScript）。

script 元素既可以直接定义内嵌脚本语句，也可以通过 src 属性引用外部脚本文件。

script 元素可以出现在 HTML 文档中的各个部分，一个文档可以包含多个 script 元素。

**注意：**

每定义或导入一段脚本，都需要使用一个 script 元素。

不能用同一个 script 元素既定义内嵌脚本，又引用外部脚本。

**属性：**

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| type | media\_type | 指定所定义或引用的脚本类型（如果使用 JavaScript 脚本，这个属性可以忽略）。 |
| async | async | 告诉浏览器异步执行脚本。 注意：该属性只能用于引用外部脚本文件，对内嵌脚本不起作用。 |
| charset | charset | 指定外部脚本文件中使用的字符编码。 注意：该属性只能与 src 属性一起使用。 |
| defer | defer | 告诉浏览器延迟执行脚本（直到页面载入并解析完毕后再执行脚本）。 注意：该属性只能用于引用外部脚本文件，对内嵌脚本不起作用。 |
| src | URL | 指定外部脚本文件的 URL。 |

## 11、div

<div> 可定义文档中的分区或节（division/section）。

<div> 标签可以把文档分割为独立的、不同的部分。它可以用作严格的组织工具，并且不使用任何格式与其关联。

如果用 id 或 class 来标记 <div>，那么该标签的作用会变得更加有效。

<div> 是一个块级元素。这意味着它的内容自动地开始一个新行。实际上，换行是 <div> 固有的唯一格式表现。可以通过 <div> 的 class 或 id 应用额外的样式。

不必为每一个 <div> 都加上类或 id，虽然这样做也有一定的好处。

可以对同一个 <div> 元素应用 class 或 id 属性，但是更常见的情况是只应用其中一种。这两者的主要差异是，class 用于元素组（类似的元素，或者可以理解为某一类元素），而 id 用于标识单独的唯一的元素。

注释：

<div> 是一个块级元素，也就是说，浏览器通常会在 div 元素前后放置一个换行符。

提示：请使用 <div> 元素来组合块级元素，这样就可以使用样式对它们进行格式化。

## 12、noscript

<noscript> 标签用来向不支持 JavaScript 的浏览器显示一些替代内容。

如果浏览器支持脚本，那么它不会显示出 noscript 元素中的内容。



## 13、br

<br>标签用于插入一个简单的换行符。

## 14、span

<span> 标签用于组合文档中的行内元素。

span 没有特定的样式。只有对它设置样式的时候，才会产生视觉上的表现。

当你想要对局部文本设置样式的时候，span元素就很有用。



# 2、图片

## 1、img

<img> 标签用于向网页中嵌入一幅图像。

从技术上讲，<img> 标签并不会在网页中插入图像，而是从网页上链接图像。<img> 标签创建的是被引用图像的占位空间。

**注意：**<img> 标签有两个必需的属性：src 属性 和 alt 属性。

**属性：**

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| alt | text | 指定图像的替代文本。 |
| src | URL | 指定显示图像的 URL。 |
| width | px(pixels)、% | 设置图像的宽度（像素或百分比）。 |
| height | px(pixels)、% | 定义图像的高度（像素或百分比）。 |
| ismap | URL | 将图像定义为服务器端图像映射。 |
| longdesc | URL | 指定包含长的图像描述文档的 URL。 |
| usemap | URL | 将图像定义为客户器端图像映射。 |

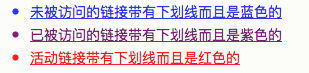
# 3、链接

## 1、a

<a> 标签用于定义超链接，超链接可以让用户从一个网页跳转到另一个网页。

<a> 标签最重要的属性是 href 属性，它指定了链接目标的 URL。

在浏览器中，通常超链接的默认外观是：



**注意：**

在 HTML 4.01 中，<a> 标签可以是超链接或锚。在 HTML5 中，<a> 标签始终是超链接，但是如果未设置 href 属性，则只是超链接的占位符。

HTML5 提供了一些新属性，同时不再支持一些 HTML 4.01 属性。

**属性：**

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| download | filename | 指定被下载的超链接目标。 |
| href | URL | 指定链接指向页面的 URL。 |
| hreflang | language\_code | 指定被链接文档的语言。 |
| media | media\_query | 指定被链接文档是为何种媒介/设备优化的。 |
| rel | text | 规定当前文档与被链接文档之间的关系。 |
| target | \_blank、\_parent、\_self、\_top、framename | 指定在何处打开超链接。 \_blank：在新窗口中打开 \_parent：在当前的父窗口中打开，如果不存在父窗口，此选项的行为方式与 \_self 等同 \_self：当前窗口打开（默认） \_top：在整个窗口中打开 framename：在指定的框架中打开 |
| type | MIME type | 指定被链接文档的的 MIME 类型。 |

## 2、link

<link> 标签用于指定外部资源。

link 元素最常见的用途是链接样式表。

**注意：**

link 元素定义了 6 个属性，其中 rel 属性是必选的，它说明了当前文档与被链接资源之间的关系。

**属性：**

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| href | URL | 指定被链接资源的 URL。 |
| hreflang | language\_code | 指定被链接资源使用的语言。 |
| sizes | HeightxWidth | 指定图标的大小（比如 sizes="16x16"）。 |
| [media](https://man.ilovefishc.com/page/media1.html) | media\_query | 指定被链接的资源将被显示到什么设备上。 |
| [rel](https://man.ilovefishc.com/page/rel.html) | alternate、author、help、icon、licence、next、pingback、prefetch、prev、search、sidebar、stylesheet、tag | 指定当前文档与被链接资源之间的关系。 |
| [type](https://man.ilovefishc.com/page/type1.html) | MIME\_type | 规定被链接文档的 MIME 类型。 |

link 最常见的另一个用法就是链接网站图标 —— favicon。

关联网站图标，只需要将 rel 属性设置为 "icon"，然后设置 type 属性为 "image/x-icon"，最后在 href 属性中指定图标的位置即可：

<link rel="icon" href="http://www.fishc.com/favicon.ico" type="image/x-icon">

# 4、格式

# 5、特殊格式

## 1、nav

<nav> 标签定义导航链接的部分。

<nav> 标签是 HTML 5 中的新标签。



## 2、section

<section> 标签定义文档中的节。

节就是章节、页眉、页脚或文档中的其他部分。

<section> 标签是 HTML 5 中的新标签。



# 6、核心属性

## 1、class

class 属性规定元素的类名（classname）。

class 属性大多数时候用于指向样式表中的类（class）。不过，也可以利用它通过 JavaScript 来改变带有指定 class 的 HTML 元素。

## 2、id

id 属性规定 HTML 元素的唯一的 id。

id 在 HTML 文档中必须是唯一的。id 属性可用作链接锚（link anchor），通过 JavaScript（HTML DOM）或通过 CSS 为带有指定 id 的元素改变或添加样式。



## 3、style

style 属性规定元素的行内样式（inline style）。

**注意：**style 属性将覆盖任何全局的样式设定，例如在 <style> 标签或在外部样式表中规定的样式。

**属性：**

|  |  |
| --- | --- |
| **值** | **描述** |
| style\_definition | 一个或多个由分号分隔的 CSS 属性和值。 |

## 4、title

title 属性规定关于元素的额外信息。

这些信息通常会在鼠标移到元素上时显示一段工具提示文本（tooltip text）。

**属性：**

|  |  |
| --- | --- |
| **值** | **描述** |
| text | 规定元素的工具提示文本（tooltip text）。 |

# 7、列表

## 1、ul

<ul> 标签用于定义无序列表。



## 2、li

<li> 标签用于定义列表中的项目。

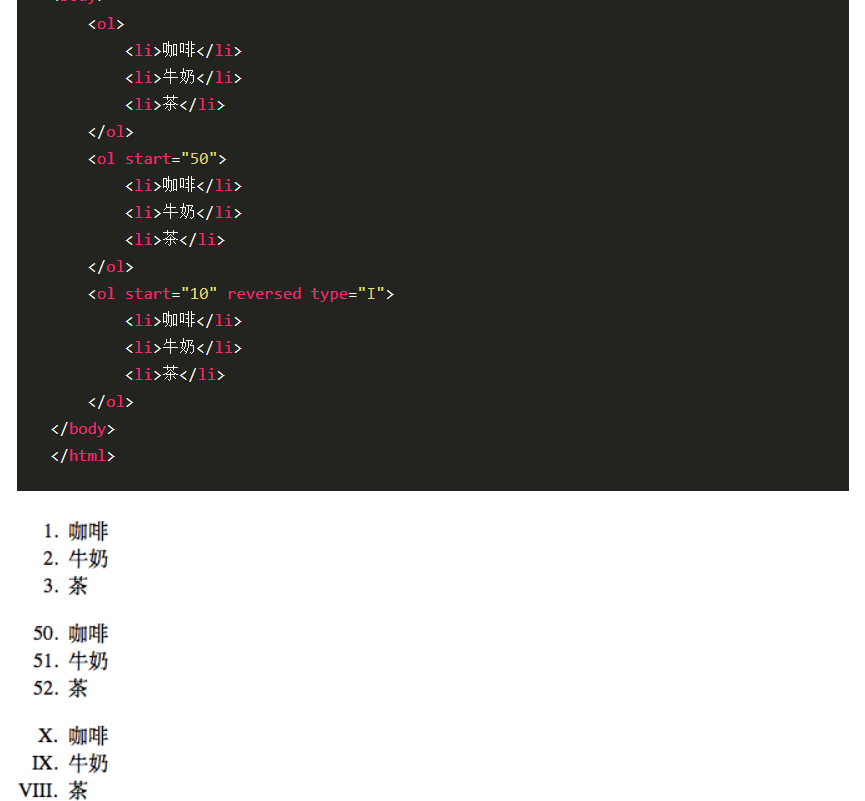
<li> 标签可用在有序列表 [<ol>](https://man.ilovefishc.com/page/ol.html) 和无序列表 [<ul>](https://man.ilovefishc.com/page/ul.html) 中。

## 3、ol

<ol> 标签用于定义有序列表。

**属性：**

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| reversed | reversed | 规定列表顺序为降序。(9, 8, 7, …) |
| start | number | 规定有序列表的起始值。 |
| type | 1、A、a、I、i | 规定在列表中使用的标记类型。 |



## 4、dt

<dt> 标签用于定义列表中的项目（即术语部分）。



## 5、dl

<dl> 标签定义了一个包含术语定义以及描述的列表

## 6、dd

<dd> 标签用于定义列表中项目的描述部分

# 8、格式化

## 1、abbr

<abbr> 标签定义简称或缩写，比如 “WWW” 或 “NATO”。

通过对缩写进行标记，能够为浏览器、拼写检查和搜索引擎提供有用的信息。

abbr 元素用于定义简称或者缩写，配合全局属性 title，可以指定该缩写代表的完整含义。



## 2、pre

<pre> 标签用于定义预格式化的文本。

pre 元素中的文本通常会保留空格和换行符，而文本自身也会呈现为等宽字体。



## 3、code

<code> 标签用于定义计算机代码片段。

如果只是为了达到某种视觉效果而使用这个标签的话，建议使用 CSS 实现，这样可能会取得更丰富的效果。



## 4、var

<var> 标签用于定义程序的变量。

如果只是为了达到某种视觉效果而使用这个标签的话，建议使用 CSS 实现，这样可能会取得更丰富的效果。

## 5、samp

<samp> 标签用于定义计算机程序的输出。

如果只是为了达到某种视觉效果而使用这个标签的话，建议使用 CSS 实现，这样可能会取得更丰富的效果。

## 6、kbd

<kbd> 标签用于定义键盘输入文本。



## 7、q

<q> 标签定义较短的引用。

浏览器通常会在引用内容的两侧添加引号。

<q> 标签在本质上与[<blockquote>](https://man.ilovefishc.com/page/blockquote.html) 是一样的。不同之处在于它们的显示和应用。

<q> 标签用于简短的行内引用。如果需要引用较长的内容，建议使用 <blockquote> 标签。

**属性：**

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| cite | URL | 规定引用的来源。 |



## 8、blockquote

<blockquote> 标签定义块引用。

<blockquote> 与 </blockquote> 标签之间的所有文本都会从常规文本中分离出来，通常浏览器会在左、右两侧添加缩进（增加外边距）。也就是说，块引用拥有它们自己的空间。

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| cite | URL | 规定引用的来源。 |



## 9、cite

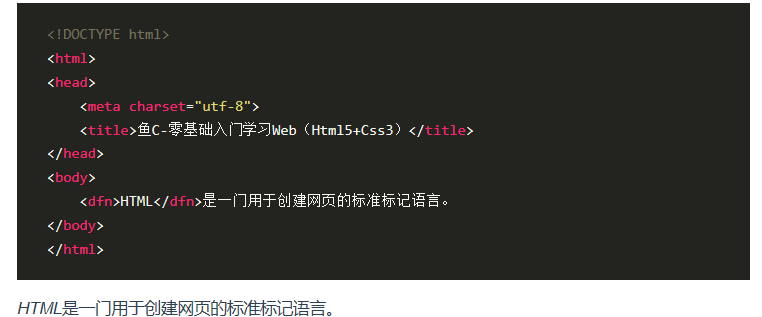
<cite> 标签定义作品（比如书籍、歌曲、电影、电视节目、绘画、雕塑等）的标题。



## 10、dfn

<dfn> 标签表现定义中的术语。

术语又称技术名词，是在特定专业领域中一般概念的词语指称，一个术语表示一个概念。



## 11、address

<address> 标签定义文档或文章的作者/拥有者的联系信息。

如果 <address> 位于 [<body>](https://man.ilovefishc.com/page/body.html) 标签内，则它表示文档的联系信息。

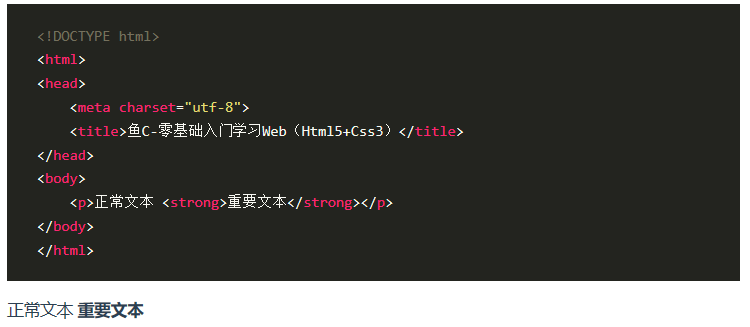
如果 <address> 位于[<article>](https://man.ilovefishc.com/page/article.html) 标签内，则它表示文章的联系信息。



## 12、strong

<strong> 标签用于定义表示重要的文本。

提示：如果只是为了达到某种视觉效果而使用这个标签的话，强烈建议使用 CSS 代替。



## 13 b

b 元素仅表示粗体的文本，并**没有**附带任何表示重要的语义。

根据 HTML5 的规范，如果你只是想实现加粗效果，推荐使用 css 样式来实现：b {font-weight: bolder;}

## 14、ruby

<ruby> 标签定义注音符号。

注意：

ruby 元素需要与 [rt](https://man.ilovefishc.com/page/rt.html) 元素和 [rp](https://man.ilovefishc.com/page/rp.html) 元素搭配使用。其中，[rt](https://man.ilovefishc.com/page/rt.html) 元素用来标记注音符号，[rp](https://man.ilovefishc.com/page/rp.html) 元素则用来标记当浏览器不支持 ruby 元素时所显示的内容。



## 15、bdo

bdo 标签修改默认的文本方向。

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| dir | ltr、rtl | 定义文字的方向（ltr -> left to right；rtl -> right to left） |

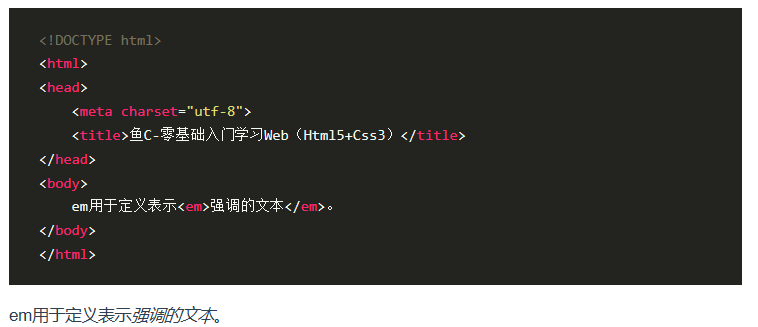


## 16、em

em 元素表示强调的语义，通常是以斜体的形式呈现；i 元素也可以让内容倾斜，只不过它没有附带任何表示强调的语义。

<em> 标签用于定义表示强调的文本。

提示：如果只是为了达到某种视觉效果而使用这个标签的话，强烈建议使用 CSS 代替。

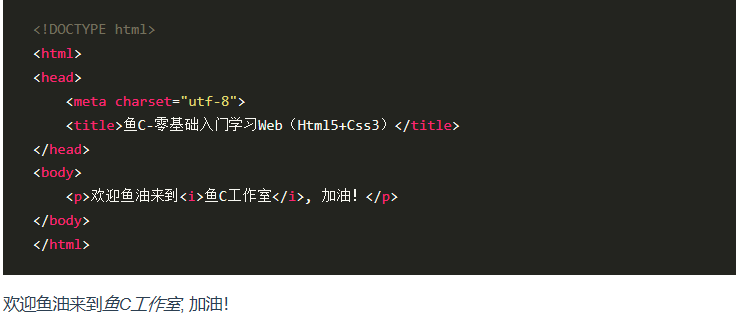


## 17、i

<i> 标签用于定义表示斜体的文本。

i 元素仅表示斜体的文本，并**没有**附带任何表示强调的语义。

根据 HTML5 的规范，如果你只是想实现斜体效果，推荐使用 css 样式来实现：i {font-style: italic;}



## 18、del

del 元素和 ins 元素搭配，用来描述文档中的更新和修正。

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| cite | URL | 指向另外一个文档的 URL，此文档可解释文本被删除的原因。 |
| datetime | YYYY-MM-DDThh:mm:ssTZD | 定义文本被删除的日期和时间。 |



## 19、ins

<ins> 标签用于定义新插入的文本。

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| cite | URL | 指向另外一个文档的 URL，此文档可解释文本被删除的原因。 |
| datetime | YYYYMMDD | 定义文本被删除的日期和时间。 |

## 20、s

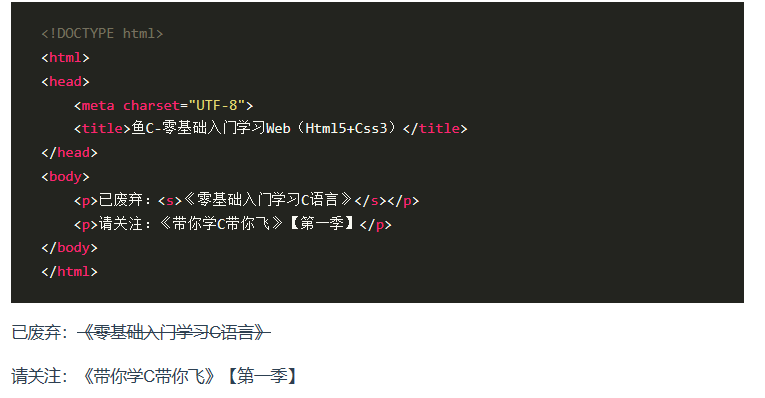
del 元素呈现的样式是删除线，默认同样的样式有 s 元素。

不过两者不能混用，因为 s 元素与 del 元素的语义是不同的，s 元素用于定义那些表示不再正确的内容。

<s>标签用来定义那些不正确的文本

<s> 对那些不正确、不准确或者没有用的文本进行标识。

<s> 标签不应该用来定义替换的或者删除的文本。



## 21、u

<u>标签定义与常规文本风格不同的文本。

**注意：**

<u> 标签通常可以用于定义像拼写错误的单词或者汉语中的专有名词，但注意要与超链接区分开来。

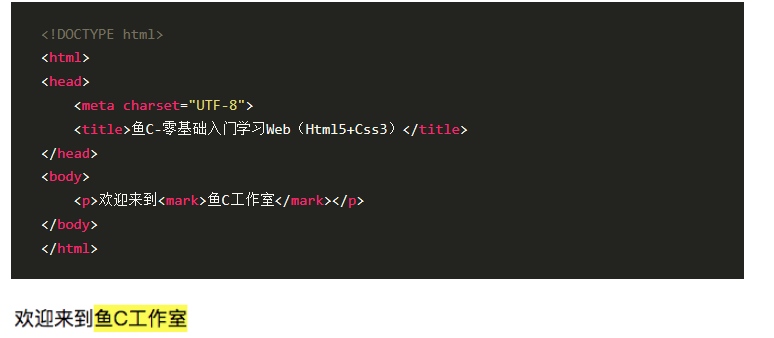
根据 HTML5 的规范，如果你只是想为文本添加下划线效果，推荐使用 css 样式来实现：u {text-decoration: underline;}



## 22、mark

<mark> 标签用于定义带有标记的文本。

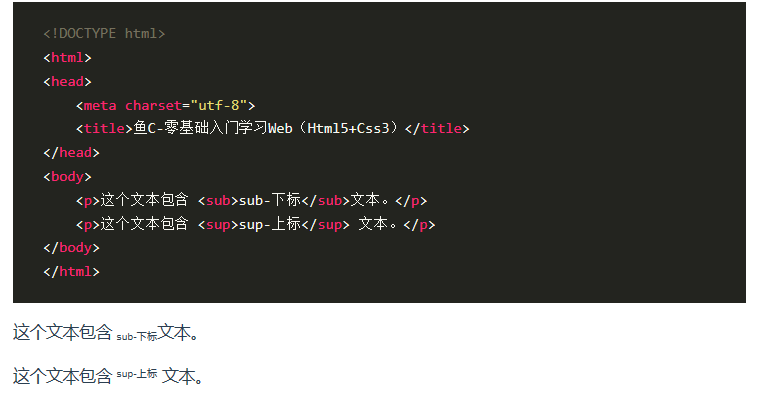
请在需要突出显示某部分文本时使用该标签。



## 23、sub

<sub> 标签用于定义下标文本。

无论是 <sub> 标签还是和它对应的 [<sup>](https://man.ilovefishc.com/page/sup.html) 标签，在数学等式、科学符号和化学公式中都非常有用。

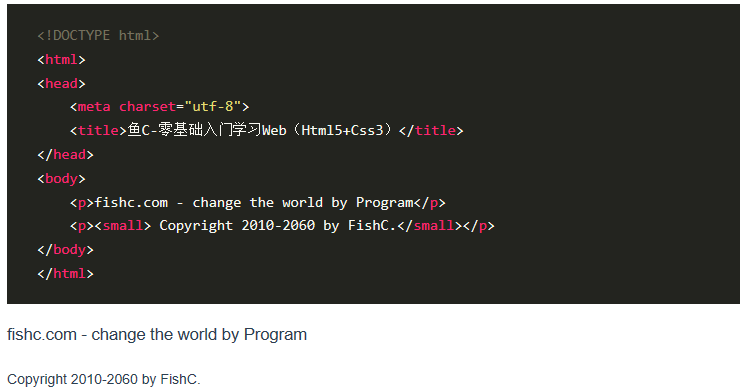


## 24、sup

<sup> 标签用于定义上标文本。

## 25、small

<small> 标签用于定义更小字体的文本（比如旁注）。



## 26、object

<object> 标签定义一个嵌入的对象。

<object> 标签用于包含对象，比如图像、音频、视频、Java applets、ActiveX、PDF 以及 Flash。object 的初衷是取代 img 和 applet 元素。不过由于漏洞以及缺乏浏览器支持，这一点并未实现。

object 对象提供了解决方案。如果未显示 object 元素，就会执行位于 <object> 和 </object> 之间的代码。通过这种方式，能够嵌套多个 object 元素（每个对应一个浏览器）。

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **说明** |
| form | form\_id | 规定对象所属的一个或多个表单。 |
| height | pixels | 规定对象的高度。 |
| width | pixels | 规定对象的宽度。 |
| name | name | 为对象规定名称。 |
| type | MIME\_type | 规定 data 属性中规定的数据的 MIME 类型。 |
| usemap | mapname | 规定与对象一同使用的客户端图像映射的名称。 |



## 27、footer

<footer> 标签定义文档或节的页脚。

页脚通常包含文档的作者、版权信息、使用条款链接、联系信息等等。



## 28、header

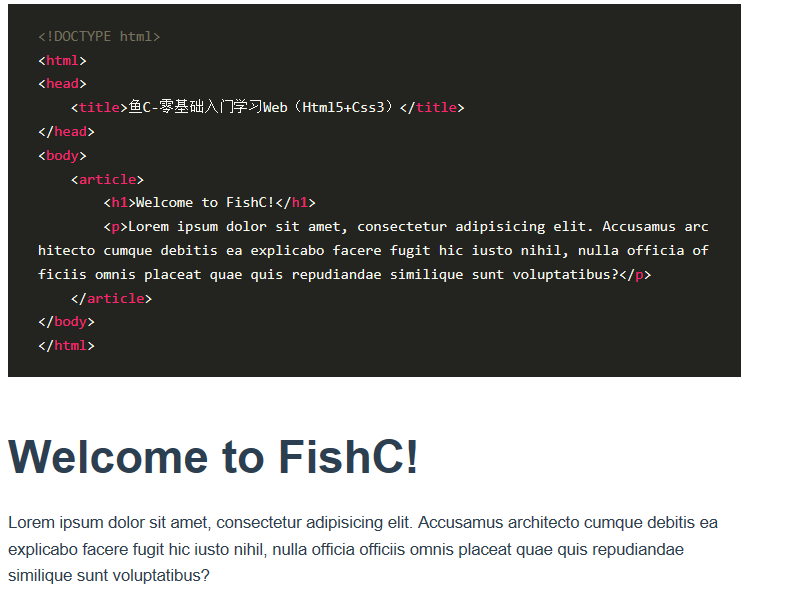
<header> 标签定义文档的页眉（介绍信息）。



## 29、article

<article> 标签定义外部的内容。

外部内容可以是来自一个外部的新闻提供者的一篇新的文章，或者来自 blog 的文本，或者是来自论坛的文本。亦或是来自其他外部源内容。



# 9、表单/表格

## 1、button

<button> 标签用于定义一个按钮。

使用 input 元素也可以设置成按钮的样式，但是在 button 元素的内部，你可以放置文本或图像，而 input 元素则不行。

请始终位 button 元素设置 type 属性，因为不同的浏览器为其指定的默认值可能不同。

**属性：**

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| autofocus | autofocus | 指定当页面加载的时候，按钮将获得焦点。 |
| disabled | disabled | 禁用按钮。 |
| form | form\_id | 指定按钮所关联的表单 ID。 |
| formaction | url | 覆盖 form 元素的 action 属性。注释：该属性与 type="submit" 配合使用。 |
| formenctype | application/x-www-form-urlencoded、multipart/form-data、text/plain | 覆盖 form 元素的 enctype 属性。注释：该属性与 type="submit" 配合使用。 |
| formmethod | get、post | 覆盖 form 元素的 method 属性。注释：该属性与 type="submit" 配合使用。 |
| formnovalidate | formnovalidate | 覆盖 form 元素的 novalidate 属性。注释：该属性与 type="submit" 配合使用。 |
| formtarget | \_blank、\_self、\_parent、\_top、framename | 覆盖 form 元素的 target 属性。注释：该属性与 type="submit" 配合使用。 |
| name | button\_name | 指定按钮的名称。 |
| type | button、reset、submit | 指定按钮的类型。 |
| value | text | 指定按钮的初始值。注释：可由脚本进行修改。 |

