**HTML**

|  |  |  |
| --- | --- | --- |
| **开始标签 \*** | **元素内容** | **结束标签 \*** |
| <p> | 这是一个段落 | </p> |
| <a href="default.htm"> | 这是一个链接 | </a> |
| <br> | 换行 |  |

**<link> 元素**

<link> 标签定义了文档与外部资源之间的关系。

<link> 标签通常用于链接到样式表:

<head>

<link rel="stylesheet" type="text/css" href="mystyle.css">

</head>

## <style> 元素

<style> 标签定义了HTML文档的样式文件引用地址.

在<style> 元素中你也可以直接添加样式来渲染 HTML 文档:

<head>

<style type="text/css">

body {background-color:yellow}

p {color:blue}

</style>

</head>

HTML中，有两种类型的列表：

* 无序列表 **ul** - 列表项标记用特殊图形（如小黑点、小方框等）
* 有序列表 **ol** - 列表项的标记有数字或字母

## HTML无序列表

无序列表是一个项目的列表，此列项目使用粗体圆点（典型的小黑圆圈）进行标记。

无序列表使用 <ul> 标签

<ul>  
<li>Coffee</li>  
<li>Milk</li>  
</ul>

浏览器显示如下：

* Coffee
* Milk

## HTML 有序列表

同样，有序列表也是一列项目，列表项目使用数字进行标记。 有序列表始于 <ol> 标签。每个列表项始于 <li> 标签。

列表项使用数字来标记。

<ol>  
<li>Coffee</li>  
<li>Milk</li>  
</ol>

浏览器中显示如下：

1. Coffee
2. Milk

## HTML 分组标签

|  |  |
| --- | --- |
| **标签** | **描述** |
| [<div>](https://www.runoob.com/tags/tag-div.html) | 定义了文档的区域，块级 (block-level) |
| [<span>](https://www.runoob.com/tags/tag-span.html) | 用来组合文档中的行内元素， 内联元素(inline) |

## HTML 表单

表单是一个包含表单元素的区域。

表单元素是允许用户在表单中输入内容,比如：文本域(textarea)、下拉列表、单选框(radio-buttons)、复选框(checkboxes)等等。

表单使用表单标签 <form> 来设置:

<form>  
.  
*input 元素*  
.  
</form>

## HTML 表单 - 输入元素

多数情况下被用到的表单标签是输入标签（<input>）。

输入类型是由类型属性（type）定义的。大多数经常被用到的输入类型如下：

## 单选按钮（Radio Buttons）

<input type="radio"> 标签定义了表单单选框选项

<form>  
<input type="radio" name="sex" value="male">Male<br>  
<input type="radio" name="sex" value="female">Female  
</form>

浏览器显示效果如下:

窗体顶端

Male  
Female

窗体底端

## 复选框（Checkboxes）

<input type="checkbox"> 定义了复选框. 用户需要从若干给定的选择中选取一个或若干选项。

<form>  
<input type="checkbox" name="vehicle" value="Bike">I have a bike<br>  
<input type="checkbox" name="vehicle" value="Car">I have a car  
</form>

浏览器显示效果如下:

窗体顶端

I have a bike  
I have a car

窗体底端

## 提交按钮(Submit Button)

<input type="submit"> 定义了提交按钮.

当用户单击确认按钮时，表单的内容会被传送到另一个文件。表单的动作属性定义了目的文件的文件名。由动作属性定义的这个文件通常会对接收到的输入数据进行相关的处理。:

<form name="input" action="html\_form\_action.php" method="get">  
Username: <input type="text" name="user">  
<input type="submit" value="Submit">  
</form>

浏览器显示效果如下:

窗体顶端

Username: 

窗体底端

VIDEO

<video> 元素提供了 播放、暂停和音量控件来控制视频。

同时 <video> 元素也提供了 width 和 height 属性控制视频的尺寸.如果设置的高度和宽度，所需的视频空间会在页面加载时保留。如果没有设置这些属性，浏览器不知道大小的视频，浏览器就不能再加载时保留特定的空间，页面就会根据原始视频的大小而改变。

<video> 与</video> 标签之间插入的内容是提供给不支持 video 元素的浏览器显示的。

<video> 元素支持多个 <source> 元素. <source> 元素可以链接不同的视频文件。

AUDIO

control 属性供添加播放、暂停和音量控件。

在<audio> 与 </audio> 之间你需要插入浏览器不支持的<audio>元素的提示文本 。

<audio> 元素允许使用多个 <source> 元素. <source> 元素可以链接不同的音频文件

**Css**

## id 选择器

id 选择器可以为标有特定 id 的 HTML 元素指定特定的样式。

HTML元素以id属性来设置id选择器,CSS 中 id 选择器以 "#" 来定义。

## class 选择器

class 选择器用于描述一组元素的样式，class 选择器有别于id选择器，class可以在多个元素中使用。

class 选择器在HTML中以class属性表示, 在 CSS 中，类选择器以一个点"."号显示：

## 文本的对齐方式

文本排列属性是用来设置文本的水平对齐方式。

文本可居中或对齐到左或右,两端对齐.

当text-align设置为"justify"，每一行被展开为宽度相等，左，右外边距是对齐

## 文本修饰

text-decoration 属性用来设置或删除文本的装饰。

从设计的角度看 text-decoration属性主要是用来删除链接的下划线：

## 文本缩进

文本缩进属性是用来指定文本的第一行的缩进。

## 实例

p {text-indent:50px;}

**链接样式**

链接的样式，可以用任何CSS属性（如颜色，字体，背景等）。

特别的链接，可以有不同的样式，这取决于他们是什么状态。

这四个链接状态是：

* a:link - 正常，未访问过的链接
* a:visited - 用户已访问过的链接
* a:hover - 当用户鼠标放在链接上时
* a:active - 链接被点击的那一刻
* 在CSS定义中，a:hover 必须被置于 a:link 和 a:visited 之后，才是有效的。
* **注意：** 在 CSS 定义中，a:active 必须被置于 a:hover 之后，才是有效的。

**盒子模型**



* **Margin(外边距)** - 清除边框外的区域，外边距是透明的。
* **Border(边框)** - 围绕在内边距和内容外的边框。
* **Padding(内边距)** - 清除内容周围的区域，内边距是透明的。
* **Content(内容)** - 盒子的内容，显示文本和图像。

## 隐藏元素 - display:none或visibility:hidden

隐藏一个元素可以通过把display属性设置为"none"，或把visibility属性设置为"hidden"。但是请注意，这两种方法会产生不同的结果。

visibility:hidden可以隐藏某个元素，但隐藏的元素仍需占用与未隐藏之前一样的空间。

display:none可以隐藏某个元素，且隐藏的元素不会占用任何空间。也就是说，该元素不但被隐藏了，而且该元素原本占用的空间也会从页面布局中消失。

## Float（浮动），会使元素向左或向右移动，其周围的元素也会重新排列。

## 清除浮动 - 使用 clear

元素浮动之后，周围的元素会重新排列，为了避免这种情况，使用 clear 属性。

clear 属性指定元素两侧不能出现浮动元素。

## 元素居中对齐

要水平居中对齐一个元素(如 <div>), 可以使用 **margin: auto;**

## 文本居中对齐

文本在元素内居中对齐，可以使用 **text-align: center;**

## 图片居中对齐

要让图片居中对齐, 可以使用 **margin: auto;**

## CSS 弹性盒子内容

弹性盒子由弹性容器(Flex container)和弹性子元素(Flex item)组成。

弹性容器通过设置 display 属性的值为 flex 或 inline-flex将其定义为弹性容器。

弹性容器内包含了一个或多个弹性子元素。

flex-direction 属性指定了弹性子元素在父容器中的位置。

flex-direction的值有:

* row：横向从左到右排列（左对齐），默认的排列方式。
* row-reverse：反转横向排列（右对齐，从后往前排，最后一项排在最前面。
* column：纵向排列。
* column-reverse：反转纵向排列，从后往前排，最后一项排在最上面。

## justify-content 属性

内容对齐（justify-content）属性应用在弹性容器上，把弹性项沿着弹性容器的主轴线（main axis）对齐。

## align-content 属性

align-content 属性用于修改 flex-wrap 属性的行为。类似于 align-items, 但它不是设置弹性子元素的对齐，而是设置各个行的对齐。

**JavaScript**

**JavaScript 显示数据**

JavaScript 可以通过不同的方式来输出数据：

* 使用 **window.alert()** 弹出警告框。
* 使用 **document.write()** 方法将内容写到 HTML 文档中。
* 使用 **innerHTML** 写入到 HTML 元素。
* 使用 **console.log()** 写入到浏览器的控制台。

## 操作 HTML 元素

如需从 JavaScript 访问某个 HTML 元素，您可以使用 document.getElementById(*id*) 方法。

请使用 "id" 属性来标识 HTML 元素，并 innerHTML 来获取或插入元素内容

 数据类型

**值类型(基本类型)**：字符串（String）、数字(Number)、布尔(Boolean)、对空（Null）、未定义（Undefined）、Symbol。

**引用数据类型**：对象(Object)、数组(Array)、函数(Function)。

## HTML事件

|  |  |
| --- | --- |
| **事件** | **描述** |
| onchange | HTML 元素改变 |
| onclick | 用户点击 HTML 元素 |
| onmouseover | 鼠标指针移动到指定的元素上时发生 |
| onmouseout | 用户从一个 HTML 元素上移开鼠标时发生 |
| onkeydown | 用户按下键盘按键 |
| onload | 浏览器已完成页面的加载 |

# this 关键字

面向对象语言中 this 表示当前对象的一个引用。

但在 JavaScript 中 this 不是固定不变的，它会随着执行环境的改变而改变。

* 在方法中，this 表示该方法所属的对象。
* 如果单独使用，this 表示全局对象。
* 在函数中，this 表示全局对象。
* 在函数中，在严格模式下，this 是未定义的(undefined)。
* 在事件中，this 表示接收事件的元素。
* 类似 call() 和 apply() 方法可以将 this 引用到任何对象。

## HTML DOM (文档对象模型)

* 通过 id 找到 HTML 元素
* 通过标签名找到 HTML 元素
* 通过类名找到 HTML 元素

## 改变 HTML 内容

修改 HTML 内容的最简单的方法是使用 innerHTML 属性。

如需改变 HTML 元素的内容，请使用这个语法：

document.getElementById(*id*).innerHTML=*新的 HTML*

## 改变 HTML 属性

如需改变 HTML 元素的属性，请使用这个语法：

document.getElementById(*id*).*attribute=新属性值*

## HTMLCollection 对象

getElementsByTagName() 方法返回 [HTMLCollection](https://www.runoob.com/jsref/dom-htmlcollection.html) 对象。

Js高级

[javascript——原型与原型链 - 雅昕 - 博客园 (cnblogs.com)](https://www.cnblogs.com/loveyaxin/p/11151586.html)