# HTML

尖括号括起的内容前面有斜线意味着一个标签的结束。这在HTML中是可选项，而在XHTML是必须的，包括XML模式下的HTML元素。

参考：<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

Basic elements：

<html>

Document metadata:

<base><head><link><meta><style><title>

Content sectioning:

<address><article><aside><footer><header><h1><h2><h3><h4><h5><h6><hgroup><nav><section>

Text content：

<dd><div><dl><dt><figcaption><figure><hr><li><main><ol><p><pre><ul>

Inline text semantics：

<a><abbr><b><bdi><bdo><br><cite><code><data><dfn><em><i><kbd><mark><q><rp><rt><rtc><ruby><s><samp><small><span><strong><sub><sup><time><u><var><wbr>

Image and multimedia：

<area><audio><map><track><video>

Embedded content：

<embed><object><param><source>

Scripting：

<canvas><noscript><script>

Demarcating edits：

<del><ins>

Table content：

<caption><col><colgroup><table><tbody><td><tfoot><th><thead><tr>

Forms：

<button><datalist><fieldset><form><input><label><legend><meter><optgroup><option><output><progress><select><textarea>

Interactive elements：

<details><dialog><menu><menuitem><summary>

Web Components：

<content><element><shadow><template>

Obsolete and deprecated elements：

<acronym><applet><basefont><big><blink><center><command><content><dir><font><frame><frameset><isindex><keygen><listing><marquee><multicol><nextid><noembed><plaintext><spacer><strike><tt><xmp>

标签：

标签(Tag)的名字不区分大小写。 但是,W3C (维护了HTML标准的国际性性联盟)建议使用小写 ( XHTML 同样要求使用小写). HTML 为由一对尖括号（<>）所括起来的内容以特定含义. 这样的标识称为一个 标签（tag）。有一个起始标签（start tag）和一个结束标签（end tag）。结束标签与起始标签在内容上是一样的，但结束标签会在小于符号（<）后多一个整斜线/。 大多数 HTML 元素由一个起始标签和一个结束标签标识。起始标签和结束标签应成对出现，也就是说在一个起始标签之后应有其对应的结束标签。 由连个标签（起始标签和结束标签）构成的元素而言，缺少其中之一，这个元素都肯能会被认为是无效的。

注释：开始标签常被称为开放标签（opening tag），结束标签常称为闭合标签（closing tag)。

HTML5之前的浏览器在遇到没有正确嵌套的标签时，它们的解析方式是不一样的，所以显示结果也不一样。现代浏览器（）对没正确嵌套的标签已经能解析出统一的文档模型了。

有些元素没有包含文本内容或者其他元素，这称为空元素。如<img>,一般在结束标签后面空格加上个斜杠（这在XHTML是必需的）.

属性：

开始标签可能包含一些信息，这些信息叫做元素的特性，包括两部分:特性名、特性值.

一些元素可以只有特性名没有特性值。它们的特性名类似“是非”，“有或没有”，所以可以省略特性值，所以下面三种写法都是一样的意思:

<input required="required">

<input required="">

<input required>

特性值如果有包含空格就要用引号，单双引号都可以。如果特性值是单个词就可以不用引号，但为了避免出错和易于识别一般会加上引号:如：

<p class=foo bar>

上面的例子就是没加引号，浏览器错误地解析成下面的:

<p class="foo" bar="">

命名字符参考：

(一般被称作实体) 用于表示在HTML中具有特定含义的字符. 举例说，HTML将“>”和“<”符号理解为标签分隔符。所以，当你想在文本中用">"符号表达“大于”的含义时，你可以使用命名字符参考来作为替代。 以下例举了四个非常重要的常用实体：

&gt; 表示大于符号">" (>)

&lt; 表示小于符号"<" (<)

&amp; 表示和符号"and"(&)

&quot; 表示引用符号" (")

其他参考：

<https://www.w3.org/TR/2011/WD-html5-20110113/named-character-references.html>

文档类型和注释：

除标签，文本内容和实体外，一个HTML文档一定需要在第一行做出文档类型断言(doctype declaration)。在现代HTML中，这句断言书写如下：

<!DOCTYPE html>

断言告诉了浏览器需要遵循W3C标准来解析HTML和CSS代码，且不必尝试去模拟90年代的IE环境。

HTML拥有一个允许我们在HTML文档添加评论的机制。评论并不会在浏览器渲染页面时一并呈现给用户，而是隐藏在源代码内。HTML评论被如下符号封闭：

<!-- This is comment text -->

所有的HTML文档都应该有一个<html>标签，<html>标签可以包含两个部分:<head>和<body>。<head>标签用于包含整个文档的一般信息，比如文档的标题（<title>标签用于包含标题），对整个文档的描述，文档的关键字等等。文档的具体内容就要放在<body>标签里了。

<a>标签用于表示链接，在浏览器（如IE,Firefox等）中查看HTML文档时，点击<a>标签括起来的内容时，通常会跳转到另一个页面。这个要跳转到的页面的地址由<a>标签的href属性指定。

通过不同的标签，HTML文档可以包含不同的内容，比如文本，链接，图片，列表，表格，表单，框架等。

HTML文档格式详细说明：

在HTML文本中，用尖括号括起来的部分称为标签。如果想在正文里使用尖括号（或者大与号小与号，总之是同一个东西），必须使用字符转义，也就是说转换字符的原有意义。<应该使用&lt;代替，>则使用&gt;，至于&符号本身,则应该使用&amp;替代（不得不说的是有很多HTML文档没有遵循这个规则，常用的浏览器也都能够分析出&到底是一个转义的开始，还是一个符号，但是这样做是不推荐的）。

嵌套的标签，最后出现的标签应该最先结束。

HTML文档里所有的空白符（空格，Tab，换行，回车）会被浏览器忽略，唯一的例外是空格，对空格的处理方式是所有连续的空格被当成一个空格。如果要显示连续的空格（比如为了缩进），应该用&nbsp;来代表空格。

常用标签介绍：

文本：最常用的标签可能是<font>了，它用于改变字体，字号，文字颜色。

<font size="6">6</font>

<font size="4">4</font>

<font color="red" size="5">红色的5</font>

<font face="黑体">黑体的字</font>

加粗，下划线，斜体字也是常用的文字效果，它们分别用<b>,<u>,<i>表示：

<b>Bold</b>

<i>italic</i>

<u>underline</u>

还有一些标签，用来指出包含的文本有特殊的意义，比如<abbr>（表示缩写），<em>（表示强调），<strong>（表示更强地强调）,<cite>（表示引用），<address>（表示地址）等等。这些标签不是为了定义显示效果而存在的，所以从浏览器里看它们可能没有任何效果，也可能不同的浏览器对这些标签的显示效果完全不同。

一篇很长的文章，如果有合适的小标题的话，就可以快速地对它的内容进行大致的了解。在HTML里，用来表示标题的标签有：<h1>,<h2>,<h3>,<h4>,<h5>,<h6>，它们分别表示一级标题，二级标题，三级标题...

图片：<hr>标签用于在页面上添加横线。可以通过指定width和color属性来控制横线的长度和颜色。

<hr width="90%" color="red" />

<img>标签用于在页面上添加图片，src属性指定图片的地址，如果无法打开src指定的图片，浏览器通常会在页面上需要显示图片的地方显示alt属性定义的文本。

<img src="http://www.w3.org/Icons/valid-xhtml10" alt="图片简介" />

链接：超级链接用<a>标签表示，href属性指定了链接到的地址。<a>标签可以包含文本，也可以包含图片。

<a href="http://deerchao.net">deerchao的个人网站</a>

<a href="http://validator.w3.org"><img src="http://www.w3.org/Icons/valid-xhtml10" alt="验证HTML" /></a>

分段与换行：由于HTML文档会忽略空白符，所以要想保证正常的分段换行的话，必须指出哪些文字是属于同一段落的，这就用到了标签<p>。如：

<p>这是第一段。</p>

<p>这是第二段。</p>

也有人不用<p>，而用<br>。<br>只表示换行，不表示段落的开始或结束，所以通常没有结束标签。如：

这是第一段。<br>

这是第二段。<br />

这是第三段。

有时，要把文档看作不同的部分组合起来的，比如一个典型的页面可能包括三个部分：页头，主体，页脚。<div>标签专门用于标明不同的部分：如：

<div>页头内容</div>

<div>主体内容</div>

<div>页脚内容</div>

表格：HTML文档在浏览器里通常是从左到右，从上到下地显示的，到了窗口右边就自动换行。为了实现分栏的效果，很多人使用表格（<table>）进行页面排版（虽然HTML里提供表格的本意不是为了排版）。

<table>标签里通常会包含几个<tr>标签，<tr>代表表格里的一行。<tr>标签又会包含<td>标签，每个<td>代表一个单元格。

<tr>标签还可以被<table>里的<thead>或<tbody>或<tfoot>包含。它们分别代表表头，表正文，表脚。在打印网页的时候，如果表格很大，一页打印不完，<thead>和<tfoot>将在每一页出现。

<th>和<td>非常相似，也用在<tr>里边，不同的是<th>代表这个单元格是它所在的行或列的标题。 如：

<table>

<thead>

<tr><th>时间</th><th>地点</th></tr>

</thead>

<tbody>

<tr><td>2000</td><td>悉尼</td></tr>

<tr><td>2004</td><td>雅典</td></tr>

<tr><td>2000</td><td>北京</td></tr>

</tbody>

</table>

列表：表格用于表示二维数据（行，列），一维数据则用列表表示。列表可以分为无序列表（<ul>），有序列表（<ol>）和定义列表（<dl>）。前两种列表更常见一些，都用<li>标签包含列表项目。如：

<ol>

<li>打开冰箱门</li>

<li>把大象赶进去</li>

</ol>

无序列表表示一系列类似的项目，它们之间没有先后顺序。

有序列表中各个项目间的顺序是很重要的，浏览器通常会自动给它们产生编号。

框架：框架使一个窗口里能同时显示多个文档。主框架页里面没有<body>标签，取代它的是<frameset>。

<frameset>标签的属性Rows和Cols用于指定框架集(frameset)里有多少行（列），以及每行（列）的高度（宽度）。

<frameset>标签可以包含<frame>标签，每个<frame>标签代表一个文档（src属性指定文档的地址）。

如果觉得这样的页面还不够复杂的话，还可以在<frameset>标签里包含<frameset>标签。

<frameset rows="15%,\*">

<frame src="top.html" name=title scrolling=no>

<frameset cols="20%,\*">

<frame src="left.html" name=sidebar>

<frame src="right.html" name=recipes>

</frameset>

</frameset>

# Css

定义：Cascading Style Sheets (CSS) 是一门指定文档该如何呈现给用户的语言。

文档 是信息的集合，它使用一门 标记语言 作为结构。

作用：CSS帮助您将文档信息的内容 和如何展现它的细节相分离。

优点：避免重复、更容易维护、为不同的目的，使用不同的样式而内容相同

如何工作：浏览器先将标记语言和CSS转换成DOM (文档对象模型)结构。 这时DOM 就代表了电脑内存中的相应文档，因为它已经融合了文档内容和相应的样式表。

最后浏览器把 DOM的内容展示出来。

层叠样式表优先级，从高到低依次为：网页开发者定义的样式、网页阅读者定义的样式、浏览器的默认样式。

选择器（Selectors）：花括号中的部分称为声明（declaration）。同一个声明中的 属性和值组成一个名值对（property-value pairs），名值对用分号分隔.

1. 似strong的选择器称为标签选择器（tag selector）.CSS规范中称之为类型选择器（type selector）;
2. 类选择器（Class selectors），通过设置元素的 class 属性，可以为元素指定类名。类名由开发者自己指定。 文档中的多个元素可以拥有同一个类名。在写样式表时，类选择器是以英文句号（.）开头的。
3. ID选择器（ID selectors），通过设置元素的 id 属性为该元素制定ID。ID名由开发者指定。每个ID在文档中必须是唯一的。在写样式表时，ID选择器是以#开头的。（id 属性值 principal必须在文档中是唯一的；但文档中的其他标签可以有和p相同的 class 属性值 key.）【CSS规定拥有更高确定度的选择器优先级更高。ID选择器比类选择器更具确定度, 而类选择器比标签选择器（tag selector）更具确定度。且它们具有相同的确定度。那么，后出现的规则优先级高。】
4. 伪类选择器（Pseudo-classes selectors）是加在选择器后面的用来指定元素状态的关键字。伪类和伪元素（pseudo-elements）不仅可以让你为符合某种文档树结构的元素指定样式，还可以为符合某些外部条件的元素指定样式：浏览历史(比如是否访问过 (:visited)， 内容状态(如 :checked ), 鼠标位置 (如:hover). 完整列表参见：<https://www.w3.org/TR/selectors/#selectors>

语法：selector:pseudo-class {

property: value;

}

伪类列表

:link

:visited

:active

:hover

:focus

:first-child

:nth-child

:nth-last-child

:nth-of-type

:first-of-type

:last-of-type

:empty

:target

:checked

:enabled

:disabled



你可以任意组合以表达更复杂的关系。

你还可以使用星号（\*）来表示”任意元素“。

注释:CSS注释以/\* 开始，以 \*/结束。

选取器组: 当很多元素具有相同的样式时，你就需要定义一个选择器组，组内用逗号分隔。也可以单独对这些选择器重新设置样式.

如：h1, h2, h3 {color: navy;}

字体：你无法预料到用户是否可以访问样式表里定义的字体。所以在设置字体时，在属性后指定一个替代的字体列表是个不错的主意。

在这个字体列表的最后加上系统字体中的一个，如：serif,sans-serif,cursive,fantasy或monospace。如果字体不支持样式表里设置的字体特征，浏览器会使用另一种字体。比如，样式表中包含字体不支持的特殊字符，如果浏览器发现另一种字体支持这些特殊字符，那浏览器就会选择使用这种字体。

使用 font-family 属性指定文本的字体。

简体中文的字体示例：

Windows：font-family:微软雅黑;

Mac OS：font-family:"Songti SC";

字号：浏览器用户浏览页面时，可以覆盖页面默认的文号大小，也可以改变页面的字号大小。所以说尽可能的使用相对的字号大小对你来说是有意义的。

你可使用系统内置的值来设置字号，比如small,medium和large。你也可以使用相对父元素字号大小的值来设置，比如：smaller,larger,150%或1.5em。1“em”等于1个字母“m”的宽度（相对于父元素字号大小）;因此1.5em就是1.5倍的父元素字号大小。

如果有必要你也可以指定一个实际的大小，比如14px(14像素)应用于显示设备或14pt(14点)应用于打印设备。但是实际大小不能应用于视力受损用户的设备上，因为这些设备不支持指定实际的值。一个比较容易实现的策略是给顶级的文档元素指定一个系统内置的值如medium，然后再给它的子元素设置个相对值。

使用font-size 属性指定字体的大小。

行高：用来指定行与行之间的距离。如果你的文档中有一个很长的段落由很多行组成，而且这个段落的字号还比较小，这时给它指定一个稍大的间距，这样阅读起来会更方便。

使用 line-height 属性指定文本的行间距。

装饰：单独的 text-decoration就可以为文本指定其他风格，比如underline或line-through。你也可以把值设置成none，把这些风格取消掉。

其他属性：使用font-style: italic;指定文本为斜体;

使用 font-weight: bold;指定文本加粗;

使用 font-variant: small-caps;指定文本为小型大写字母;

如果我们想单独设置某个效果失效，我们可以把其相应的属性设置为normal或inherit.

参考链接：字体 <http://www.w3.org/TR/CSS21/fonts.html>

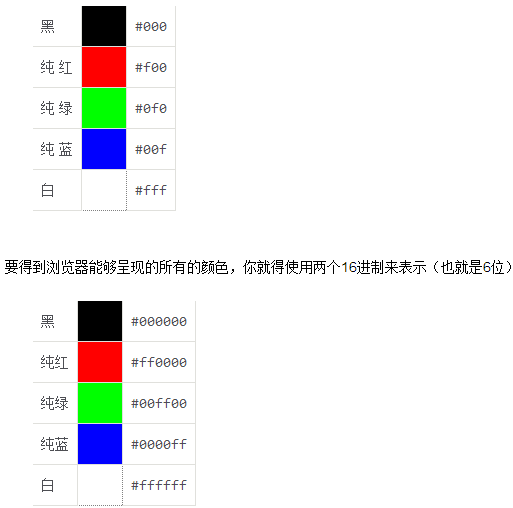
文本修饰 <http://www.w3.org/TR/CSS21/text.html>

颜色：CSS2支持17种名字的颜色。



Css3颜色参考：<https://www.w3.org/TR/2003/CR-css3-color-20030514/#svg-color>

可以使用代表红，绿，蓝三个颜色的16进制数字来表示。16进制数字的范围0-9，a-f。其中a-f代表的数值就是10-15：



以使用background-color 属性来改变元素的背景色.；背景色可以设置 transparent 属性来移除掉所有的颜色，呈现出父元素的背景色。

文本内容：CSS可以在元素的前后插入文本：在选择器的后面加上:before 或者 :after 。在声明中，指定 content 属性，并设置文本内容。如：

.ref:before {

font-weight: bold;

color: navy;

content: "Reference: ";

}

样式表默认使用UTF-8字符集。也可以通过link属性或样式表以及其他方式指定。还可以通过转义机制（通过反斜杠转义）指定单个字符。比如， \265B 是国际象棋黑皇后的符号 ♛。

图片内容：可以通过将content 属性值设置为某个图片的URL，可以将图片插到元素的前面或后面。下面这条规则在所有类名包含glossary的a标签后面插入一个空格和一个图标：

a.glossary:after {content: " " url("../images/glossary-icon.gif");}

#sidebar-box {background: url("../images/sidebar-ground.png") no-repeat;}

背景图默认是 repeat（重复）的，无需明确指出。图片在水平和垂直方向重复.

列表: 使用list-style 属性来指定列表项标记的样式。你的CSS中的选择器可以选中列表项 (比如, <li>)。也可以选中列表项的父节点 (比如, <ul>)。无序列表的每个列表项都用同样的方式标记。CSS 有三种标记样式：disc/circle/square

有序列表:在有序列表中，每个列表项都被标记了不同的序号。比如<ol> 元素，用list-style 属性指定标记样式：decimal/lower-roman/upper-roman/lower-latin/upper-latin

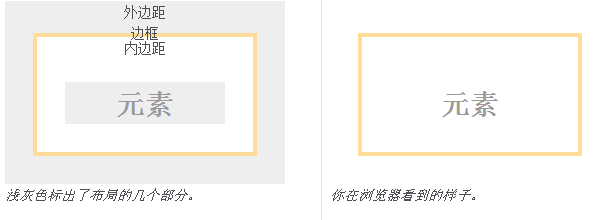
计数器：在计数开始前的某个元素上，设置 counter-reset属性以重置计数器。被计数元素的父节点是一个不错的选择。当然，任何出现在被计数元素前面的元素都可以。

设置每个需要计数的元素的counter-increment 属性为你的计数器名。

通过为选择器增加 :before 或 :after 并设置 content 属性来显示计数器。在content属性的值中设置 counter()，在括号内填上计数器的名字。可选的是设置计数器类型。其类型和前面一节 有序列表 中相同。

正常情况下，显示计数器的元素也会递增计数器。

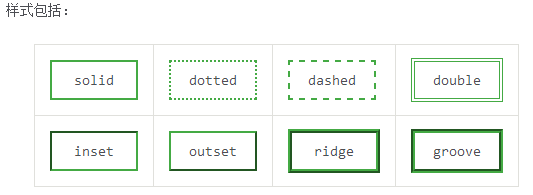
盒模型：当你的浏览器展现一个元素时，这个元素会占据一定的空间。这个空间由四部分组成。中间是元素呈现内容的区域。这个区域的外面是内边距。再外面是边框。最外面的是外边距，外边距将该元素与其它元素分开。



内边距，边框和外边距在元素的上、右、下、左都可以有不同的大小。所有这些大小值都可以为0。

颜色：内边距总是跟元素的背景色一样，所以当你设置背景色时，你会发现背景色在元素本身和内边距上都生效了。外边距总是透明的。

边框：用 border 属性给元素四周指定统一的边框。在属性值中指定边框的宽度（通常是以显示到屏幕上的像素为单位）， 样式， 还有颜色。



你也可以通过设置样式为 none 或 hidden 来明确地移除边框，或者设置边框颜色为 transparent 来让边框不可见，后者不会改变布局。

如果一次只指定某一个方向的边框，就用属性： border-top， border-right， border-bottom， border-left。 你可以用这些属性指定某个方向上的边框，或者不同方向上的不同边框。

外边距和内边距：用 margin 属性或者 padding 属性分别设置外边距和内边距的宽度

如果你指定一个宽度，它将会作用于元素四周（上、右、下、左）。

如果你指定两个宽度， 第一个宽度会作用于顶部和底部，第二个宽度作用于右边和左边。

你也可以按照顺序指定四个宽度： 上、右、下、左。如：h3 {border-top: 1px solid gray;}

文档结构：页面标记语言通常都会有公共标签来创建结构。例如, 在 HTML 中你可以使用 <div> 元素来创建结构。

大小单位：你可以通过像素来指定大小。这在有些情况下是非常合适的，比如电脑屏幕显示。 但当用户改变字体大小之后，你的布局可能会发生错位。因此，最好通过百分比或 ems (em) 来指定大小。 em 通常是指当前字体大小(字母m的宽度)。当用户改变字体大小时，你的布局会自己修正。

文本左边的border通过像素来指定大小。

文本右边的border通过 ems来指定大小。

在你的浏览器中，修改字体大小，会发现右边的border会自己修正大小而左边的不会。

文本布局: 有两个属性可以指定元素内容的对齐方式。

text-align:内容对齐。 可以使用下面几个值： left, right, center, justify。

text-indent:指定内容缩进。

这两个属性可以应用于任何文本类内容，不只是纯文本。 需要注意的是，它们会被元素的子元素继承， 所以需要在子元素中将它们关闭，以免出现意想不到的效果。

在 HTML 文档中, 标题之后的内容并不属于标题。当你对齐一个标题时，其后的元素不会继承该样式。

浮动: float 属性强制元素靠左或靠右。 这是控制元素位置和大小的简单方法。可以使用 clear 属性来避免其它元素受到浮动效果的影响。如：

ul, #numbered {float: left;}

h3 {clear: left;}

(box右侧需要增加一些padding ,防止文本与边框太近)

位置：可以为一个元素指定 position 属性为以下值之一，来设置其位置。

relative：通过为元素指定一个值，元素相对于其原来位置移动。也可以使用margin来达到同样的效果。

fixed：为元素指定相对于窗口的确切位置 。即使文档的其它元素出现滚动，元素位置仍然不变。

absolute：为元素指定相对于其父元素的确切位置。只有在父元素使用 relative, fixed or absolute 时才有效。你可以为任何父元素指定 position: relative;因为它不会产生移动。

static：默认值。当明确要关闭位置属性时使用。

和 position 属性(除了 static)一起使用的, 有下列属性: top, right, bottom, left, width, height 通过设置它们来指定元素的位置或大小。

表格结构：在表格中，信息显示在一个个的单元格（cell）中.

在页面横向上一条直线的单元格构成了行（row）。

在一些表格中，行可能被分组。表格开始的特定的行组是表头 （header）。表格最后的特定行组是表尾（footer）。表格中主要的行就是表体（body），这些表体也可能被分组。

在页面纵向上一条直线的单元格构成了列（column），但是在CSS表格中，列的使用是受限的。

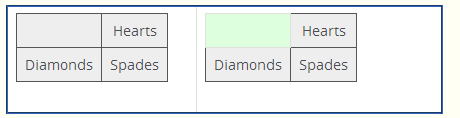
边框：单元格没有外边距。但是单元格有边框和内边距。默认情况下，边框被表格的border-spacing属性值间隔。你也可以通过设置表格的border-collapse属性值为collapse来完全移除间隔。

标题：<caption>元素是用在整个表格的一个标签。默认下，它显示在表格的顶部。

可以设置<caption>的caption-side属性值为bottom来将标签移到表格的底部。

想要样式化caption的文本，可以使用任何常规的文本属性。

空单元格：通过为表格元素指定empty-cells属性值show来显示空单元格（就是其边框和背景）。你也可以指定empty-cells: hide;来隐藏边框和背景，那么如果一个单元格的父元素设置了背景，背景将通过空单元格显示出来。



Media：CSS通过使用@media 的格式来对特定的媒介指定适配规则。在标签语言中，导航区域父元素的id是 nav-area.(在 HTML5中, 可以使用 <nav> 元素代替带有id属性的 <div>。)

为了网页在被打印的时候去掉无用的信息，我们在样式表中加一条适配规则，使导航区域在打印时是被隐藏起来的:

@media print {

#nav-area {display: none;}

}

一些常见的媒介类型:

screen 彩色计算机显示

print 打印（分页式媒体）

projection 投影

all 所有媒体 (默认)

类型可以在样式表通过link方式加到文档时被指定，这是文档的标签语言允许的 。例如，在HTML中，你可以通过在 LINK 标签上添加media属性来指定媒介类型。

在CSS中，你可以在样式表开头使用 @import 一个url来引入另外的样式表，同时指定其媒介类型。

打印：@page 规则能够设置页间距，对于双面打印，还可以分开设置 @page:left 和 @page:right。

对于打印媒介，可以使用适当的长度单位，像是英寸(in)、点（1pt = 1/72 inch）、厘米（cm）还有毫米（mm）。这等同于使用em来配合字体大小和百分比。

可以通过使用 page-break-before, page-break-after 和 page-break-inside 属性来控制文档内容的分页边界。

这个规则把四个方向的页边距都设置为1 inch:

@page {margin: 1in;}

这个规则确保每个H1元素都从新的一页开始:

h1 {page-break-before: always;}

用户界面：



outline 属性通过创建轮廓来表明获得键盘焦点。只有在父元素使用 relative, fixed or absolute 时才有效。你可以为任何父元素指定 position: relative;因为它不会产生移动。

它的作用相当于 border 属性，但与其不同的是它不能指明个别方向。

一些其他的用户界面特性通常会通过属性来应用。例如，禁用或者只读的元素可以设置 disabled 属性和 readonly 属性。选择器可以通过方括: [disabled] 或者 [readonly]来指定这些属性。

display 是CSS中最重要的用于控制布局的属性。每个元素都有一个默认的 display 值，这与元素的类型有关。对于大多数元素它们的默认值通常是 block 或 inline 。一个 block 元素通常被叫做块级元素。一个 inline 元素通常被叫做行内元素。

block:

div 是一个标准的块级元素。一个块级元素会新开始一行并且尽可能撑满容器。其他常用的块级元素包括 p 、 form 和HTML5中的新元素： header 、 footer 、 section 等等。

inline:

span 是一个标准的行内元素。一个行内元素可以在段落中 <span> 像这样 </span> 包裹一些文字而不会打乱段落的布局。 a 元素是最常用的行内元素，它可以被用作链接。

none:

一些特殊元素的默认 display 值是它，例如 script 。 display:none 通常被 JavaScript 用来在不删除元素的情况下隐藏或显示元素。

它和 visibility 属性不一样。把 display 设置成 none 不会保留元素本该显示的空间，但是 visibility: hidden; 还会保留。

margin: auto;

设置块级元素的 width 可以阻止它从左到右撑满容器。然后你就可以设置左右外边距为 auto 来使其水平居中。元素会占据你所指定的宽度，然后剩余的宽度会一分为二成为左右外边距。

唯一的问题是，当浏览器窗口比元素的宽度还要窄时，浏览器会显示一个水平滚动条来容纳页面。

#main {

width: 600px;

margin: 0 auto;

}

max-width: 在这种情况下使用 max-width 替代 width 可以使浏览器更好地处理小窗口的情况。这点在移动设备上显得尤为重要. 所有的主流浏览器包括IE7+在内都支持 max-width

#main {

max-width: 600px;

margin: 0 auto;

}

盒模型。当你设置了元素的宽度，实际展现的元素却能够超出你的设置：因为元素的边框和内边距会撑开元素。看下面的例子，两个相同宽度的元素显示的实际宽度却不一样。如：

.simple {

width: 500px;

margin: 20px auto;

}

.fancy {

width: 500px;

margin: 20px auto;

padding: 50px;

border-width: 10px;

}

box-sizing：当你设置一个元素为 box-sizing: border-box; 时，此元素的内边距和边框不再会增加它的宽度。如：

\* {

-webkit-box-sizing: border-box;

-moz-box-sizing: border-box;

box-sizing: border-box;

}

既然 box-sizing 是个很新的属性，目前你还应该像我之前在例子中那样使用 -webkit- 和 -moz- 前缀。这可以启用特定浏览器实验中的特性。同时记住它是支持IE8+。

position的属性：

static 是默认值。任意 position: static; 的元素不会被特殊的定位。一个 static 元素表示它不会被“positioned”，一个 position 属性被设置为其他值的元素表示它会被“positioned”。

.static {

position: static;

}

relative 表现的和 static 一样，除非你添加了一些额外的属性。

在一个相对定位（position属性的值为relative）的元素上设置 top 、 right 、 bottom 和 left 属性会使其偏离其正常位置。其他的元素则不会调整位置来弥补它偏离后剩下的空隙。如：

.relative1 {

position: relative;

}

.relative2 {

position: relative;

top: -20px;

left: 20px;

background-color: white;

width: 500px;

}

fixed：一个固定定位（position属性的值为fixed）元素会相对于视窗来定位，这意味着即便页面滚动，它还是会停留在相同的位置。和 relative 一样， top 、 right 、 bottom 和 left 属性都可用。如：

.fixed {

position: fixed;

bottom: 0;

right: 0;

width: 200px;

background-color: white;

}

一个固定定位元素不会保留它原本在页面应有的空隙。

令人惊讶地是移动浏览器对 fixed 的支持很差。

absolute：absolute 是最棘手的position值。 absolute 与 fixed 的表现类似，除了它不是相对于视窗而是相对于最近的“positioned”祖先元素。如果绝对定位（position属性的值为absolute）的元素没有“positioned”祖先元素，那么它是相对于文档的 body 元素，并且它会随着页面滚动而移动。记住一个“positioned”元素是指p osition 值不是 static 的元素。

如：

.relative {

position: relative;

width: 600px;

height: 400px;

}

.absolute {

position: absolute;

top: 120px;

right: 0;

width: 300px;

height: 200px;

}

float: 可用于实现文字环绕图片.如：

img {

float: right;

margin: 0 0 1em 1em;

}

clear 属性被用于控制浮动。

清除浮动（clearfix hack）：overflow，如：

.clearfix {

overflow: auto;

}

这个可以在现代浏览器上工作。如果你想要支持IE6，你就需要再加入如下样式：

.clearfix {

overflow: auto;

zoom: 1;

}

有些独特的浏览器需要“额外的关照”。

百分比宽度：百分比是一种相对于包含块的计量单位。它对图片很有用：如下我们实现了图片宽度始终是容器宽度的50%。把页面缩小看下效果！如：

article img {

float: right;

width: 50%;

}

甚至还能同时使用 min-width 和 max-width 来限制图片的最大或最小宽度！

百分比宽度布局：当布局很窄时， nav 就会被挤扁。更糟糕的是，你不能在 nav 上使用 min-width 来修复这个问题，因为右边的那列是不会遵守它的。

媒体查询：

“响应式设计（Responsive Design）”是一种让网站针对不同的浏览器和设备“响应”不同显示效果的策略，这样可以让网站在任何情况下显示的很棒！

媒体查询是做此事所需的最强大的工具。让我们使用百分比宽度来布局，然后在浏览器变窄到无法容纳侧边栏中的菜单时，把布局显示成一列：

@media screen and (min-width:600px) {

nav {

float: left;

width: 25%;

}

section {

margin-left: 25%;

}

}

@media screen and (max-width:599px) {

nav li {

display: inline;

}

}

使用 meta viewport 之后可以让你的布局在移动浏览器上显示的更好。

inline-block：铺满浏览器

1. 浮动（困难）

.box {

float: left;

width: 200px;

height: 100px;

margin: 1em;

}

.after-box {

clear: left;

}

1. 用 display 属性的值 inline-block 来实现相同效果。

.box2 {

display: inline-block;

width: 200px;

height: 100px;

margin: 1em;

}

得做些额外工作来让IE6和IE7支持 inline-block 。有些时候人们谈到 inline-block 会触发叫做 hasLayout 的东西，你只需要知道那是用来支持旧浏览器的。

inline-block 来布局：

vertical-align 属性会影响到 inline-block 元素，你可能会把它的值设置为 top 。

你需要设置每一列的宽度

如果HTML源代码中元素之间有空格，那么列与列之间会产生空隙。 如：

nav {

display: inline-block;

vertical-align: top;

width: 25%;

}

.column {

display: inline-block;

vertical-align: top;

width: 75%;

}

Column：实现文字的多列布局。

.three-column {

padding: 1em;

-moz-column-count: 3;

-moz-column-gap: 1em;

-webkit-column-count: 3;

-webkit-column-gap: 1em;

column-count: 3;

column-gap: 1em;

}

CSS columns是很新的标准，所以你需要使用前缀，并且它不被IE9及以下和Opera Mini支持。

flexbox 布局模式被用来重新定义CSS中的布局方式。很遗憾的是最近规范变动过多，导致各个浏览器对它的实现也有所不同。目前只能在支持 flexbox 的 Chrome 浏览器中运行

# JavaScript

1、直接写入 HTML 输出流:

document.write("<h1>这是一个标题</h1>");

您只能在 HTML 输出中使用 document.write。如果您在文档加载后使用该方法，会覆盖整个文档。

1. 对事件的反应：

<button type="button" onclick="alert('欢迎!')">点我!</button>

alert() 函数在 JavaScript 中并不常用，但它对于代码测试非常方便。

1. 改变 HTML 内容：

x=document.getElementById("demo") //查找元素

x.innerHTML="Hello JavaScript"; //改变内容

您会经常看到 document.getElementById("some id")。这个方法是 HTML DOM 中定义的。

DOM (Document Object Model)（文档对象模型）是用于访问 HTML 元素的正式W3C 标准。

1. 改变 HTML 图像：

<script>

function changeImage()

{

element=document.getElementById('myimage')

if (element.src.match("bulbon"))

{

element.src="/images/pic\_bulboff.gif";

}

else

{

element.src="/images/pic\_bulbon.gif";

}

}

</script>

<img id="myimage" onclick="changeImage()"

src="/images/pic\_bulboff.gif" width="100" height="180">

JavaScript 能够改变任意 HTML 元素的大多数属性，而不仅仅是图片。（match是判断是否包含某字符串）

1. 改变 HTML 样式：

<p id="demo">

JavaScript 能改变 HTML 元素的样式。

</p>

<script>

function myFunction()

{

x=document.getElementById("demo") // 找到元素

x.style.color="#ff0000"; // 改变样式

}

</script>

<button type="button" onclick="myFunction()">点击这里</button>

改变 HTML 元素的样式，属于改变 HTML 属性的变种。

1. 验证输入：

if isNaN(x) {alert("不是数字")};

验证用户的输入。

JavaScript 用法：

1. HTML 中的脚本必须位于 <script> 与 </script> 标签之间。脚本可被放置在 HTML 页面的 <body> 和 <head> 部分中。
2. <script> 标签：<script> 和 </script> 会告诉 JavaScript 在何处开始和结束。<script> 和 </script> 之间的代码行包含了 JavaScript:
3. 在 <head> 或者 <body> 的JavaScript：您可以在 HTML 文档中放入不限数量的脚本。脚本可位于 HTML 的 <body> 或 <head> 部分中，或者同时存在于两个部分中。通常的做法是把函数放入 <head> 部分中，或者放在页面底部。这样就可以把它们安置到同一处位置，不会干扰页面的内容。
4. 外部的 JavaScript：把脚本保存到外部文件中。外部文件通常包含被多个网页使用的代码。外部 JavaScript 文件的文件扩展名是 .js。如需使用外部文件，请在 <script> 标签的 "src" 属性中设置该 .js 文件。

Eg: <script src="myScript.js"></script>

JavaScript 输出:

1、