

HTML 标签梳理

标签	描述
<!--...-->	定义注释
<!doctype>	定义文档类型
<a>	定义超文本链接
<abbr>	定义缩写
<acronym>	定义只取首字母的缩写，不支持 HTML5
<address>	定义文档作者或拥有者的联系信息
<applet>	HTML5 中不赞成使用定义嵌入的 applet
<area>	定义图像映射内部的区域
<article>	定义一个文章区域（HTML5 新标签）
<aside>	定义页面的侧边栏内容（HTML5 新标签）
<audio>	定义音频内容（HTML5 新标签）
	定义文本粗体
<base>	定义页面中所有链接的默认地址或默认目标
<bdi>	允许您设置一段文本，使其脱离其父元素的文本方向设置（HTML5 新标签）
<bdo>	定义文字方向
<big>	定义大号文本，HTML5 不支持
<blockquote>	定义长的引用
<body>	定义文档的主体
 	定义换行
<button>	定义一个点击按钮
<canvas>	定义图形，比如图表和其他图像，标签只是图形容器，您必须使用脚本来绘制图形（HTML5 新标签）
<caption>	定义表格标题
<center>	HTML5 不支持，不赞成使用定义居中文本

<cite>	定义引用(citation)
<code>	定义计算机代码文本
<col>	定义表格中一个或多个列的属性值
<colgroup>	定义表格中供格式化的列组
<command>	定义命令按钮，比如单选按钮、复选框或按钮（HTML5 新标签）
<datalist>	定义选项列表请与 input 元素配合使用该元素，来定义 input 可能的值（HTML5 新标签）
<dd>	定义定义列表中项目的描述
	定义被删除文本
<details>	用于描述文档或文档某个部分的细节（HTML5 新标签）
<dfn>	定义定义项目
<dialog>	定义对话框，比如提示框（HTML5 新标签）
<dir>	HTML5 不支持，不赞成使用定义目录列表
<div>	定义文档中的节
<dl>	定义列表详情
<dt>	定义列表中的项目
	定义强调文本
<embed>	定义嵌入的内容，比如插件（HTML5 新标签）
<fieldset>	定义围绕表单中元素的边框
<figcaption>	定义<figure>元素的标题（HTML5 新标签）
<figure>	规定独立的流内容（图像、图表、照片、代码等等）（HTML5 新标签）
	HTML5 不支持，不赞成使用定义文字的字体、尺寸和颜色（HTML5 新标签）
<footer>	定义 section 或 document 的页脚（HTML5 新标签）
<form>	定义了 HTML 文档的表单
<frame>	定义框架集的窗口或框架
<frameset>	定义框架集
<h1> to <h6>	定义 HTML 标题
<head>	定义关于文档的信息

<header>	定义了文档的头部区域（HTML5 新标签）
<hr>	定义水平线
<html>	定义 HTML 文档
<i>	定义斜体字
<iframe>	定义内联框架
	定义图像
<input>	定义输入控件
<ins>	定义被插入文本
<kbd>	定义键盘文本
<keygen>	规定用于表单的密钥对生成器字段（HTML5 新标签）
<label>	定义 input 元素的标注
<legend>	定义 fieldset 元素的标题
	定义列表的项目
<link>	定义文档与外部资源的关系
<map>	定义图像映射
<mark>	定义带有记号的文本请在需要突出显示文本时使用<m>标签（HTML5 新标签）
<menu>	不赞成使用定义菜单列表
<meta>	定义关于 HTML 文档的元信息
<meter>	定义度量衡仅用于已知最大和最小值的度量（HTML5 新标签）
<nav>	定义导航链接的部分（HTML5 新标签）
<noframes>	定义针对不支持框架的用户的替代内容 HTML5 不支持
<noscript>	定义针对不支持客户端脚本的用户的替代内容
<object>	定义内嵌对象
	定义有序列表
<optgroup>	定义选择列表中相关选项的组合
<option>	定义选择列表中的选项
<output>	定义不同类型的输出，比如脚本的输出（HTML5 新标签）
<p>	定义段落

<param>	定义对象的参数
<pre>	定义预格式文本
<progress>	定义运行中的进度（进程）（HTML5 新标签）
<q>	定义短的引用
<rp>	<rp>标签在 ruby 注释中使用，以定义不支持 ruby 元素的浏览器所显示的内容（HTML5 新标签）
<rt>	<rt>标签定义字符（中文注音或字符）的解释或发音（HTML5 新标签）
<ruby>	<ruby>标签定义 ruby 注释（中文注音或字符）（HTML5 新标签）
<s>	不赞成使用定义加删除线的文本
<samp>	定义计算机代码样本
<script>	定义客户端脚本
<section>	<section>标签定义文档中的节（section、区段）比如章节、页眉、页脚或文档中的其他部分（HTML5 新标签）
<select>	定义选择列表（下拉列表）
<small>	定义小号文本
<source>	<source>标签为媒介元素（比如<video>和<audio>）定义媒介资源（HTML5 新标签）
	定义文档中的节
<strike>	HTML5 不支持，不赞成使用定义加删除线文本
	定义强调文本
<style>	定义文档的样式信息
<sub>	定义下标文本
<summary>	<summary>标签包含 details 元素的标题，"details"元素用于描述有关文档或文档片段的详细信息（HTML5 新标签）
<sup>	定义上标文本
<table>	定义表格
<tbody>	定义表格中的主体内容
<td>	定义表格中的单元
<textarea>	定义多行的文本输入控件

<tfoot>	定义表格中的表注内容（脚注）
<th>	定义表格中的表头单元格
<thead>	定义表格中的表头内容
<time>	定义日期或时间，或者两者（HTML5 新标签）
<title>	定义文档的标题
<tr>	定义表格中的行
<track>	<track>标签为诸如 video 元素之类的媒介规定外部文本轨道（HTML5 新标签）
<tt>	定义打字机文本
<u>	不赞成使用定义下划线文本
	定义无序列表
<var>	定义文本的变量部分
<video>	<video>标签定义视频，比如电影片段或其他视频流（HTML5 新标签）
<wbr>	规定在文本中的何处适合添加换行符（HTML5 新标签）

表 1 HTML 参考手册(HTML5 标准)（按字母顺序排列）

WXML 中的标签

view

属性	类型	默认值	必填	说明	最低版本
hover-class	string	none	否	指定按下去的样式类。当 hover-class="none"时，没有点击态效果	1.0.0
hover-stop-propagation	boolean	false	否	指定是否阻止本节点的祖先节点出现点击态	1.5.0
hover-start-time	number	50	否	按住后多久出现点击态，单位毫秒	1.0.0
hover-stay-time	number	400	否	手指松开后点击态保留时间，单位毫秒	1.0.0

表 2 view 视图容器

注：如果需要使用滚动视图，使用 scroll-view。

text

属性	类型	默认值	必填	说明	最低版本
selectable	boolean	false	否	文本是否可选	1.1.0
space	string		否	显示连续空格	1.4.0
decode	boolean	false	否	是否解码	1.4.0

表 3 text 文本

其中，space 的合法值有，ensp：中文字符空格一半大小；emsp：中文字符空格大小；nbsp：根据字体设置的空格大小。

注：

- (1) decode 可以解析的有 < > & '    
- (2) 各个操作系统的空格标准并不一致；
- (3) text 组件内只支持 text 嵌套；
- (4) 除了文本节点以外的其他节点都无法长按选中；
- (5) 基础库版本低于 2.1.0 时，text 组件内嵌的 text-style 设置可能不会生效。

CSS 基本样式梳理

字体属性：(font)

大小 {font-size: x-large;} (特大) xx-small; (极小)，单位：px、pd

样式 {font-style: oblique;} (偏斜体) italic; (斜体) normal; (正常)

行高 {line-height: normal;} (正常) 单位：px、pd

粗细 {font-weight: bold;} (粗体) lighter; (细体) normal; (正常)

变体 {font-variant: small-caps;} (小型大写字母) normal; (正常)

大小写 {text-transform: capitalize;} uppercase; (大写) lowercase; (小写) none; (无)

修饰 {text-decoration: underline;} (下划线) overline; (上划线) line-through; (删除)

线) blink;(闪烁)

常用字体(font-family) "Courier New", "Times New Roman", Times, serif

背景属性: (background)

色彩 {background-color: #FFFFFF;

图片 {background-image: url();}

重复 {background-repeat: no-repeat;}

滚动 {background-attachment: fixed;}(固定) scroll;(滚动)

位置 {background-position: left;}(水平) top(垂直);

简写方法 {background:#000 url(..) repeat fixed left top;}

方框属性: (box)

width::; height::; float::; clear:both; margin::; padding::; 顺序: 上右下左

边框属性: (border)

border-style: dotted;(点线) dashed;(虚线) solid; double;(双线) groove;(槽线)
ridge;(脊状) inset;(凹陷) outset;

border-width::; 边框宽度

border-color::#;

简写方法 border: width style color;

列表属性: (list-style)

类型 list-style-type: disc;(圆点) circle;(圆圈) square;(方块) decimal;(数字)
lower-roman;(小罗马数字) upper-roman; lower-alpha; upper-alpha;

位置 list-style-position: outside;(外) inside;

图像 list-style-image: url(..);

定位属性: (Position)Position: absolute; relative; static;

visibility: inherit; visible; hidden;

overflow: visible; hidden; scroll; auto;

clip: rect(12px,auto,12px,auto)

CSS 盒子模型

CSS 盒子模型(Box Model)，所有 HTML 元素可以看作盒子，在 CSS 中，"box model"这一术语是用来设计和布局时使用。

CSS 盒模型本质上是一个盒子，封装周围的 HTML 元素，它包括：边距，边框，填充，和实际内容。

盒模型允许我们在其它元素和周围元素边框之间的空间放置元素。

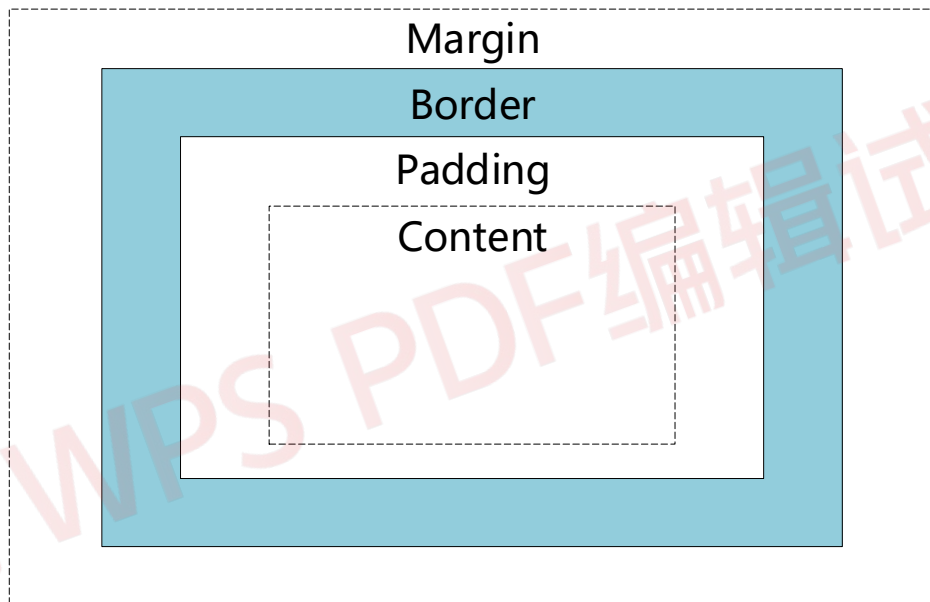


图 1 盒子模型(Box Model)

- 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。

为了正确设置元素在所有浏览器中的宽度和高度，需要知道的盒模型是如何工作的。

元素的宽度和高度：

当指定一个 CSS 元素的宽度和高度属性时，只是设置内容区域的宽度和高

度。要知道，完全大小的元素，还必须添加填充，边框和边距。.

最终元素的总宽度计算公式是这样的：

总元素的宽度=宽度+左填充+右填充+左边框+右边框+左边距+右边距

元素的总高度最终计算公式是这样的：

总元素的高度=高度+顶部填充+底部填充+上边框+下边框+上边距+下边距

浏览器的兼容性问题：

一旦为页面设置了恰当的 DTD，大多数浏览器都会按照上面的图示来呈现内容。然而 IE5 和 6 的呈现却是不正确的。根据 W3C 的规范，元素内容占据的空间是由 width 属性设置的，而内容周围的 padding 和 border 值是另外计算的。不幸的是，IE5.X 和 6 在怪异模式中使用自己的非标准模型。这些浏览器的 width 属性不是内容的宽度，而是内容、内边距和边框的宽度的总和。

虽然有方法解决这个问题。但是目前最好的解决方案是回避这个问题。也就是，不要给元素添加具有指定宽度的内边距，而是尝试将内边距或外边距添加到元素的父元素和子元素。

IE8 及更早 IE 版本不支持设置填充的宽度和边框的宽度属性。

解决 IE8 及更早版本不兼容问题可以在 HTML 页面声明<!DOCTYPE html> 即可。

CSS 定位

position 属性指定了元素的定位类型。

position 属性的五个值：

- static
- relative
- fixed
- absolute
- sticky

元素可以使用的顶部，底部，左侧和右侧属性定位。然而，这些属性无法工作，除非是先设定 position 属性。他们也有不同的工作方式，这取决于定位方法。

static 定位

HTML 元素的默认值，即没有定位，遵循正常的文档流对象。

静态定位的元素不会受到 `top`, `bottom`, `left`, `right` 影响。

fixed 定位

元素的位置相对于浏览器窗口是固定位置。

即使窗口是滚动的它也不会移动：

注意：Fixed 定位在 IE7 和 IE8 下需要描述!DOCTYPE 才能支持。Fixed 定位使元素的位置与文档流无关，因此不占据空间。Fixed 定位的元素和其他元素重叠。

relative 定位

相对定位元素的定位是相对其正常位置。

移动相对定位元素，但它原本所占的空间不会改变。

相对定位元素经常被用来作为绝对定位元素的容器块。

absolute 定位

绝对定位的元素的位置相对于最近的已定位父元素，如果元素没有已定位的父元素，那么它的位置相对于<html>。

absolute 定位使元素的位置与文档流无关，因此不占据空间。

absolute 定位的元素和其他元素重叠。

sticky 定位

sticky 英文字面意思是粘，粘贴，所以可以把它称之为粘性定位。

position: sticky; 基于用户的滚动位置来定位。

粘性定位的元素是依赖于用户的滚动，在 position: relative 与 position: fixed 定位之间切换。

它的行为就像 position: relative; 而当页面滚动超出目标区域时，它的表现就像

`position:fixed;`，它会固定在目标位置。

元素定位表现为在跨越特定阈值前为相对定位，之后为固定定位。

这个特定阈值指的是 `top`, `right`, `bottom` 或 `left` 之一，换言之，指定 `top`, `right`, `bottom` 或 `left` 四个阈值其中之一，才可使粘性定位生效。否则其行为与相对定位相同。

注意：Internet Explorer, Edge 15 及更早 IE 版本不支持 `sticky` 定位。Safari 需要使用 `-webkit-` prefix。

重叠的元素

元素的定位与文档流无关，所以它们可以覆盖页面上的其它元素

`z-index` 属性指定了一个元素的堆叠顺序（哪个元素应该放在前面，或后面）

一个元素可以有正数或负数的堆叠顺序：

具有更高堆叠顺序的元素总是在较低的堆叠顺序元素的前面。

注意：如果两个定位元素重叠，没有指定 `z-index`，最后定位在 HTML 代码中的元素将被显示在最前面。

所有的 CSS 定位属性

属性	说明
<code>bottom</code>	定义了定位元素下外边距边界与其包含块下边界之间的偏移
<code>clip</code>	剪辑一个绝对定位的元素
<code>cursor</code>	显示光标移动到指定的类型
<code>left</code>	定义了定位元素左外边距边界与其包含块左边界之间的偏移
<code>overflow</code>	设置当元素的内容溢出其区域时发生的事情
<code>overflow-y</code>	指定如何处理顶部/底部边缘的内容溢出元素的内容区域
<code>overflow-x</code>	指定如何处理右边/左边边缘的内容溢出元素的内容区域
<code>position</code>	指定元素的定位类型
<code>right</code>	定义了定位元素右外边距边界与其包含块右边界之间的偏移
<code>top</code>	定义了一个定位元素的上外边距边界与其包含块上边界之间的偏移
<code>z-index</code>	设置元素的堆叠顺序

表 5 CSS 定位属性表

relative 和 absolute 的区别

relative: 定位是相对于自身位置定位（设置偏移量的时候，会相对于自身所在的位置偏移）。设置了 **relative** 的元素仍然处在文档流中，元素的宽高不变，设置偏移量也不会影响其他元素的位置。最外层容器设置为 **relative** 定位，在没有设置宽度的情况下，宽度是整个浏览器的宽度。

absolute: 定位是相对于离元素最近的设置了绝对或相对定位的父元素决定的，如果没有父元素设置绝对或相对定位，则元素相对于根元素即 **html** 元素定位。设置了 **absolute** 的元素脱离了文档流，元素在没有设置宽度的情况下，宽度由元素里面的内容决定。脱离后原来的位置相当于是空的，下面的元素会来占据位置。

CSS 浮动

CSS 的 **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。**Float**（浮动），往往是用于图像，但它在布局时一样非常有用。

元素怎样浮动

元素的水平方向浮动，意味着元素只能左右移动而不能上下移动。

一个浮动元素会尽量向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。

浮动元素之后的元素将围绕它。

浮动元素之前的元素将不会受到影响。

如果图像是右浮动，（**float:right;**）的文本流将环绕在它左边：

彼此相邻的浮动元素

如果你把几个浮动的元素放到一起，如果有空间的话，它们将彼此相邻。

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。

CSS 中所有的浮动属性

属性	描述
<code>clear</code>	指定不允许元素周围有浮动元素
<code>float</code>	指定一个盒子（元素）是否可以浮动

表 6 浮动属性表

CSS 弹性布局

弹性盒子即弹性布局，是 CSS3 的一种新的布局模式。

CSS3 弹性盒（Flexible Box 或 flexbox），是一种当页面需要适应不同的屏幕大小以及设备类型时确保元素拥有恰当的行为的布局方式。

引入弹性盒布局模型的目的是提供一种更加有效的方式来对一个容器中的子元素进行排列、对齐和分配空白空间。

CSS3 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。

注意：弹性容器外及弹性子元素内是正常渲染的。弹性盒子只定义了弹性子元素如何在弹性容器内布局。

弹性子元素通常在弹性盒子内一行显示。默认情况每个容器只有一行。

flex-direction 属性

flex-direction 属性指定了弹性子元素在父容器中的位置。

语法

flex-direction: row | row-reverse | column | column-reverse

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。

justify-content 语法如下:

justify-content: flex-start | flex-end | center | space-between | space-around

各个值解析:

- **flex-start**:
弹性项目向行头紧挨着填充。这个是默认值。第一个弹性项的 **main-start** 外边距边线被放置在该行的 **main-start** 边线，而后续弹性项依次平齐摆放。
- **flex-end**:
弹性项目向行尾紧挨着填充。第一个弹性项的 **main-end** 外边距边线被放置在该行的 **main-end** 边线，而后续弹性项依次平齐摆放。
- **center**:
弹性项目居中紧挨着填充。（如果剩余的自由空间是负的，则弹性项目将在两个方向上同时溢出）。
- **space-between**:

弹性项目平均分布在该行上。如果剩余空间为负或者只有一个弹性项，则该值等同于 **flex-start**。否则，第 1 个弹性项的外边距和行的 **main-start** 边线对齐，而最后 1 个弹性项的外边距和行的 **main-end** 边线对齐，然后剩余的弹性项分布在该行上，相邻项目的间隔相等。

- **space-around:**

弹性项目平均分布在该行上，两边留有一半的间隔空间。如果剩余空间为负或者只有一个弹性项，则该值等同于 **center**。否则，弹性项目沿该行分布，且彼此间隔相等（比如是 **20px**），同时首尾两边和弹性容器之间留有一半的间隔（ $1/2 * 20px = 10px$ ）。

align-items 属性

align-items 设置或检索弹性盒子元素在侧轴（纵轴）方向上的对齐方式。

语法

align-items: **flex-start** | **flex-end** | **center** | **baseline** | **stretch**

各个值解析：

- **flex-start:** 弹性盒子元素的侧轴（纵轴）起始位置的边界紧靠住该行的侧轴起始边界。
- **flex-end:** 弹性盒子元素的侧轴（纵轴）起始位置的边界紧靠住该行的侧轴结束边界。
- **center:** 弹性盒子元素在该行的侧轴（纵轴）上居中放置。（如果该行的尺寸小于弹性盒子元素的尺寸，则会向两个方向溢出相同的长度）。
- **baseline:** 如弹性盒子元素的行内轴与侧轴为同一条，则该值与 '**flex-start**' 等效。其它情况下，该值将参与基线对齐。
- **stretch:** 如果指定侧轴大小的属性值为 '**auto**'，则其值会使项目的边距盒的尺寸尽可能接近所在行的尺寸，但同时会遵照 '**min/max-width/height**' 属性的限制。

flex-wrap 属性

flex-wrap 属性用于指定弹性盒子的子元素换行方式。

语法

`flex-wrap: nowrap | wrap | wrap-reverse | initial | inherit;`

各个值解析:

- `nowrap` - 默认, 弹性容器为单行。该情况下弹性子项可能会溢出容器。
- `wrap` - 弹性容器为多行。该情况下弹性子项溢出的部分会被放置到新行, 子项内部会发生断行
- `wrap-reverse` - 反转 `wrap` 排列。

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。

语法

`align-content: flex-start | flex-end | center | space-between | space-around | stretch`

各个值解析:

- `stretch` - 默认。各行将会伸展以占用剩余的空间。
- `flex-start` - 各行向弹性盒容器的起始位置堆叠。
- `flex-end` - 各行向弹性盒容器的结束位置堆叠。
- `center` - 各行向弹性盒容器的中间位置堆叠。
- `space-between` - 各行在弹性盒容器中平均分布。
- `space-around` - 各行在弹性盒容器中平均分布, 两端保留子元素与子元素之间间距大小的一半。

弹性子元素属性

排序

语法

`order:`

各个值解析:

- `<integer>`: 用整数来定义排列顺序, 数值小的排在前面。可以为负值。

对齐

设置"margin"值为"auto"值，自动获取弹性容器中剩余的空间。所以设置垂直方向 margin 值为"auto"，可以使弹性子元素在弹性容器的两上轴方向都完全居中。

完美的居中

使用弹性盒子，居中变的很简单，只想要设置 `margin: auto;` 可以使得弹性子元素在两上轴方向上完全居中。

align-self 属性

align-self 属性用于设置弹性元素自身在侧轴（纵轴）方向上的对齐方式。

语法

align-self: auto | flex-start | flex-end | center | baseline | stretch

各个值解析：

- **auto**：如果'align-self'的值为'auto'，则其计算值为元素的父元素的'align-items'值，如果其没有父元素，则计算值为'stretch'。
- **flex-start**：弹性盒子元素的侧轴（纵轴）起始位置的边界紧靠住该行的侧轴起始边界。
- **flex-end**：弹性盒子元素的侧轴（纵轴）起始位置的边界紧靠住该行的侧轴结束边界。
- **center**：弹性盒子元素在该行的侧轴（纵轴）上居中放置。（如果该行的尺寸小于弹性盒子元素的尺寸，则会向两个方向溢出相同的长度）。
- **baseline**：如弹性盒子元素的行内轴与侧轴为同一条，则该值与'flex-start'等效。其它情况下，该值将参与基线对齐。
- **stretch**：如果指定侧轴大小的属性值为'auto'，则其值会使项目的边距盒的尺寸尽可能接近所在行的尺寸，但同时会遵照'min/max-width/height'属性的限制。

flex 属性

flex 属性用于指定弹性子元素如何分配空间。

语法

flex: auto | initial | none | inherit | [flex-grow] || [flex-shrink] || [flex-basis]

各个值解析：

- auto: 计算值为 1 1 auto
- initial: 计算值为 0 1 auto
- none: 计算值为 0 0 auto
- inherit: 从父元素继承
- [flex-grow]: 定义弹性盒子元素的扩展比率。
- [flex-shrink]: 定义弹性盒子元素的收缩比率。
- [flex-basis]: 定义弹性盒子元素的默认基准值。

CSS3 弹性盒子属性

属性	描述
display	指定 HTML 元素盒子类型
flex-direction	指定了弹性容器中子元素的排列方式
justify-content	设置弹性盒子元素在主轴（横轴）方向上的对齐方式
align-items	设置弹性盒子元素在侧轴（纵轴）方向上的对齐方式
flex-wrap	设置弹性盒子的子元素超出父容器时是否换行
align-content	修改 flex-wrap 属性的行为，类似 align-items，但不是设置子元素对齐，而是设置行对齐
flex-flow	flex-direction 和 flex-wrap 的简写
order	设置弹性盒子的子元素排列顺序
align-self	在弹性子元素上使用。覆盖容器的 align-items 属性
flex	设置弹性盒子的子元素如何分配空间
justify-content	设置弹性盒子元素在主轴（横轴）方向上的对齐方式

表 7 弹性盒子属性表