**多重样式优先级**

内联样式）Inline style > （内部样式）Internal style sheet >（外部样式）External style sheet > 浏览器默认样式

**<!-- 外部样式 style.css -->**

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

**下列是一份优先级逐级增加的选择器列表：**

通用选择器（\*）

元素(类型)选择器

类选择器

属性选择器

伪类

ID 选择器

内联样式

!important 规则例外

伪类有：[:first-child](https://baike.baidu.com/item/:first-child/9102907" \t "https://baike.baidu.com/item/%E4%BC%AA%E7%B1%BB/_blank) ，[:link](https://baike.baidu.com/item/:link/8751680" \t "https://baike.baidu.com/item/%E4%BC%AA%E7%B1%BB/_blank):，vistited，[:hover](https://baike.baidu.com/item/:hover/8751300" \t "https://baike.baidu.com/item/%E4%BC%AA%E7%B1%BB/_blank):，[:active](https://baike.baidu.com/item/:active/8952813" \t "https://baike.baidu.com/item/%E4%BC%AA%E7%B1%BB/_blank)，[:focus](https://baike.baidu.com/item/:focus/8954055" \t "https://baike.baidu.com/item/%E4%BC%AA%E7%B1%BB/_blank)，:lang，:right，:left，:first

|  |  |  |  |
| --- | --- | --- | --- |
| **选择器** | **示例** | **示例说明** | **CSS** |
| [.](https://www.runoob.com/cssref/sel-class.html)*[class](https://www.runoob.com/cssref/sel-class.html)* | .intro | 选择所有class="intro"的元素 | 1 |
| [#](https://www.runoob.com/cssref/sel-id.html)*[id](https://www.runoob.com/cssref/sel-id.html)* | #firstname | 选择所有id="firstname"的元素 | 1 |
| [\*](https://www.runoob.com/cssref/sel-all.html) | \* | 选择所有元素 | 2 |
| *[element](https://www.runoob.com/cssref/sel-element.html)* | p | 选择所有<p>元素 | 1 |
| *[element,element](https://www.runoob.com/cssref/sel-element-comma.html)* | div,p | 选择所有<div>元素和<p>元素 | 1 |
| *[element](https://www.runoob.com/cssref/sel-element-element.html)**[element](https://www.runoob.com/cssref/sel-element-element.html)* | div p | 选择<div>元素内的所有<p>元素 | 1 |
| *[element](https://www.runoob.com/cssref/sel-element-gt.html)*[>](https://www.runoob.com/cssref/sel-element-gt.html)*[element](https://www.runoob.com/cssref/sel-element-gt.html)* | div>p | 选择所有父级是 <div> 元素的 <p> 元素 | 2 |
| *[element](https://www.runoob.com/cssref/sel-element-pluss.html)*[+](https://www.runoob.com/cssref/sel-element-pluss.html)*[element](https://www.runoob.com/cssref/sel-element-pluss.html)* | div+p | 选择所有紧跟在 <div> 元素之后的第一个 <p> 元素 | 2 |
| [[](https://www.runoob.com/cssref/sel-attribute.html)*[attribute](https://www.runoob.com/cssref/sel-attribute.html)*[]](https://www.runoob.com/cssref/sel-attribute.html) | [target] | 选择所有带有target属性元素 | 2 |
| [[](https://www.runoob.com/cssref/sel-attribute-value.html)*[attribute](https://www.runoob.com/cssref/sel-attribute-value.html)*[=](https://www.runoob.com/cssref/sel-attribute-value.html)*[value](https://www.runoob.com/cssref/sel-attribute-value.html)*[]](https://www.runoob.com/cssref/sel-attribute-value.html) | [target=-blank] | 选择所有使用target="-blank"的元素 | 2 |
| [[](https://www.runoob.com/cssref/sel-attribute-value-contains.html)*[attribute](https://www.runoob.com/cssref/sel-attribute-value-contains.html)*[~=](https://www.runoob.com/cssref/sel-attribute-value-contains.html)*[value](https://www.runoob.com/cssref/sel-attribute-value-contains.html)*[]](https://www.runoob.com/cssref/sel-attribute-value-contains.html) | [title~=flower] | 选择标题属性包含单词"flower"的所有元素 | 2 |
| [[](https://www.runoob.com/cssref/sel-attribute-value-lang.html)*[attribute](https://www.runoob.com/cssref/sel-attribute-value-lang.html)*[|=](https://www.runoob.com/cssref/sel-attribute-value-lang.html)*[language](https://www.runoob.com/cssref/sel-attribute-value-lang.html)*[]](https://www.runoob.com/cssref/sel-attribute-value-lang.html) | [lang|=en] | 选择 lang 属性等于 **en**，或者以 **en-** 为开头的所有元素 | 2 |
| [:link](https://www.runoob.com/cssref/sel-link.html) | a:link | 选择所有未访问链接 | 1 |
| [:visited](https://www.runoob.com/cssref/sel-visited.html) | a:visited | 选择所有访问过的链接 | 1 |
| [:active](https://www.runoob.com/cssref/sel-active.html) | a:active | 选择活动链接 | 1 |
| [:hover](https://www.runoob.com/cssref/sel-hover.html) | a:hover | 选择鼠标在链接上面时 | 1 |
| [:focus](https://www.runoob.com/cssref/sel-focus.html) | input:focus | 选择具有焦点的输入元素 | 2 |
| [:first-letter](https://www.runoob.com/cssref/sel-firstletter.html) | p:first-letter | 选择每一个<p>元素的第一个字母 | 1 |
| [:first-line](https://www.runoob.com/cssref/sel-firstline.html) | p:first-line | 选择每一个<p>元素的第一行 | 1 |
| [:first-child](https://www.runoob.com/cssref/sel-firstchild.html) | p:first-child | 指定只有当<p>元素是其父级的第一个子级的样式。 | 2 |
| [:before](https://www.runoob.com/cssref/sel-before.html) | p:before | 在每个<p>元素之前插入内容 | 2 |
| [:after](https://www.runoob.com/cssref/sel-after.html) | p:after | 在每个<p>元素之后插入内容 | 2 |
| [:lang(](https://www.runoob.com/cssref/sel-lang.html)*[language](https://www.runoob.com/cssref/sel-lang.html)*[)](https://www.runoob.com/cssref/sel-lang.html) | p:lang(it) | 选择一个lang属性的起始值="it"的所有<p>元素 | 2 |
| *[element1](https://www.runoob.com/cssref/sel-gen-sibling.html)*[~](https://www.runoob.com/cssref/sel-gen-sibling.html)*[element2](https://www.runoob.com/cssref/sel-gen-sibling.html)* | p~ul | 选择p元素之后的每一个ul元素 | 3 |
| [[](https://www.runoob.com/cssref/sel-attr-begin.html)*[attribute](https://www.runoob.com/cssref/sel-attr-begin.html)*[^=](https://www.runoob.com/cssref/sel-attr-begin.html)*[value](https://www.runoob.com/cssref/sel-attr-begin.html)*[]](https://www.runoob.com/cssref/sel-attr-begin.html) | a[src^="https"] | 选择每一个src属性的值以"https"开头的元素 | 3 |
| [[](https://www.runoob.com/cssref/sel-attr-end.html)*[attribute](https://www.runoob.com/cssref/sel-attr-end.html)*[$=](https://www.runoob.com/cssref/sel-attr-end.html)*[value](https://www.runoob.com/cssref/sel-attr-end.html)*[]](https://www.runoob.com/cssref/sel-attr-end.html) | a[src$=".pdf"] | 选择每一个src属性的值以".pdf"结尾的元素 | 3 |
| [[](https://www.runoob.com/cssref/sel-attr-contain.html)*[attribute](https://www.runoob.com/cssref/sel-attr-contain.html)*[\*=](https://www.runoob.com/cssref/sel-attr-contain.html)*[value](https://www.runoob.com/cssref/sel-attr-contain.html)*[]](https://www.runoob.com/cssref/sel-attr-contain.html) | a[src\*="runoob"] | 选择每一个src属性的值包含子字符串"runoob"的元素 | 3 |
| [:first-of-type](https://www.runoob.com/cssref/sel-first-of-type.html) | p:first-of-type | 选择每个p元素是其父级的第一个p元素 | 3 |
| [:last-of-type](https://www.runoob.com/cssref/sel-last-of-type.html) | p:last-of-type | 选择每个p元素是其父级的最后一个p元素 | 3 |
| [:only-of-type](https://www.runoob.com/cssref/sel-only-of-type.html) | p:only-of-type | 选择每个p元素是其父级的唯一p元素 | 3 |
| [:only-child](https://www.runoob.com/cssref/sel-only-child.html) | p:only-child | 选择每个p元素是其父级的唯一子元素 | 3 |
| [:nth-child(](https://www.runoob.com/cssref/sel-nth-child.html)*[n](https://www.runoob.com/cssref/sel-nth-child.html)*[)](https://www.runoob.com/cssref/sel-nth-child.html) | p:nth-child(2) | 选择每个p元素是其父级的第二个子元素 | 3 |
| [:nth-last-child(](https://www.runoob.com/cssref/sel-nth-last-child.html)*[n](https://www.runoob.com/cssref/sel-nth-last-child.html)*[)](https://www.runoob.com/cssref/sel-nth-last-child.html) | p:nth-last-child(2) | 选择每个p元素的是其父级的第二个子元素，从最后一个子项计数 | 3 |
| [:nth-of-type(](https://www.runoob.com/cssref/sel-nth-of-type.html)*[n](https://www.runoob.com/cssref/sel-nth-of-type.html)*[)](https://www.runoob.com/cssref/sel-nth-of-type.html) | p:nth-of-type(2) | 选择每个p元素是其父级的第二个p元素 | 3 |
| [:nth-last-of-type(](https://www.runoob.com/cssref/sel-nth-last-of-type.html)*[n](https://www.runoob.com/cssref/sel-nth-last-of-type.html)*[)](https://www.runoob.com/cssref/sel-nth-last-of-type.html) | p:nth-last-of-type(2) | 选择每个p元素的是其父级的第二个p元素，从最后一个子项计数 | 3 |
| [:last-child](https://www.runoob.com/cssref/sel-last-child.html) | p:last-child | 选择每个p元素是其父级的最后一个子级。 | 3 |
| [:root](https://www.runoob.com/cssref/sel-root.html) | :root | 选择文档的根元素 | 3 |
| [:empty](https://www.runoob.com/cssref/sel-empty.html) | p:empty | 选择每个没有任何子级的p元素（包括文本节点） | 3 |
| [:target](https://www.runoob.com/cssref/sel-target.html) | #news:target | 选择当前活动的#news元素（包含该锚名称的点击的URL） | 3 |
| [:enabled](https://www.runoob.com/cssref/sel-enabled.html) | input:enabled | 选择每一个已启用的输入元素 | 3 |
| [:disabled](https://www.runoob.com/cssref/sel-disabled.html) | input:disabled | 选择每一个禁用的输入元素 | 3 |
| [:checked](https://www.runoob.com/cssref/sel-checked.html) | input:checked | 选择每个选中的输入元素 | 3 |
| [:not(](https://www.runoob.com/cssref/sel-not.html)*[selector](https://www.runoob.com/cssref/sel-not.html)*[)](https://www.runoob.com/cssref/sel-not.html) | :not(p) | 选择每个并非p元素的元素 | 3 |
| [::selection](https://www.runoob.com/cssref/sel-selection.html) | ::selection | 匹配元素中被用户选中或处于高亮状态的部分 | 3 |
| [:out-of-range](https://www.runoob.com/cssref/sel-out-of-range.html) | :out-of-range | 匹配值在指定区间之外的input元素 | 3 |
| [:in-range](https://www.runoob.com/cssref/sel-in-range.html) | :in-range | 匹配值在指定区间之内的input元素 | 3 |
| [:read-write](https://www.runoob.com/cssref/sel-read-write.html) | :read-write | 用于匹配可读及可写的元素 | 3 |
| [:read-only](https://www.runoob.com/cssref/sel-read-only.html) | :read-only | 用于匹配设置 "readonly"（只读） 属性的元素 | 3 |
| [:optional](https://www.runoob.com/cssref/sel-optional.html) | :optional | 用于匹配可选的输入元素 | 3 |
| [:required](https://www.runoob.com/cssref/sel-required.html) | :required | 用于匹配设置了 "required" 属性的元素 | 3 |
| [:valid](https://www.runoob.com/cssref/sel-valid.html) | :valid | 用于匹配输入值为合法的元素 | 3 |
| [:invalid](https://www.runoob.com/cssref/sel-invalid.html) | :invalid | 用于匹配输入值为非法的元素 | 3 |

当使用简写属性时，属性值的顺序为：:

background-color

background-image

background-repeat

background-attachment

background-position

body {background:#ffffff url('img\_tree.png') no-repeat right top;}

background-attachment

|  |  |
| --- | --- |
| **值** | **描述** |
| scroll | 背景图片随着页面的滚动而滚动，这是默认的。 |
| fixed | 背景图片不会随着页面的滚动而滚动。 |
| local | 背景图片会随着元素内容的滚动而滚动。 |
| initial | 设置该属性的默认值。 [阅读关于 initial 内容](https://www.runoob.com/cssref/css-initial.html) |
| inherit | 指定 background-attachment 的设置应该从父元素继承。 [阅读关于 inherit 内容](https://www.runoob.com/cssref/css-inherit.html) |

文本的对齐方式

h1 {text-align:center;}

p.date {text-align:right;}

p.main {text-align:justify;}

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（如杂志和报纸）

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：

a {text-decoration:none;}

h1 {text-decoration:overline;}

h2 {text-decoration:line-through;}

h3 {text-decoration:underline;}

文本转换

p.uppercase {text-transform:uppercase;}

p.lowercase {text-transform:lowercase;}

p.capitalize {text-transform:capitalize;} 每个单词的首字母大写。

文本缩进

p {text-indent:50px;}

## 所有CSS文本属性。

|  |  |
| --- | --- |
| **属性** | **描述** |
| [color](https://www.runoob.com/cssref/pr-text-color.html) | 设置文本颜色 |
| [direction](https://www.runoob.com/cssref/pr-text-direction.html) | 设置文本方向。 |
| [letter-spacing](https://www.runoob.com/cssref/pr-text-letter-spacing.html) | 设置字符间距 |
| [line-height](https://www.runoob.com/cssref/pr-dim-line-height.html) | 设置行高 |
| [text-align](https://www.runoob.com/cssref/pr-text-text-align.html) | 对齐元素中的文本 |
| [text-decoration](https://www.runoob.com/cssref/pr-text-text-decoration.html) | 向文本添加修饰 |
| [text-indent](https://www.runoob.com/cssref/pr-text-text-indent.html) | 缩进元素中文本的首行 |
| [text-shadow](https://www.runoob.com/cssref/css3-pr-text-shadow.html) | 设置文本阴影 text-shadow: *h-shadow v-shadow blur color*; |
| [text-transform](https://www.runoob.com/cssref/pr-text-text-transform.html) | 控制元素中的字母 |
| [unicode-bidi](https://www.runoob.com/cssref/pr-text-unicode-bidi.html" \t "https://www.runoob.com/css/_blank) | 设置或返回文本是否被重写 |
| [vertical-align](https://www.runoob.com/cssref/pr-pos-vertical-align.html) | 设置元素的垂直对齐 |
| [white-space](https://www.runoob.com/cssref/pr-text-white-space.html) | 设置元素中空白的处理方式 |
| [word-spacing](https://www.runoob.com/cssref/pr-text-word-spacing.html) | 设置字间距 |

## white-space属性值

|  |  |
| --- | --- |
| **值** | **描述** |
| normal | 默认。空白会被浏览器忽略。 |
| pre | 空白会被浏览器保留。其行为方式类似 HTML 中的 <pre> 标签。 |
| nowrap | 文本不会换行，文本会在在同一行上继续，直到遇到 <br> 标签为止。 |
| pre-wrap | 保留空白符序列，但是正常地进行换行。 |
| pre-line | 合并空白符序列，但是保留换行符。 |
| inherit | 规定应该从父元素继承 white-space 属性的值。 |

字体系列

p{font-family:"Times New Roman", Times, serif;}

字体样式

p.normal {font-style:normal;}正常 - 正常显示文本  
p.italic {font-style:italic;}斜体 - 以斜体字显示的文字  
p.oblique {font-style:oblique;}倾斜的文字 - 文字向一边倾斜（和斜体非常类似，但不太支持）

1em和当前字体大小相等。在浏览器中默认的文字大小是16px。

## 所有CSS字体属性

|  |  |
| --- | --- |
| **Property** | **描述** |
| [font](https://www.runoob.com/cssref/pr-font-font.html) | 在一个声明中设置所有的字体属性 |
| [font-family](https://www.runoob.com/cssref/pr-font-font-family.html) | 指定文本的字体系列 |
| [font-size](https://www.runoob.com/cssref/pr-font-font-size.html) | 指定文本的字体大小 |
| [font-style](https://www.runoob.com/cssref/pr-font-font-style.html) | 指定文本的字体样式 |
| [font-variant](https://www.runoob.com/cssref/pr-font-font-variant.html) | 以小型大写字体或者正常字体显示文本。 |
| [font-weight](https://www.runoob.com/cssref/pr-font-weight.html) | 指定字体的粗细。 |

## 链接样式

a:link {color:#000000;} /\* 未访问链接\*/

a:visited {color:#00FF00;} /\* 已访问链接 \*/

a:hover {color:#FF00FF;} /\* 鼠标移动到链接上 \*/

a:active {color:#0000FF;} /\* 鼠标点击时 \*/

当设置为若干链路状态的样式，也有一些顺序规则：

a:hover 必须跟在 a:link 和 a:visited后面

a:active 必须跟在 a:hover后面

# CSS 列表

ul.a {list-style-type: circle;}

ul.b {list-style-type: square;}

ol.c {list-style-type: upper-roman;} 罗马数字

ol.d {list-style-type: lower-alpha;} 小写字母

Css盒模型

box-sizing: content-box | border-box

分别设置盒模型为标准模型（content-box）和IE模型（border-box）

.content-box设置为标准模型，它的元素宽度width = 100px

.border-box设置为IE模型，它的元素宽度width = content + 2 \* padding + 2 \* border = 70px + 2 x10px + 2 x5px = 100px

border:5px solid red;

border-width

border-style (required)

border-color

## 边框样式border-style

none: 默认无边框

dotted: 定义一个点线边框

dashed: 定义一个虚线边框

solid: 定义实线边框

double: 定义两个边框。 两个边框的宽度和 border-width 的值相同

groove: 定义3D沟槽边框。效果取决于边框的颜色值

ridge: 定义3D脊边框。效果取决于边框的颜色值

inset:定义一个3D的嵌入边框。效果取决于边框的颜色值

outset: 定义一个3D突出边框。 效果取决于边框的颜色值

border-width

为边框指定宽度有两种方法：可以指定长度值，比如 2px 或 0.1em(单位为 px, pt, cm, em 等)，或者使用 3 个关键字之一，它们分别是 thick（粗的） 、medium（默认值） 和 thin（细的）

## 边框-单独设置各边

p

{

border-top-style:dotted;

border-right-style:solid;

border-bottom-style:dotted;

border-left-style:solid;

}

# CSS3 border-radius 属性

**order-radius**: 4px 3px 6px / 2px 4px;  
*/\* 等价于： \*/*  
**border-top-left-radius**:     4px 2px;  
**border-top-right-radius**:    3px 4px;  
**border-bottom-right-radius**: 6px 2px;  
**border-bottom-left-radius**:  3px 4px;

# CSS 轮廓（outline）

轮廓（outline）是绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用。

轮廓（outline）属性指定元素轮廓的样式、颜色和宽度。

|  |
| --- |
|  |
| **属性** | **说明** | **值** | **CSS** |
| [outline](https://www.runoob.com/cssref/pr-outline.html) | 在一个声明中设置所有的轮廓属性 | *outline-color outline-style outline-width* inherit | 2 |
| [outline-color](https://www.runoob.com/cssref/pr-outline-color.html) | 设置轮廓的颜色 | *color-name hex-number rgb-number* invert inherit | 2 |
| [outline-style](https://www.runoob.com/cssref/pr-outline-style.html) | 设置轮廓的样式 | none dotted dashed solid double groove ridge inset outset inherit | 2 |
| [outline-width](https://www.runoob.com/cssref/pr-outline-width.html) | 设置轮廓的宽度 | thin medium thick *length* inherit |  |

margin 清除周围的（外边框）元素区域。margin 没有背景颜色，是完全透明的。

margin 可以单独改变元素的上，下，左，右边距，也可以一次改变所有的属性。

## padding（填充）

当元素的 padding（填充）内边距被清除时，所释放的区域将会受到元素背景颜色的填充。

单独使用 padding 属性可以改变上下左右的填充。

## 嵌套选择器

p{ }: 为所有 p 元素指定一个样式。

.marked{ }: 为所有 class="marked" 的元素指定一个样式。

.marked p{ }: 为所有 class="marked" 元素内的 p 元素指定一个样式。

p.marked{ }: 为所有 class="marked" 的 p 元素指定一个样式。

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

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。也就是说，该元素虽然被隐藏了，但仍然会影响布局。

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。

# CSS Position(定位)

## 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：

## 实例

p.pos\_fixed { position:fixed; top:30px; right:5px; }

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。

## 实例

h2.pos\_left { position:relative; left:-20px; } h2.pos\_right { position:relative; left:20px; }

## 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。

## 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）

## 所有的CSS定位属性

"CSS" 列中的数字表示哪个CSS(CSS1 或者CSS2)版本定义了该属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **说明** | **值** | **CSS** |
| [bottom](https://www.runoob.com/cssref/pr-pos-bottom.html) | 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。 | auto *length %* inherit | 2 |
| [clip](https://www.runoob.com/cssref/pr-pos-clip.html) | 剪辑一个绝对定位的元素 | *shape* auto inherit | 2 |
| [cursor](https://www.runoob.com/cssref/pr-class-cursor.html) | 显示光标移动到指定的类型 | *url* auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help | 2 |
| [left](https://www.runoob.com/cssref/pr-pos-left.html) | 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。 | auto *length %* inherit | 2 |
| [overflow](https://www.runoob.com/cssref/pr-pos-overflow.html) | 设置当元素的内容溢出其区域时发生的事情。 | auto hidden scroll visible inherit | 2 |
| [overflow-y](https://www.runoob.com/css/cssref/css3-pr-overflow-y.html) | 指定如何处理顶部/底部边缘的内容溢出元素的内容区域 | auto hidden scroll visible no-display no-content | 2 |
| [overflow-x](https://www.runoob.com/css/cssref//cssref/css3-pr-overflow-x.html) | 指定如何处理右边/左边边缘的内容溢出元素的内容区域 | auto hidden scroll visible no-display no-content | 2 |
| [position](https://www.runoob.com/cssref/pr-class-position.html) | 指定元素的定位类型 | absolute fixed relative static inherit | 2 |
| [right](https://www.runoob.com/cssref/pr-pos-right.html) | 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。 | auto *length %* inherit | 2 |
| [top](https://www.runoob.com/cssref/pr-pos-top.html) | 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。 | auto *length %* inherit | 2 |
| [z-index](https://www.runoob.com/cssref/pr-pos-z-index.html) | 设置元素的堆叠顺序 | *number* auto inherit |  |

## CSS Overflow

CSS overflow 属性可以控制内容溢出元素框时在对应的元素区间内添加滚动条。

overflow属性有以下值：

|  |  |
| --- | --- |
| **值** | **描述** |
| visible | 默认值。内容不会被修剪，会呈现在元素框之外。 |
| hidden | 内容会被修剪，并且其余内容是不可见的。 |
| scroll | 内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。 |
| auto | 如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。 |
| inherit | 规定应该从父元素继承 overflow 属性的值。 |

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。

## 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。

使用 clear 属性往文本中添加图片廊：

## 实例

.text\_line { clear:both; }

介绍一下清除浮动的几种方法：

1.给父级div定义伪类：after和zoom

// 用的多，推荐使用

.clearfloat:after{

display:block;

clear:both;

content:"";

overflow:hidden;

height:0;

}

.clearfloat{

zoom:1;//兼容IE6、7、8所带的浮动问题

}

这种是最常用的清除浮动的方法也是最推荐，直接在浮动的两个元素的父级加一个clearfloat类名就可以

2、父级div定义overflow:hidden

.div {

overflow:hidden;

}

不推荐，这种清除浮动的方法只推荐没有使用position或对overflow:hidden理解比较深的使用,必须定义width或zoom:1，同时不能定义height，使用overflow:hidden时，浏览器会自动检查浮动区域的高度，用多了就会出现问题的

overflow:hidden可以解决外边距塌陷问题：

当父级元素没有高度时，子元素会顶上去，从而造成页面塌陷，此时加上overflow:hidden属性，父级的高度就随子级容器及子级内容的高度而自适应。

为了更好的去兼容IE,加上一个 zoom:1

3、在浮动尾部添加空div标签 clear:both;

.clearfloat{

clear:both;

}

<div class='clearfloat'></div>

不推荐，会增加一些不必要的代码，此方法是以前主要使用的一种清除浮动方法，可以兼容IE

4、父级div定义 overflow:auto

.div{

overflow:auto;

}

不推荐使用，如果你需要出现滚动条或者确保你的代码不会出现滚动条就使用吧。

后面几种大家了解就好了

5、父级div定义为表格 display:table

.div{

display:table;

}

6、给父级div定义一个height

不推荐使用，只建议高度固定的布局时使用

7、父级div定义为表格display:table

.div{

display:table;

}

这种清除浮动的方式也不推荐,这样的话div就变成表格的形式了

8、加br标签

这种清除浮动的方式也不推荐,了解就好

# CSS 布局 - 水平 & 垂直对齐

## 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

如果仅仅是为了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;** 并将它放到 **块** 元素中:

img {

display: block;

margin: auto;

width: 40%;

}

## 左右对齐 - 使用定位方式

我们可以使用 **position: absolute;** 属性来对齐元素:

.right {

position: absolute;

right: 0px;

width: 300px;

border: 3px solid #73AD21;

padding: 10px;

}\

## 左右对齐 - 使用 float 方式

## 我们也可以使用 **float** 属性来对齐元素:

.right {

float: right;

width: 300px;

border: 3px solid #73AD21;

padding: 10px;

}

## 垂直居中对齐 - 使用 padding

.center {

padding: 70px 0;

border: 3px solid green;

}

如果要水平和垂直都居中，可以使用 **padding** 和 **text-align: center**:

## 垂直居中 - 使用 line-height

.center {

line-height: 200px;

height: 200px;

border: 3px solid green;

text-align: center;

}

/\* 如果文本有多行，添加以下代码: \*/

.center p {

line-height: 1.5;

display: inline-block;

vertical-align: middle;

}

## 垂直居中 - 使用 position 和 transform

除了使用 **padding** 和 **line-height** 属性外,我们还可以使用 **transform** 属性来设置垂直居中:

.center {

height: 200px;

position: relative;

border: 3px solid green;

}

.center p {

margin: 0;

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

}

# CSS 组合选择符

## 后代选择器

后代选择器用于选取某元素的后代元素。

以下实例选取所有 <p> 元素插入到 <div> 元素中:

div p

{

background-color:yellow;

}

## 子元素选择器

与后代选择器相比，子元素选择器（Child selectors）只能选择作为某元素直接/一级子元素的元素。

以下实例选择了<div>元素中所有直接子元素 <p> ：

div>p

{

background-color:yellow;

}

## 相邻兄弟选择器

相邻兄弟选择器（Adjacent sibling selector）可选择紧接在另一元素后的元素，且二者有相同父元素。

如果需要选择紧接在另一个元素后的元素，而且二者有相同的父元素，可以使用相邻兄弟选择器（Adjacent sibling selector）。

以下实例选取了所有位于 <div> 元素后的第一个 <p> 元素:

div+p

{

background-color:yellow;

}

## 后续兄弟选择器

后续兄弟选择器选取所有指定元素之后的相邻兄弟元素。

以下实例选取了所有 <div> 元素之后的所有相邻兄弟元素 <p> :

div~p

{

background-color:yellow;

}