# CSS布局

# 浮动float

* left
* right
* none — 默认值, 不浮动
* inherit — 继承父元素的浮动属性

多列浮动

**两列浮动**

div:nth-of-type(1) {

width: 48%;

float: left;

}

div:nth-of-type(2) {

width: 48%;

float: right;

}

一个向左一个向右整列浮动，各占48%，总跨度不大于100%

**三列浮动**

div:nth-of-type(1) {

width: 36%;

float: left;

}

div:nth-of-type(2) {

width: 30%;

float: left;

margin-left: 4%;

}

div:nth-of-type(3) {

width: 26%;

float: right;

}

若改变第二列的 [float](https://developer.mozilla.org/zh-CN/docs/Web/CSS/float)值为 right，则视觉顺序为

因为第二个<div>源代码顺序上比第三个<div>等级要高 (DOM上第二个<div>先出现，所以会更加靠右。

首字下沉

p::first-line {

text-transform: uppercase;

}

p::first-letter {

font-size: 3em;

border: 1px solid black;

background: red;

float: left;

padding: 2px;

margin-right: 4px;

}

清除浮动

footer {

clear: both;

}

clear可以使元素“停止浮动”，即这个元素和源码中后面的元素将不浮动（环绕）

[clear](https://developer.mozilla.org/zh-CN/docs/Web/CSS/clear) 可以取三个值：

* left：停止任何活动的左浮动
* right：停止任何活动的右浮动
* both：停止任何活动的左右浮动

footer {

clear: both;

margin-top: 4%;

}

清除页脚浮动的同时给出顶部外边距（ [margin](https://developer.mozilla.org/en-US/docs/Web/CSS/margin-top)）不起作用（浮动的元素存在于正常的文档布局流之外所以拥有margin，而取消浮动后没有）

解决方法：

添加新的<div class=“clearfix”>元素，在<footer>前面

.clearfix {

clear: both;

}

移除页脚的clear：both

# 定位技术position

**外边距折叠**：两个外边距接触，则较大者保留

静态定位static

默认

相对定位relative

允许移动位置，相对于元素原始位置移动

相对定位通常用来对布局进行微调，比如将一个图标往下调一点，以便放置文字

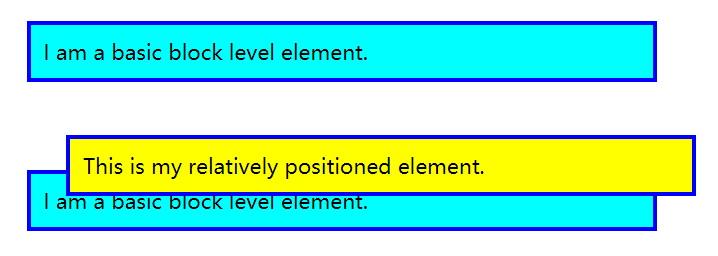
position: relative;

top: 30px;

left: 30px;

position本身不做事，需要top和left等使它移动

相对于原始位置，向右下移动



绝对定位absolute

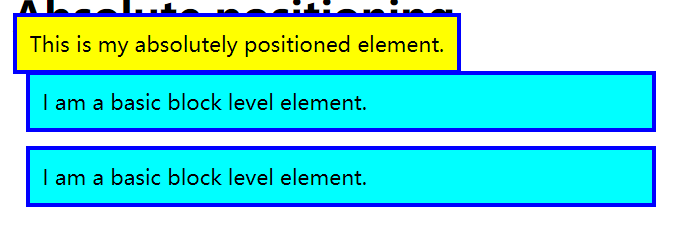
绝对定位用于将元素移动到web页面的任何位置，以创建复杂的布局

position: absolute;

top: 30px;

left: 30px;

[top](https://developer.mozilla.org/zh-CN/docs/Web/CSS/top)，[bottom](https://developer.mozilla.org/zh-CN/docs/Web/CSS/bottom)，[left](https://developer.mozilla.org/zh-CN/docs/Web/CSS/left)和[right](https://developer.mozilla.org/zh-CN/docs/Web/CSS/right)以不同的方式在绝对定位



指定元素应该从页面边界的顶部和左边的距离(确切地说，是 <html>元素的距离)，并且不保留元素原始位置

定位上下文

绝对定位元素会被放在<html>元素的**外面（即与<html>不作为一个整体，内容可能重叠）**，并且根据浏览器视口来定位

z-index

[z-index](https://developer.mozilla.org/zh-CN/docs/Web/CSS/z-index) 值影响定位元素位于该轴上的位置；正值将它们移动到堆栈上方，负值将它们向下移动到堆栈中。默认情况下，定位的元素都具有z-index为auto，实际上为0。

z-index: 1;

 较高的值将高于较低的值

固定定位

绝对定位固定元素是相对于 [<html>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/html) 元素或其最近的定位祖先，而固定定位固定元素则是相对于浏览器视口本身（滚动页面，固定定位内容保持不动）。可应用在导航栏

position：sticky

是相对位置和固定位置之间的混合，其允许定位的元件像它被相对定位一样动作，直到其滚动到某一阈值点（例如，从视口顶部10像素），之后它变得固定。

# CSS表格

display

form {

display: table;

margin: 0 auto;

}

form div {

display: table-row;

}

form label, form input {

display: table-cell;

margin-bottom: 10px;

}

form label {

width: 200px;

padding-right: 5%;

text-align: right;

}

form input {

width: 300px;

}

form p {

display: table-caption;

caption-side: bottom;

width: 300px;

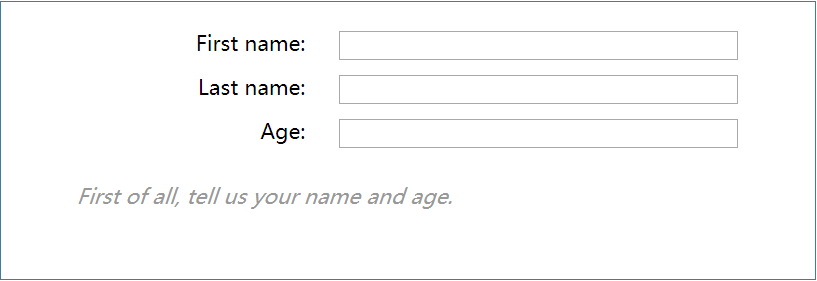
color: #999;

font-style: italic;

}

display: table-caption;——这使得它看起来就像一个表格[<caption>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/caption)

caption-side: bottom; 让标题在表格的底部



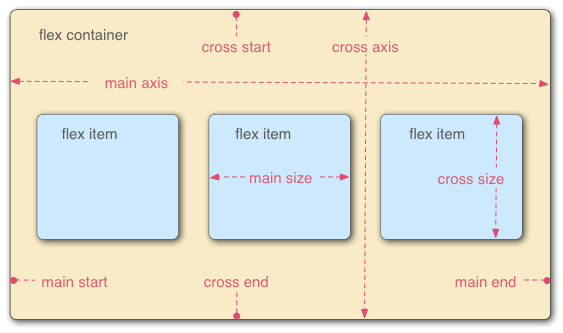
# 弹性盒子

指定元素的布局为flexible

display: flex;

默认情况下，多列布局具有大小相等的列，并且列的高度都是一样

假如你想设置行内元素为 flexible box，也可以置 [display](https://developer.mozilla.org/zh-CN/docs/Web/CSS/display) 属性的值为 inline-flex。



[flex-direction](https://developer.mozilla.org/zh-CN/docs/Web/CSS/flex-direction)指定主轴方向

默认值是row

flex-direction: column;

将元素设置为列布局

改变 [flex-direction](https://developer.mozilla.org/zh-CN/docs/Web/CSS/flex-direction) 属性值为 row-reverse —每一行元素排列的方向和原来相反。

换行

在section中添加，用于元素换行

flex-wrap: wrap

article中添加，规定每个元素宽度至少是200px；

flex: 200px;

flex-flow缩写

flex-flow: row wrap;

flex项的动态尺寸

flex: 1; 在行中获得相同数量的空间

还可以指定 flex 的最小值

article {

flex: 1 200px;

}

article:nth-of-type(3) {

flex: 2 200px;

}

每个flex 项将首先给出200px的可用空间，然后，剩余的可用空间将根据分配的比例共享

水平垂直对齐

div {

display: flex;

align-items: center;

justify-content: space-around;

}

[align-items](https://developer.mozilla.org/zh-CN/docs/Web/CSS/align-items) 控制 flex 项在交叉轴上的位置：

stretch 使所有 flex 项沿着交叉轴的方向拉伸以填充父容器

center使这些项保持其原有的高度，但是会在交叉轴居中

flex-start 或 flex-end在交叉轴的开始或结束处对齐所有的值

[justify-content](https://developer.mozilla.org/zh-CN/docs/Web/CSS/justify-content) 控制 flex 项在主轴上的位置：

flex-start/ flex-end 使所有flex项都位于主轴的开始或结尾处

center 可以使flex项在主轴居中

space-around 使所有flex项沿着主轴均匀地分布，在任意一端都会留有一点空间

space-between，和 space-around 非常相似，但不会在两端留下任何空间。

flex项排序

button:first-child {

order: 1;

}

所有order值默认为0

也可以设置负值

flex嵌套

设置一个元素为flex项目，那么他同样成为一个 flex 容器，它的孩子(直接子节点)也表现为 flexible box

# 网格布局

**列（column）**，**行（row）**，以及在每行和列之间的间隙——通常称为**沟槽（gutter）**

****创建流体网格****

将固定的宽度变为基于百分比的灵活（flexible）宽度的公式

target / context = result

用result来设置列宽

calc((6.25%\*4) + (2.08333333%\*3))