

1. 怎么用css显示a标签的背景图片
2. 怎么用css排版无序列表元素为左右两格分列展开
3. 嵌套div怎么用css，嵌套div是不是为了用css

## CSS引入

---

## 带透明度的颜色

---

```
<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
```

[https://www.w3school.com.cn/css/css\\_colors\\_rgb.asp](https://www.w3school.com.cn/css/css_colors_rgb.asp)

## 水平拼接小图片以形成整个背景

---

## 图片固定不滚动

---

background-attachment: fixed;

## 设置边框

---

border-style: solid;

border-width: 5px 20px; /\* 上边框和下边框为 5px，其他边为 20px \*/

border-color: red green blue yellow; /\* 上红、右绿、下蓝、左黄 \*/

## 分别设置4条边框

border-top-style: dotted;

border-right-style: solid;

border-bottom-style: dotted;

border-left-style: solid;

或者：

border-style: dotted solid double dashed;

# 简写设置

```
border: 5px solid red;  
border-left: 6px solid red;
```

# 圆角边框

border-radius: 5px;

# 所有 CSS 边框属性

属性	描述
<a href="#">border</a>	简写属性，在一条声明中设置所有边框属性。
<a href="#">border-color</a>	简写属性，设置四条边框的颜色。
<a href="#">border-radius</a>	简写属性，可设置圆角的所有四个 border-*-radius 属性。
<a href="#">border-style</a>	简写属性，设置四条边框的样式。
<a href="#">border-width</a>	简写属性，设置四条边框的宽度。
<a href="#">border-bottom</a>	简写属性，在一条声明中设置所有下边框属性。
<a href="#">border-bottom-color</a>	设置下边框的颜色。
<a href="#">border-bottom-style</a>	设置下边框的样式。
<a href="#">border-bottom-width</a>	设置下边框的宽度。
<a href="#">border-left</a>	简写属性，在一条声明中设置所有左边框属性。
<a href="#">border-left-color</a>	设置左边框的颜色。
<a href="#">border-left-style</a>	设置左边框的样式。
<a href="#">border-left-width</a>	设置左边框的宽度。
<a href="#">border-right</a>	简写属性，在一条声明中设置所有右边框属性。
<a href="#">border-right-color</a>	设置右边框的颜色。
<a href="#">border-right-style</a>	设置右边框的样式。
<a href="#">border-right-width</a>	设置右边框的宽度。
<a href="#">border-top</a>	简写属性，在一条声明中设置所有上边框属性。
<a href="#">border-top-color</a>	设置上边框的颜色。
<a href="#">border-top-style</a>	设置上边框的样式。
<a href="#">border-top-width</a>	设置上边框的宽度。

# display: flex 弹性盒模型

flex水平布局套路：

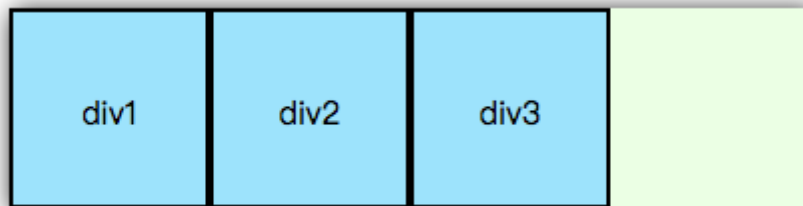
```
padding: 0.75rem 3rem;  
display: flex;  
justify-content: space-around;  
align-items: center;
```

[http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html?utm\\_source=tuicool](http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html?utm_source=tuicool)

最爱的解决方案，给父元素设置 `display: flex` 即可。

```
#wrap{  
  display: flex;  
}
```

效果图：



<http://blog.csdn.net/zmhawk>

还可以通过 `justify-content` 属性调整子元素的水平对齐方式：

```
#wrap{  
  display: flex;  
  justify-content: flex-start;  
}
```

**flex-start:**

默认，图同上。

**flex-end:**



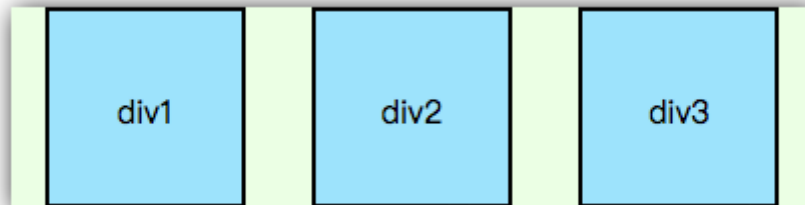
<http://blog.csdn.net/zmhawk>

**center:**



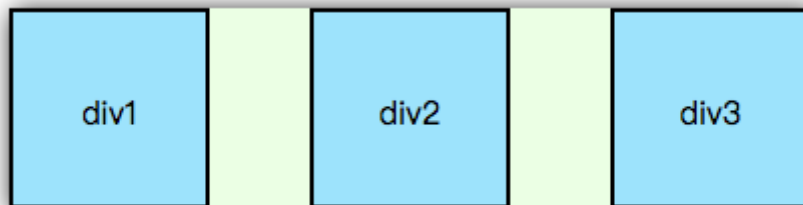
<http://blog.csdn.net/zmhawk>

**space-around:**



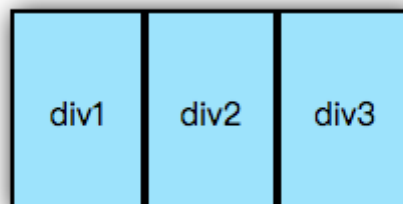
<http://blog.csdn.net/zmhawk>

**space-between:**



<http://blog.csdn.net/zmhawk>

不过当父元素宽度不够时，flex 默认是不会换行的，而是会等比例压缩，缩放比例 `flex-shrink` 属性或复合属性 `flex` 相关。



<http://blog.csdn.net/zmhawk>

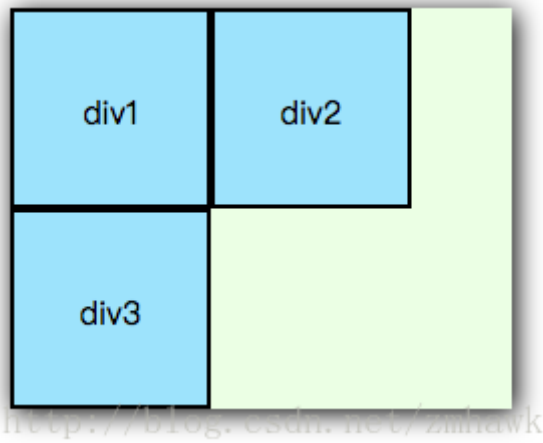
通过 `flex-wrap` 属性可以使其换行，该值有三个取值 `nowrap`、`wrap`、`wrap-reverse`，默认为 `nowrap`。

```
#wrap{
  display: flex;
  flex-wrap: nowrap;
}
```

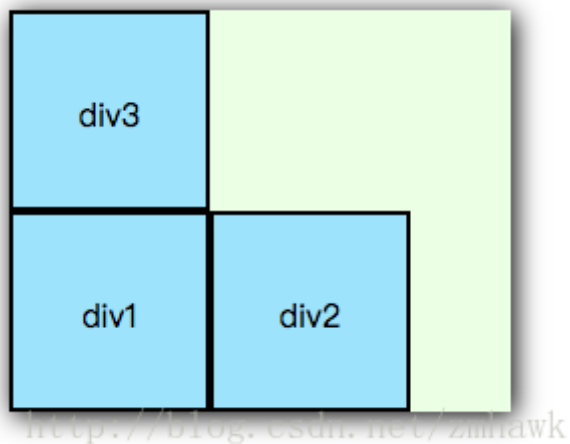
**nowrap**

默认，图略。

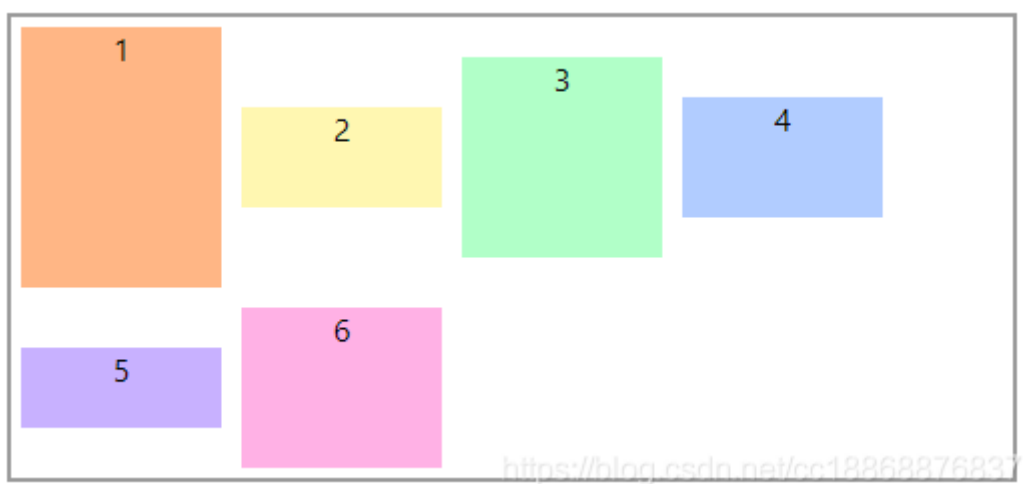
**wrap**



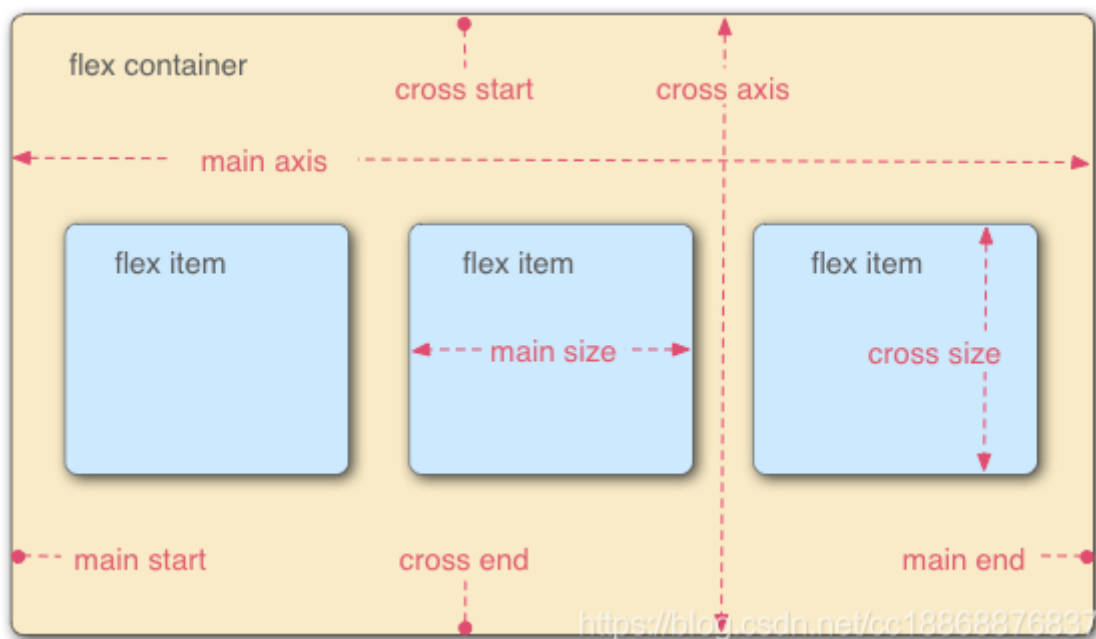
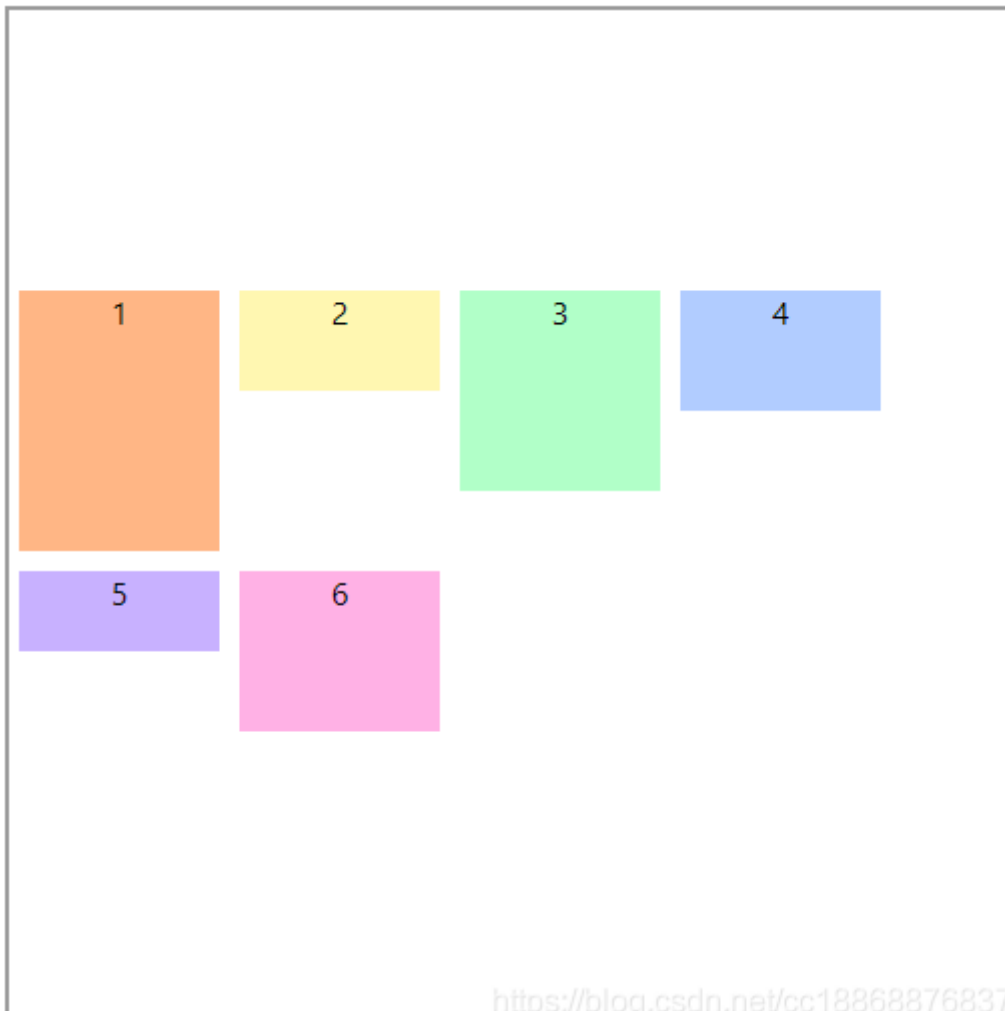
**wrap-reverse**



`align-items : center` 在多层是的表现：分别使得每一行都居中对其，但是整体不对齐。

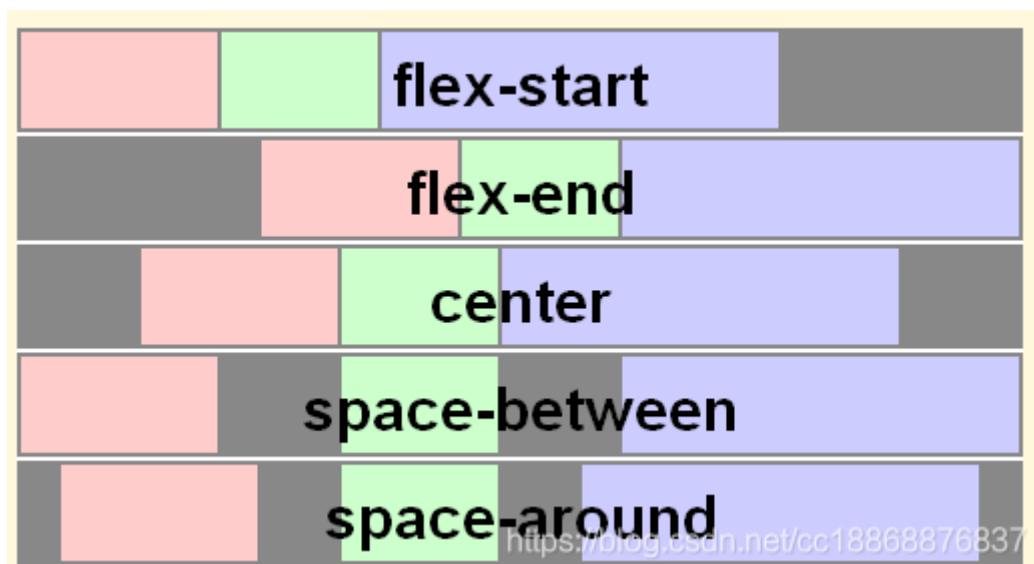


`align-content: center` 是将子项作为一个整体，然后这个整体在flex容器的交叉轴上居中对齐的。

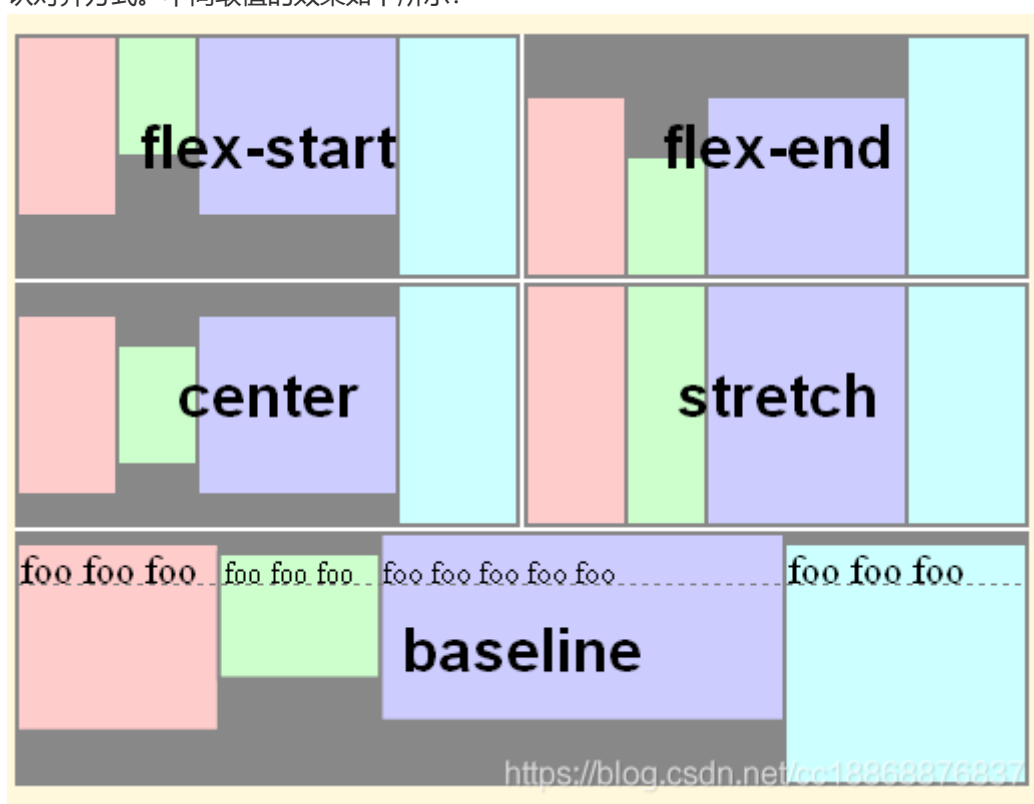


详见问题: <https://stackoverflow.com/questions/31250174/css-flexbox-difference-between-align-items-and-align-content>

- [justfy-content](#)属性可应用于所有的flex容器, 它的作用是设置flex子项 (flex items) 在主轴上的对齐方式。不同取值的效果如下所示:

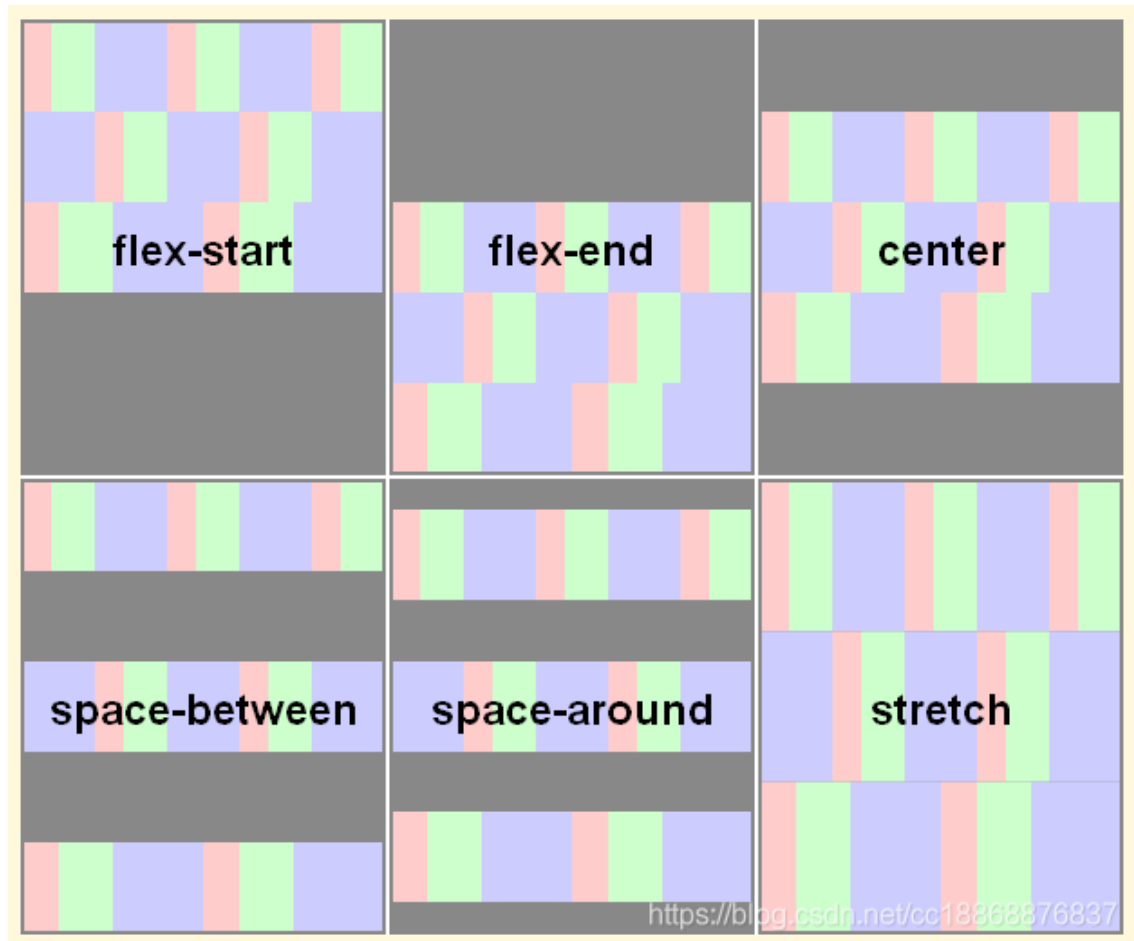


- [align-items](#)属性可以应用于所有的flex容器，它的作用是设置flex子项在每个flex行的交叉轴上的默认对齐方式。不同取值的效果如下所示：



- [align-content](#) 只适用多行的flex容器（也就是flex容器中的子项不止一行时该属性才有效果），它的作用是当flex容器在交叉轴上有多余的空间时，将子项作为一个整体（属性值为：flex-start、

flex-end、center时) 进行对齐。不同取值的效果如下所示:



## 文字边框设计1

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia">
<link rel="stylesheet" href="https://fonts.googleapis.com/icon?
family=Material+Icons">
<style>
.main {
  border: 5px solid green;
  /*outline是边框之外的*/
  outline-style: dotted;
  outline-color: red;
  outline-width: 2px;
  background-color: pink;

  padding: 0 30px 0 30px;
}

h2 {
  font-family: "Sofia";
  text-align: center;
  margin-top: 5px;
}
```



```
margin-bottom: 0px;
background-color: white;
text-shadow: 2px 2px 5px red;
}

p {
margin-top: 2px;
padding: 5px;
background-color: lightblue;
outline: 2px dotted red;
/*字间距和首行缩进*/
font-size: 1em; /* 1em=16px 浏览器默认值*/
letter-spacing: 1px;
text-indent: 34px; /* 缩进距离=2*字体大小+2*字体间距 */
line-height: 1.5;
text-align: justify;
/*字体粗细*/
font-weight: 120;
}

img {
width: 35px;
height: 35px;
vertical-align: middle;
}

span {
text-decoration: underline;
}

.button:link, .button:visited {
/*添加按钮背景; 改变字体颜色; 修改字体位置; 增加padding; 消除下划线; inline-block*/
background-color: #f44336;
color: white;
padding: 10px 20px;
text-align: center;
text-decoration: none;
display: inline-block;
}

.ita:link {
color: red;
text-decoration: none;
}

.ita:visited {
color: green;
text-decoration: none;
}

.ita:hover {
color: hotpink;
text-decoration: underline;
}
```

```
}

.ita:active {
  color: blue;
  text-decoration: underline;
}

.down {
  margin-top: 10px;
  display: flex; /*灵活水平排布子div块*/
  align-items: stretch; /*等高度*/
  justify-content: space-between; /**/
}

.img {
  order: 1; /*设置左右显示顺序*/
  background-color: lightgrey;
  margin-top: 10px;
}

.list {
  order: 0;
  margin-top: 10px;
  float: right;
}

.list:hover{

  background-color: lightgrey;
}

ul {
  background: #ff9999;
  padding: 20px;
}

ul:hover{
  list-style-image: url('/i/photo/sqpurple.gif');
}

ul li{
  background: #ffe5e5;
  padding: 3px;
  margin-left: 10px;
}

</style>
</head>
<body>

<div class="main">
  <h2>&nbsp;Chase</h2>
```

```
<p><span>此文本是盒子里的内容。我们添加了 50px 的内边距, 20px 的外边距和 15px 的绿色边框。</span>此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。</p></div>
```

```
<div class="down">
  <div class="img" id="left">
    <i class="material-icons">cloud</i>
    <i class="material-icons" style="font-size:48px;color:red;">favorite</i>
    <i class="material-icons">attachment</i>
    <i class="material-icons">computer</i>
    <i class="material-icons">traffic</i>
    <br />
    <a class="button" href="https://t.weixue100.com/toefl/special/vocab-result?aeid=4641117">超链接按钮</a>
    <br />
    <a class="ita" href="https://t.weixue100.com/toefl/special/vocab-result?aeid=4641117">变色超链接</a>
  </div>
  <div class="list" id="right">
    <ul class="unlist">
      <li>Coffee</li>
      <li>Tea</li>
      <li>Coca Cola</li>
      <li>Coffee</li>
      <li>Tea</li>
      <li>Coca Cola</li>
    </ul>
  </div>
</div>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia">
<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
<style>
.main {
  border: 5px solid green;
  /*outline是边框之外的*/
  outline-style: dotted;
  outline-color: red;
  outline-width: 2px;
  background-color: pink;
```

```
padding: 0 30px 0 30px;
}

h2 {
  font-family: "Sofia";
  text-align: center;
  margin-top: 5px;
  margin-bottom: 0px;
  background-color: white;
  text-shadow: 2px 2px 5px red;
}

p {
  margin-top: 2px;
  padding: 5px;
  background-color: lightblue;
  outline: 2px dotted red;
  /*字间距和首行缩进*/
  font-size: 1em; /* 1em=16px 浏览器默认值*/
  letter-spacing: 1px;
  text-indent: 34px; /* 缩进距离=2*字体大小+2*字体间距 */
  line-height: 1.5;
  text-align: justify;
  /*字体粗细*/
  font-weight: 120;
}

img {
  width: 35px;
  height: 35px;
  vertical-align: middle;
}

span {
  text-decoration: underline;
}

.button:link, .button:visited {
/*添加按钮背景; 改变字体颜色; 修改字体位置; 增加padding; 消除下划线; inline-block*/
  background-color: #f44336;
  color: white;
  padding: 10px 20px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

.ita:link {
  color: red;
  text-decoration: none;
}

.ita:visited {
```

```
color: green;
text-decoration: none;
}

.ita:hover {
color: hotpink;
text-decoration: underline;
}

.ita:active {
color: blue;
text-decoration: underline;
}

.down {
margin-top: 10px;
display: flex; /*灵活水平排布子div块*/
align-items: stretch; /*等高度*/
justify-content: space-between; /**/
}

.img {
order: 1; /*设置左右显示顺序*/
background-color: lightgrey;
margin-top: 10px;
}

.list {
order: 0;
background-color: lightgrey;
margin-top: 10px;
float: right;
}

ul {
background: #ff9999;
padding: 20px;
}

ul:hover{
list-style-image: url('/i/photo/sqpurple.gif');
}

ul li{
background: #ffe5e5;
padding: 3px;
margin-left: 10px;
}

#customers {
font-family: Arial, Helvetica, sans-serif;
/*border-collapse 属性设置是否将表格边框折叠为单一边框*/
border-collapse: collapse;
```

```
width: 40%;  
margin-top: 10px;  
}  
  
#customers td, #customers th {  
    border: 1px solid #ddd;  
    padding: 8px;  
}  
  
#customers tr:nth-child(even){background-color: #f2f2f2;}  
  
#customers tr:hover {background-color: #ddd;}  
  
#customers th {  
    padding-top: 12px;  
    padding-bottom: 12px;  
    text-align: left;  
    background-color: #4CAF50;  
    color: white;  
}  
  
</style>  
</head>  
<body>  
  
<div class="main">  
    <h2>&nbspChase</h2>  
    <p><span>此文本是盒子里的内容。我们添加了 50px 的内边距，20px 的外边距和 15px 的绿色边框。</span>此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。  
    此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子  
    里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的  
    内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文  
    本。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。此文本是盒子里的内容。</p>  
</div>  
  
<div class="down">  
    <div class="img" id="left">  
        <i class="material-icons">cloud</i>  
        <i class="material-icons" style="font-size:48px;color:red;">favorite</i>  
        <i class="material-icons">attachment</i>  
        <i class="material-icons">computer</i>  
        <i class="material-icons">traffic</i>  
        <br />  
        <a class="button" href="https://t.weixue100.com/toefl/special/vocab-result?aeid=4641117">超链接按钮</a>  
        <br />  
        <a class="ita" href="https://t.weixue100.com/toefl/special/vocab-result?aeid=4641117">变色超链接</a>  
    </div>  
    <div class="list" id="right">  
        <ul class="unlist">  
            <li>Coffee</li>  
            <li>Tea</li>  
            <li>Coca Cola</li>
```

```
    <li>Coffee</li>
    <li>Tea</li>
    <li>Coca Cola</li>
  </ul>
</div>
```

```
<table id="customers">
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Address</th>
    <th>City</th>
  </tr>
  <tr>
    <td>Alibaba</td>
    <td>Ma Yun</td>
    <td>No. 699, Wangshang Road, Binjiang District</td>
    <td>Hangzhou</td>
  </tr>
  <tr>
    <td>APPLE</td>
    <td>Tim Cook</td>
    <td>1 Infinite Loop Cupertino, CA 95014</td>
    <td>Cupertino</td>
  </tr>
  <tr>
    <td>BAIDU</td>
    <td>Li YanHong</td>
    <td>Lixiang guoji dasha, No 58, Beisihuanxilu</td>
    <td>Beijing</td>
  </tr>
  <tr>
    <td>Canon</td>
    <td>Tsuneji Uchida</td>
    <td>One Canon Plaza Lake Success, NY 11042</td>
    <td>New York</td>
  </tr>
  <tr>
    <td>Google</td>
    <td>Larry Page</td>
    <td>1600 Amphitheatre Parkway Mountain View, CA 94043</td>
    <td>Mountain View</td>
  </tr>
  <tr>
    <td>HUAWEI</td>
    <td>Ren Zhengfei</td>
    <td>Putian Huawei Base, Longgang District</td>
    <td>Shenzhen</td>
  </tr>
  <tr>
    <td>Microsoft</td>
    <td>Bill Gates</td>
    <td>15700 NE 39th St Redmond, WA 98052</td>
    <td>Redmond</td>
  </tr>
```

```
<tr>
  <td>Nokia</td>
  <td>Olli-Pekka Kallasvuo</td>
  <td>P.O. Box 226, FIN-00045 Nokia Group</td>
  <td>Helsinki</td>
</tr>
<tr>
  <td>SONY</td>
  <td>Kazuo Hirai</td>
  <td>Park Ridge, NJ 07656</td>
  <td>Park Ridge</td>
</tr>
<tr>
  <td>Tencent</td>
  <td>Ma Huateng</td>
  <td>Tencent Building, High-tech Park, Nanshan District</td>
  <td>Shenzhen</td>
</tr>
</table>

</div>

</body>
</html>
```

## 简易网站设计

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}

/* 设置 body 元素的样式 */
body {
  font-family: Arial;
  margin: 0;
}

/* 标题 / LOGO */
.header {
  padding: 60px;
  text-align: center;
  background: #1abc9c;
  color: white;
```



```
}

/* 设置顶部导航栏样式 */
.navbar {
  display: flex;
  background-color: #333;
}

/* 设置导航条链接演示 */
.navbar a {
  color: white;
  padding: 14px 20px;
  text-decoration: none;
  text-align: center;
}

/* 更改鼠标悬停时的颜色 */
.navbar a:hover {
  background-color: #ddd;
  color: black;
}

/* 列容器 */
.row {
  display: flex;
  flex-wrap: wrap;
}

/* 创建并排的非等列 */
/* 侧栏 / 左侧列 */
.side {
  flex: 30%;
  background-color: #f1f1f1;
  padding: 20px;
}

/* 主列 */
.main {
  flex: 70%;
  background-color: white;
  padding: 20px;
}

/* 伪图像，仅供示例 */
.fakeimg {
  background-color: #aaa;
  width: 100%;
  padding: 20px;
}

/* 页脚 */
.footer {
  padding: 20px;
  text-align: center;
  background: #ddd;
}
```



```

<h5>标题描述, 2021 年 1 月 2 日</h5>
<div class="fakeimg" style="height:200px;">图像</div>
<p>一些文本..</p>
<p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur
adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>
</div>
</div>

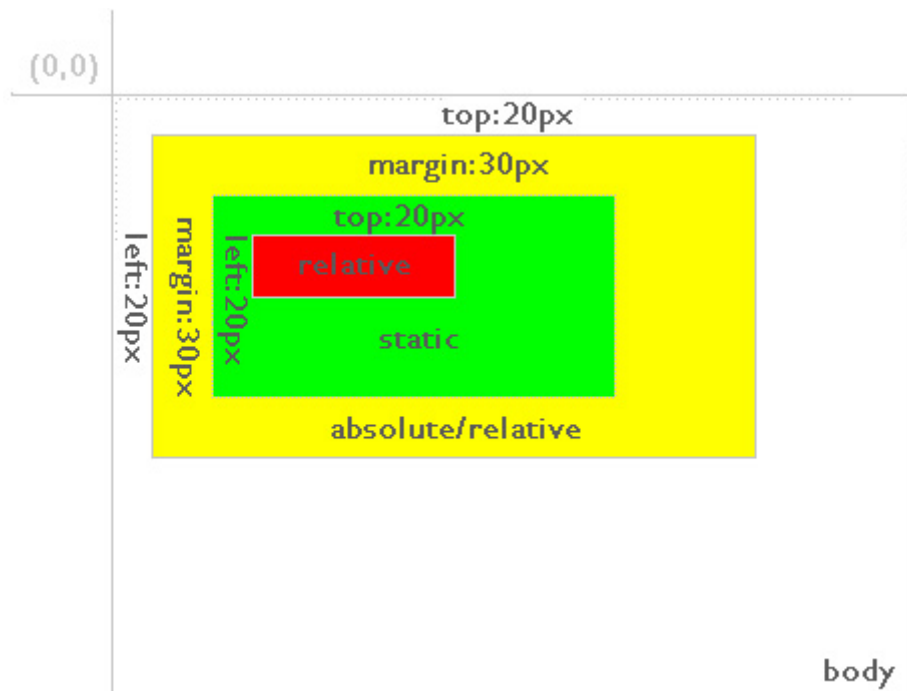
<!-- Footer -->
<div class="footer">
  <h2>页脚</h2>
</div>

</body>
</html>

```

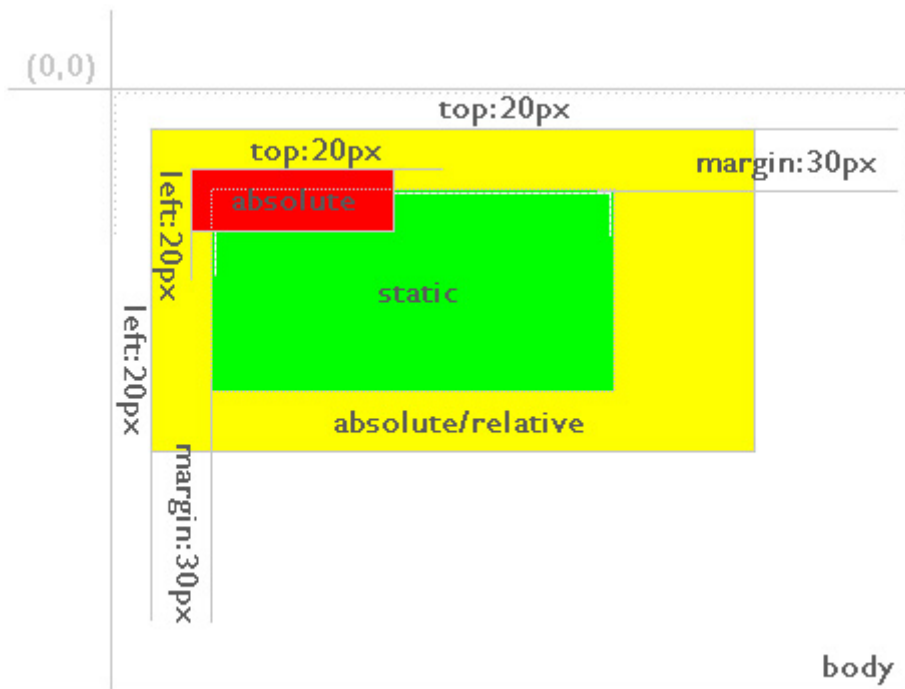
## position: absolute和relative的区别

- relative: 不论父元素是absolute还是relative, 其位置永远是相对于最近父层的

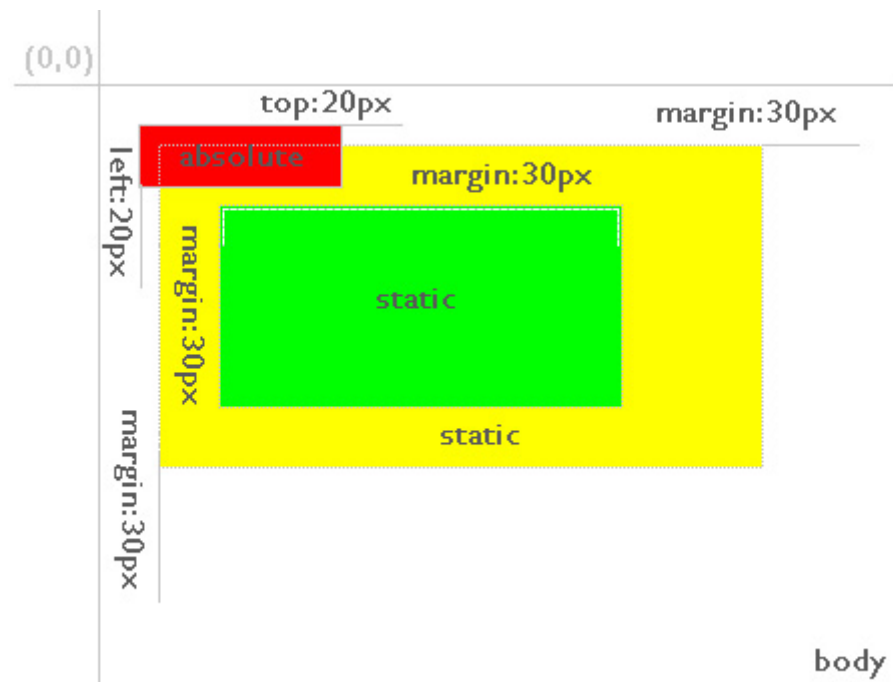


- absolute: 其位置相对于最近的定义为absolute或relative的父层。如果其父层中都未定义absolute或relative, 则其将相对body进行定位。

--外层为absolute或relative



--外层不是absolute或relative，自动定位body



absolute	相对于最近的声明为absolute或relative的父类
fixed	生成绝对定位的元素，相对于浏览器窗口进行定位。
relative	相对于最近的父类
static	默认值。没有定位，元素出现在正常的流中（忽略 top, bottom, left, right 或者 z-index 声明）。
inherit	规定应该从父元素继承 position 属性的值。
sticky	粘性，滚动后跟着页面一起动

如果子元素开启了绝对定位，则父元素都会开启相对定位

经验：如果你写了absolute, 一般都得补一个relative, 如果你写了absolute或fixed, -定要补top和left, sticky兼容性很差，主要用于面试装逼

## display 隐藏、内联、块元素

- block的好处在于可以设置背景图片，修改width和height
- inline的好处在于可以水平排布

none	此元素不会被显示。
block	此元素将显示为块级元素，此元素前后会带有换行符。
inline	默认。此元素会被显示为内联元素，元素前后没有换行符。
inline-block	行内块元素。（CSS2.1 新增的值）

## overflow

`overflow` 属性指定在元素的内容太大而无法放入指定区域时是剪裁内容还是添加滚动条。

`overflow` 属性可设置以下值：

- `visible` - 默认。溢出没有被剪裁。内容在元素框外渲染
- `hidden` - 溢出被剪裁，其余内容将不可见
- `scroll` - 溢出被剪裁，同时添加滚动条以查看其余内容
- `auto` - 与 `scroll` 类似，但仅在必要时添加滚动条

**注释：** `overflow` 属性仅适用于具有指定高度的块元素。

```
overflow-x: scroll;
```

```
overflow-y: scroll;
```

## float

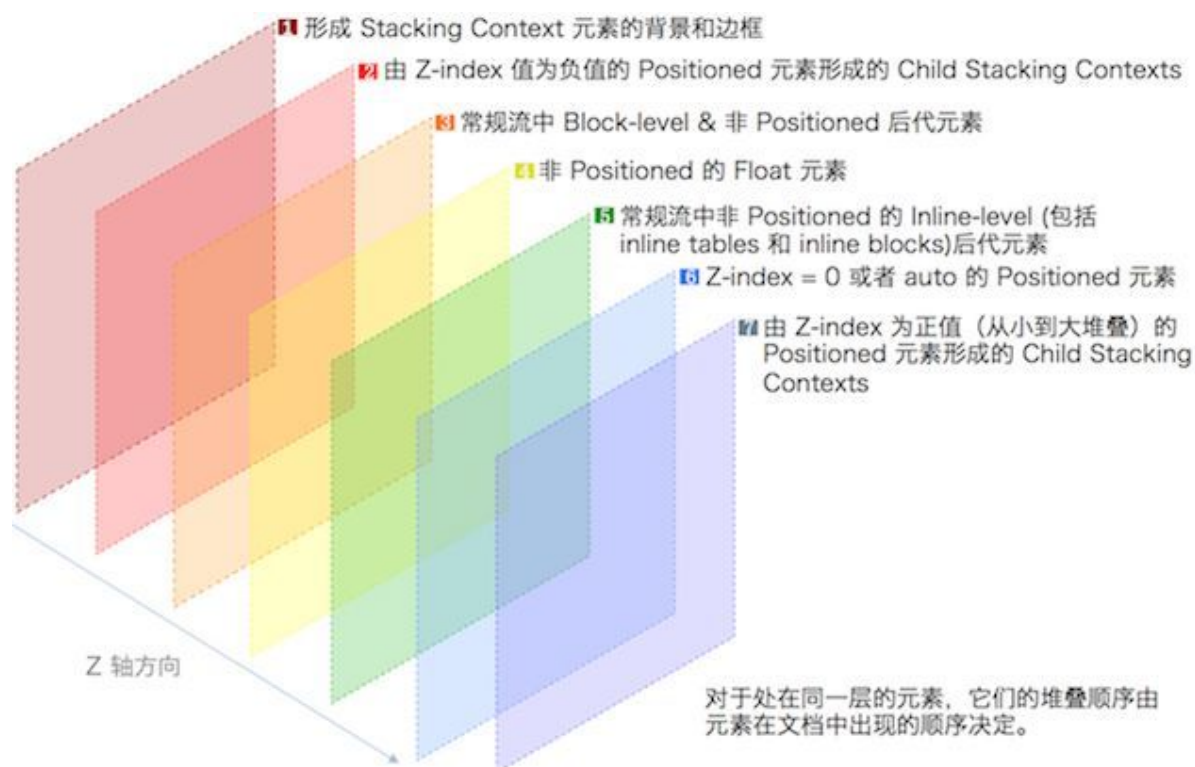
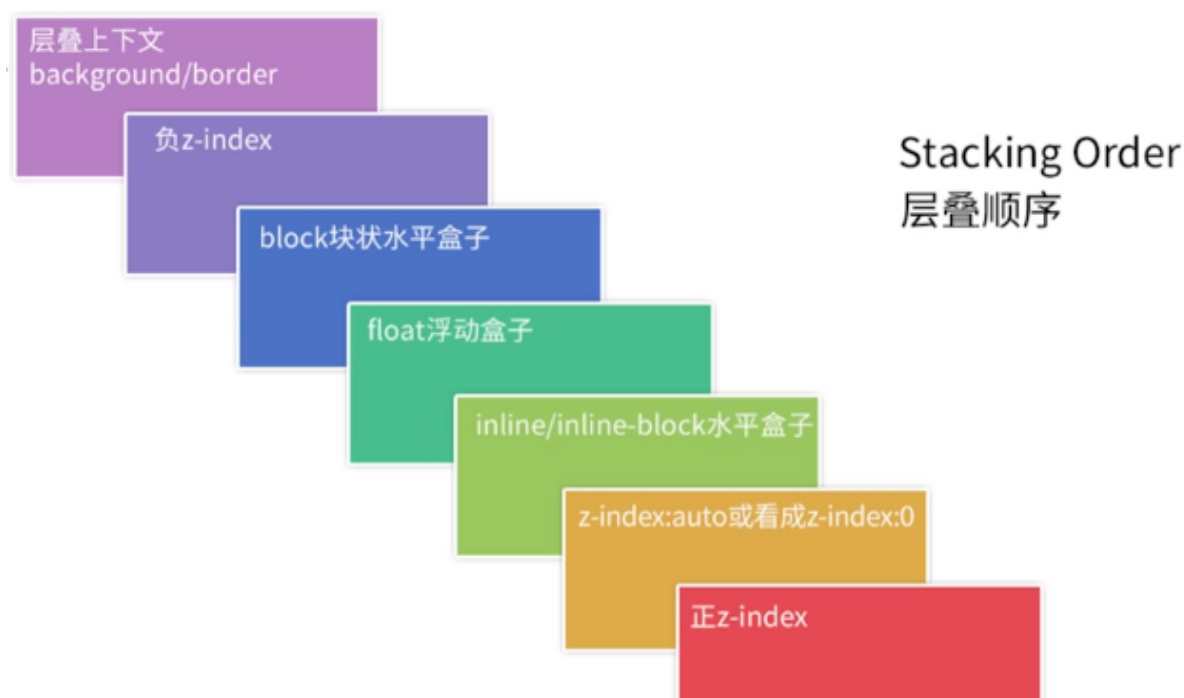
float的设计初衷就是用来做图片漂浮（文本环绕）。

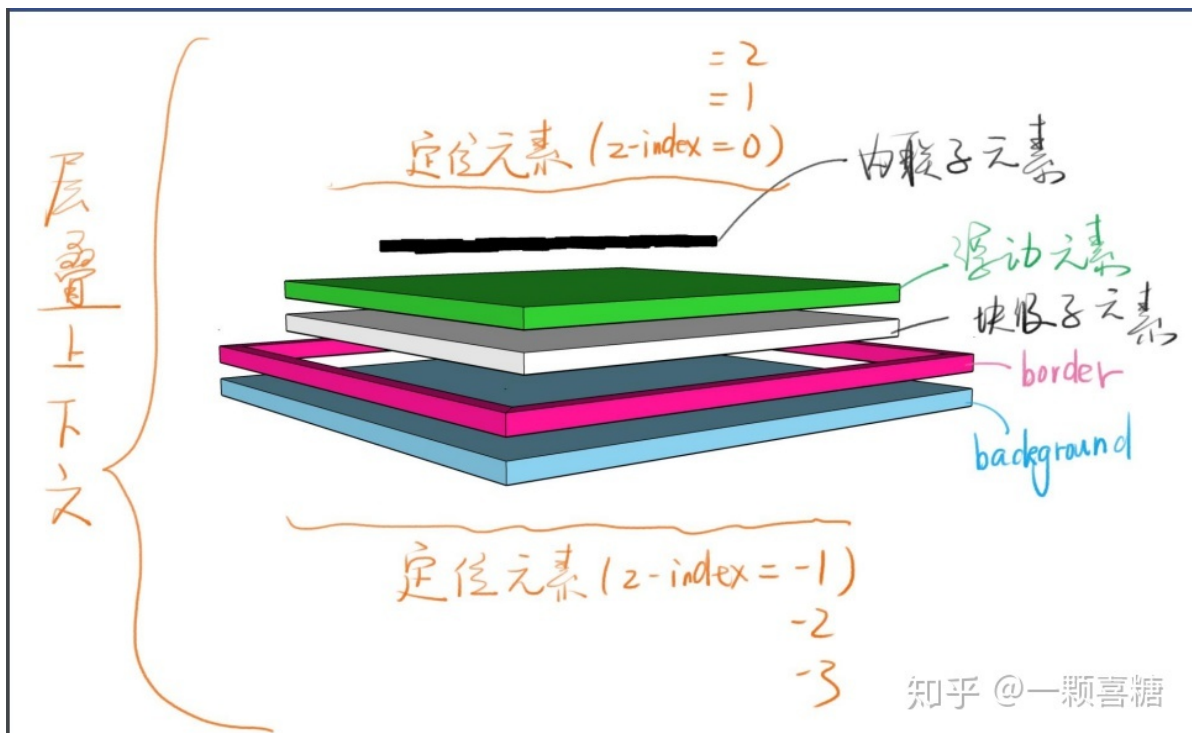
```
img {  
  float: left;  
}
```

## 文档流和层叠上下文

<https://zhuanlan.zhihu.com/p/124263128>

# 著名的7阶层叠水平(stacking level)





- block元素在最底层
- 非position的float元素在block元素上面
- inline元素在float元素上面
- position的float元素在inline元素上面，相当于z-index=0
- 后面是z-index>0的元素
- **注意：父元素永远没办法盖住子元素**

使用float: left

使用position: absolute

## 居中

文字居中：

```
方法1:  
display: flex;  
justify-content: center; /*文字水平居中*/  
align-items: center; /*文字y轴居中*/
```

```
方法2:  
text-align: center;  
line-height: height;
```

图片居中：

```
img {  
  display: block;  
  margin-left: auto;  
  margin-right: auto;  
  width: 40%;  
}
```

div居中:

```
.center {
  margin: auto;
  width: 50%;
  border: 3px solid green;
  padding: 20px;
}
```

## 选择器

- 空格: 选定所有代的所有后代
- '>': 选定第一代的所有后代
- '+': 选定之后第一个相邻的兄弟
- '~': 选定所有兄弟, 不论是前面还是后面

`nth-of-type` 和 `nth-last-of-type` 都是CSS选择器中的伪类, 用于匹配同一父元素下的指定类型的子元素中, 特定位置的元素。

- `nth-of-type` 用于选择同一父元素下指定类型的子元素中, 第n个元素。例如:

```
cssCopy code/* 选择同一父元素下的所有p元素中的第2个元素 */
p:nth-of-type(2) {
  color: red;
}
```

- `nth-last-of-type` 用于选择同一父元素下指定类型的子元素中, 倒数第n个元素。例如:

```
cssCopy code/* 选择同一父元素下的所有p元素中的倒数第2个元素 */
p:nth-last-of-type(2) {
  color: blue;
}
```

## 输入框

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=text], select {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: block;
  border: 1px solid #ccc; /* 输入框边框 */
  border-radius: 4px;
  box-sizing: border-box;
```



```

background-color: white; /* 输入框背景的颜色 */
color: black; /* 输入光标的颜色 */
}

#change {
  width: 15%;
  transition: width 0.4s ease-in-out;
}

input[type=submit] {
  width: 100%;
  background-color: #4CAF50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
}

input[type=submit]:hover {
  background-color: #45a049;
}

input[type=button],input[type=reset] {
  background-color: #4CAF50;
  border: none;
  color: white;
  padding: 16px 32px;
  text-decoration: none;
  margin: 4px 2px;
  cursor: pointer;
}

/* 增加焦点 */
input[type=text]:focus {
  background-color: lightgrey;
}

/* 展开输入框 */
#change:focus {
  width: 100%;
}

div {
  border-radius: 5px;
  background-color: #f2f2f2;
  padding: 20px;
}
</style>

<body>

<h1>Using CSS to style an HTML Form</h1>

```

```

<div>
  <form action="/action_page.php">
    <label for="fname">First Name</label>
    <input type="text" id="fname" name="firstname" placeholder="Your name.."
value="Gates">

    <label for="lname">Last Name</label>
    <input type="text" id="lname" name="lastname" placeholder="Your last
name..">

    <label for="change">展开输入框</label>
    <input type="text" id="change" placeholder="Search....">

    <label for="country">Country</label>
    <select id="country" name="country">
      <option value="australia">Australia</option>
      <option value="canada">China</option>
      <option value="usa">USA</option>
    </select>

    <input type="submit" value="Submit">
    <input type="button" value="按钮">
    <input type="reset" value="清空所有用户输入">
  </form>
</div>

</body>
</html>

```

## Using CSS to style an HTML Form

First Name

Last Name

Country

Australia

▼

Submit

## 按钮样式

[https://www.w3school.com.cn/css/css3\\_buttons.asp](https://www.w3school.com.cn/css/css3_buttons.asp)

常用套路:

```
border: none;
outline: none;
cursor: pointer;
```

## 放上去变色

```
.button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 16px 32px;
  text-align: center;
  font-size: 16px;
  margin: 4px 2px;
  transition-duration: 0.3s; /*动画渐变*/
  cursor: pointer; /*鼠标停在上面变为手指*/
}
```

```
.button {
  border-radius: 5px;
  background-color: #555555;
  color: white;
  border: 2px solid #555555;
}
```

```
.button:hover {
  background-color: white;
  color: black;
}
```

## 放上去显示阴影

```
.button2:hover {
  box-shadow: 0 12px 16px 0 rgba(0,0,0,0.14),0 17px 50px 0 rgba(0,0,0,0.19);
}
```

## 禁用

```
.disabled {
  opacity: 0.6;
  cursor: not-allowed;
}
```

## 悬停显示箭头

```
<!DOCTYPE html>
<html>
<head>
```

```
<style>
.button {
  border-radius: 4px;
  background-color: #f4511e;
  border: none;
  color: #FFFFFF;
  text-align: center;
  font-size: 28px;
  padding: 20px;
  width: 200px;
  transition: all 0.5s;
  cursor: pointer;
  margin: 5px;
}

.button span {
  display: inline-block; /*保证图标和span不重叠*/
  cursor: pointer;
  position: relative;
  transition: 0.5s;
}

.button span:before {
  content: '\00bc';
  position: absolute; /*相对于span*/
  opacity: 1;
  top: 0;
  left: -20px;
  transition: 0.5s;
}

/* 在span后面新添加一个新图标元素 */
.button span:after {
  content: '\00bb';
  position: absolute; /*相对于span*/
  opacity: 0;
  top: 0;
  right: -20px;
  transition: 0.5s;
}

/*文字左移，给箭头留出位置*/
.button:hover span {
  padding-right: 25px;
}

.button:hover span:after {
  opacity: 1;
  right: 0;
}

</style>
```

```
</head>
<body>

<h1>带动画效果的按钮</h1>

<button class="button"><span>请悬停在我上方</span></button>

</body>
</html>
```

## 闪烁按钮

```
.button {
  position: relative;
  background-color: #4CAF50;
  border: none;
  font-size: 28px;
  color: #FFFFFF;
  padding: 20px;
  width: 200px;
  text-align: center;
  transition-duration: 0.4s;
  text-decoration: none;
  overflow: hidden;
  cursor: pointer;
}

.button:after {
  content: "";
  background: black;
  display: block;
  position: absolute;
  padding-top: 2000%;
  padding-left: 1000%;
  margin-left: -20px !important;
  margin-top: -120%;
  opacity: 0;
  transition: all 0.8s
}

.button:active:after {
  padding: 0;
  margin: 0;
  opacity: 0.6;
  transition: 0s;
}
```

## transition渐变

```
<style>
```

```
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s, height 2s, transform 2s;
}

div:hover {
  width: 300px;
  height: 300px;
  transform: rotate(180deg);
}
</style>
```

```
div {
  transition-property: width;
  transition-duration: 2s;
  transition-timing-function: linear;
  transition-delay: 1s;
}
```

## animation动画

[https://www.w3school.com.cn/css/css3\\_animations.asp](https://www.w3school.com.cn/css/css3_animations.asp)

```
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: infinite;
  /*animation-direction: reverse; 反向动画 */、
  /*animation-direction: alternate;"alternate" 使动画先向前运行，然后向后运行*/
}

@keyframes example {
  0% {background-color:red; left:0px; top:0px;}
  25% {background-color:yellow; left:200px; top:0px;}
  50% {background-color:blue; left:200px; top:200px;}
  75% {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
</style>
```

简写:

```
div {
  animation-name: example;
  animation-duration: 5s;
  animation-timing-function: linear;
  animation-delay: 2s;
  animation-iteration-count: infinite;
  animation-direction: alternate;
}

div {
  animation: example 5s linear 2s infinite alternate;
}
```

执行完动画后维持最终状态:

```
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  position: relative;
  animation-name: example;
  animation-duration: 1s;
  animation-fill-mode: both;
}

@keyframes example {
  from {top: 0px; left: 0px;}
  to {top: 200px; left: 200px; background-color: blue;}
}
</style>
```

animation-fill-mode 属性可接受以下值:

- **none** - 默认值。动画在执行之前或之后不会对元素应用任何样式。
- **forwards** - 元素将保留由最后一个关键帧设置的样式值（依赖 animation-direction 和 animation-iteration-count）。
- **backwards** - 元素将获取由第一个关键帧设置的样式值（取决于 animation-direction），并在动画延迟期间保留该值。
- **both** - 动画会同时遵循向前和向后的规则，从而在两个方向上扩展动画属性。

## CSS+Js+Html

```
<!DOCTYPE html>
<html>
<head>
<style>
#myImg {
  border-radius: 5px;
  cursor: pointer;
  transition: 0.3s;
```

```

}

#myImg:hover {opacity: 0.7;}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  padding-top: 100px; /* Location of the box */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.9); /* Black w/ opacity */
}

/* Modal Content (image) */
.modal-content {
  margin: auto;
  display: block;
  width: 80%;
  max-width: 1500px;
}

/* Caption of Modal Image */
#caption {
  margin: auto;
  display: block;
  width: 80%;
  max-width: 700px;
  text-align: center;
  color: #ccc;
  padding: 10px 0;
  height: 150px;
}

/* Add Animation */
.modal-content, #caption {
  animation-name: zoom;
  animation-duration: 0.6s;
}

@keyframes zoom {
  from {transform: scale(0.1)}
  to {transform: scale(1)}
}

/* The Close Button */
.close {
  position: absolute;
  top: 15px;
  right: 35px;

```



```

    color: #f1f1f1;
    font-size: 40px;
    font-weight: bold;
    transition: 0.3s;
}

.close:hover,
.close:focus {
    color: #bbb;
    text-decoration: none;
    cursor: pointer;
}

/* 100% Image width on Smaller Screens */
@media only screen and (max-width: 700px){
    .modal-content {
        width: 100%;
    }
}
</style>
</head>
<body>

<h1>体会Html、Css、Js的关系和应用</h1>

<p>
    Html: 基础元素引入-图片、标题、文字... <br>
    Css: 样式布局absolute flex, 部分响应动画-渐入, 隐藏元素display:none配合js交互<br>
    Js: 获取Html元素, 操纵Dom树, 修改Html元素内容captionText.innerHTML 和Css样式
    modal.style.display<br>
</p>



<!-- The Modal -->
<div id="myModal" class="modal">
    <span class="close">x</span>
    <img class="modal-content" id="img01">
    <div id="caption"></div>
</div>

<script>
// Get the modal
var modal = document.getElementById('myModal');

// Get the image and insert it inside the modal - use its "alt" text as a
caption
var img = document.getElementById('myImg');
var modalImg = document.getElementById("img01");
var captionText = document.getElementById("caption");
img.onclick = function(){
    modal.style.display = "block";
    modalImg.src = this.src;
    captionText.innerHTML = this.alt;
}

```

```
// Get the <span> element that closes the modal
var span = document.getElementsByClassName("close")[0];

// When the user clicks on <span> (x), close the modal
span.onclick = function() {
    modal.style.display = "none";
}
</script>

</body>
</html>
```

## 使用object-fit缩放图片

`object-fit` 属性可接受如下值:

- `fill` - 默认值。调整替换后的内容大小，以填充元素的内容框。如有必要，将拉伸或挤压物体以适应该对象。
- `contain` - 缩放替换后的内容以保持其纵横比，同时将其放入元素的内容框。
- `cover` - 调整替换内容的大小，以在填充元素的整个内容框时保持其长宽比。该对象将被裁剪以适应。
- `none` - 不对替换的内容调整大小。
- `scale-down` - 调整内容大小就像没有指定内容或包含内容一样（将导致较小的具体对象尺寸）

## 使用padding+backgroundColor设置span

```
padding: 0.25rem 1rem;
border-radius: 25px;
background-color: #b94517;
```

例如图中的【3】就是一个span，使用padding撑开，然后染色和设置圆角。