# 第一部分参考：[学习笔记05-css学习1](学习笔记05-css学习1.docx)

# 第二部分：

# 7.组合选择器

## 1）组选择器，用逗号隔开，可以选择多个元素如body,div,p{},如

|  |
| --- |
|  |

### 注意：需要用逗号隔开

## 2）后的选择器，如：div p{} 表示div的后的p可以是儿子孙子重孙子，看看下面的例子

|  |
| --- |
|  |

### 然后我们添加一个div在里面添加一个ul在ul的li里面添加一个p，发现div p{}的样式依然有效效果如下

|  |
| --- |
|  |

## 3.亲儿子选择器，只选择儿子，孙子不受影响

|  |
| --- |
|  |

### 可见，只有儿子受影响

## 4.选择器链，也就是一个元素同时有几种选择器，这个有时候非常有用。实例

|  |
| --- |
|  |

### 需要注意的是，1.选择器链不要有空格，这几个选择器影响连在一起写，在用一个选择器无法选取特定元素时，可以考虑使用选择器链2.在选择器中，标签选择器必须首先写，然后才是类选择器或者id选择器

## 5.后代选择器+选择器链

|  |
| --- |
|  |

### 意思是：空格后面的选择器链是同一个元素，他是div的后代，可以是孙子或者重孙子

# 8.css定位

## 1.相对定位

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。

relative表示，相对于默认位置（即static时的位置）进行偏移，即定位基点是元素的默认位置

它必须搭配top、bottom、left、right这四个属性一起使用，用来指定偏移的方向和距离。

**注意：相对定位是相对于元素自己原来的位置而不是相当于其他元素**

## 2.绝对定位，是以父元素的位置作为参考点的

**absolute表示，相对于上级元素（一般是父元素）进行偏移，即定位基点是父元素。**

**它有一个重要的限制条件：定位基点（一般是父元素）不能是static定位，否则定位基点就会变成整个网页的根元素html。**

**另外，absolute定位也必须搭配top、bottom、left、right这四个属性一起使用。**

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。

注意，absolute定位的元素会被"正常页面流"忽略，即在"正常页面流"中，该元素所占空间为零，周边元素不受影响。

## 实例

### 新建一个test.html文件，创建3层div，为了学习方便，我们使用内部样式内容如下

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Document</title>      <style>         #grand{            border: 2px solid pink;            height: 600px;            padding: 20px;            background-color: antiquewhite;           }         #dad{           border: 2px dashed skyblue;           height: 550px;           padding: 10px;           background-color: cornflowerblue;           }         .child{          width: 200px;          height: 150px;           }         #one{           background-color: aquamarine;         }         #two{          background-color: beige;           }         #three{           background-color: blue;         }      </style>  </head>  <body>     <div id="grand">      我是爷爷      <div id="dad">          我是父亲          <div class="child" id="one">div number1</div>          <div class="child" id="two">div number2</div>          <div class="child" id="three">div number3</div>      </div>     </div>  </body>  </html> |

### 定位之前，效果如下

|  |
| --- |
|  |

### 1）此时使用的是默认static定位也就是没有定位，我们想给第二个son做绝对定位，我们在#two选择器里面添加代码

|  |
| --- |
|  |

#### 此时，因为父亲没有孙子定位，爷爷也没有设置定位，body也没有，所以它就参考html根元素，效果如下

|  |
| --- |
|  |

### 2）我们给爷爷添加一个相对定位

|  |
| --- |
|  |

#### 效果：

|  |
| --- |
|  |

## 3）我们给父亲添加一个相对定位

|  |
| --- |
|  |

#### 效果：

|  |
| --- |
|  |

## 3.固定定位

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

fixed表示，相对于视口（viewport，浏览器窗口）进行偏移，即定位基点是浏览器窗口。这会导致元素的位置不随页面滚动而变化，好像固定在网页上一样。

它如果搭配top、bottom、left、right这四个属性一起使用，表示元素的初始位置是基于视口计算的，否则初始位置就是元素的默认位置。

## 4.静态定位

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。

每个块级元素占据自己的区块（block），元素与元素之间不产生重叠

注意，static定位所导致的元素位置，是浏览器自主决定的，所以这时top、bottom、left、right这四个属性无效。

## 5.粘性定位sticky

**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

**sticky跟前面四个属性值都不一样，它会产生动态效果，很像relative和fixed的结合：一些时候是relative定位（定位基点是自身默认位置），另一些时候自动变成fixed定位（定位基点是视口）。**

**因此，它能够形成"动态固定"的效果。比如，网页的搜索工具栏，初始加载时在自己的默认位置（relative定位）。**

**sticky生效的前提是，必须搭配top、bottom、left、right这四个属性一起使用，不能省略，否则等同于relative定位，不产生"动态固定"的效果。原因是这四个属性用来定义"偏移距离"，浏览器把它当作sticky的生效门槛。**

**它的具体规则是，当页面滚动，父元素开始脱离视口时（即部分不可见），只要与sticky元素的距离达到生效门槛，relative定位自动切换为fixed定位；等到父元素完全脱离视口时（即完全不可见），fixed定位自动切换回relative定位**。

## sticky 的应用

sticky定位可以实现一些很有用的效果。除了上面提到"动态固定"效果，这里再介绍两个。

### 5.1 堆叠效果

堆叠效果（stacking）指的是页面滚动时，下方的元素覆盖上方的元素。下面是一个图片堆叠的例子，下方的图片会随着页面滚动，覆盖上方的图片（查看 [demo](https://jsbin.com/fegiqoquki/edit?html,css,output)）。



HTML 代码就是几张图片。

<div><img src="pic1.jpg"></div>

<div><img src="pic2.jpg"></div>

<div><img src="pic3.jpg"></div>

CSS 代码极其简单，只要两行。

div {

position: sticky;

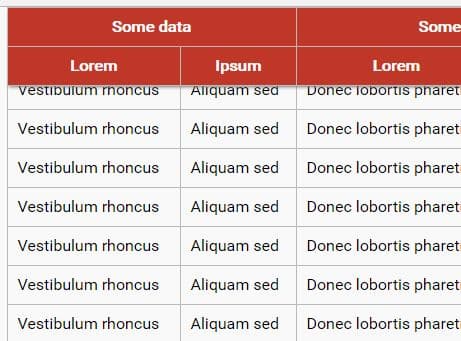
top: 0;

}

它的原理是页面向下滚动时，每张图片都会变成fixed定位，导致后一张图片重叠在前一张图片上面。详细解释可以看[这里](https://dev.to/vinceumo/slide-stacking-effect-using-position-sticky-91f)。

### 5.2 表格的表头锁定

大型表格滚动的时候，表头始终固定，也可以用sticky实现（查看 [demo](https://jsbin.com/decemanohe/edit?html,css,output)）。



CSS 代码也很简单。

th {

position: sticky;

top: 0;

}

需要注意的是，sticky必须设在<th>元素上面，不能设在<thead>和<tr>元素，因为这两个元素没有relative定位，也就无法产生sticky效果

# 9.z-index

## 可以决定元素的层级，数值越大越往上层，数值越小越往下

# 10.元素显示属性display

## 有3种方式：inline，block，inline-block

## 需要注意：

### inline和inline-block都可以在同一行显示

### block和inlinne-block都可以任意设置大小，inline方式的大小是有标签里面的内容决定的，调整宽高没有效果

### 即使你把一个block元素的宽度设置得很小，它仍然会独自占据一行

# 11.浮动float

## 参考：[CSS-浮动详解 - 掘金 (juejin.cn)](https://juejin.cn/post/7084895918855553032)

## 什么是 CSS Float（浮动）？

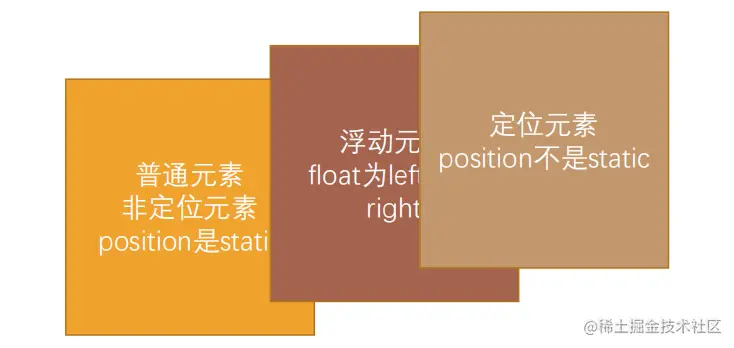


CSS 的 Float（浮动），会使元素向左或向右移动，其周围的元素也会重新排列。

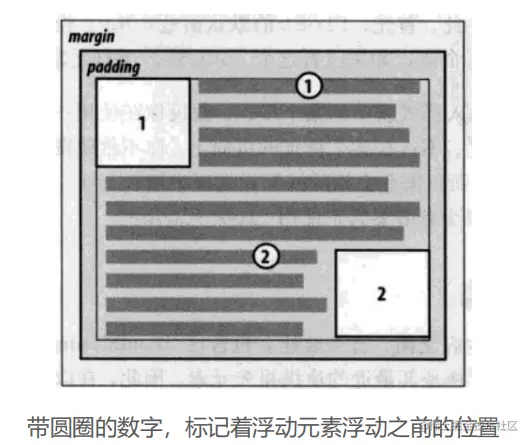
Float（浮动），往往是用于图像，但它在布局时一样非常有用。

### 浮动规则

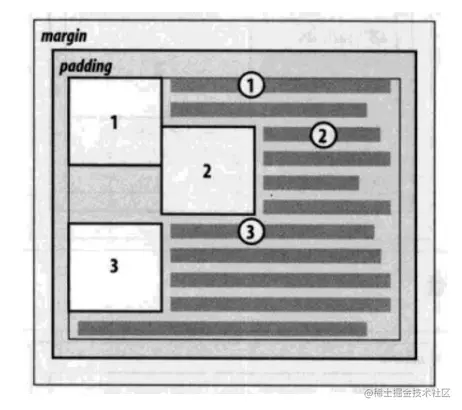
* 规则一：向左浮动或者向右浮动
  + **向左或向右**方向移动，直到**自己的边界紧贴着包含块**（一般是父元素）或者**其他浮动元素的边界**为止
  + 定位元素会**层叠**在浮动元素上面



* 规则二：不能超出包含块
  + 如果元素是向左（右）浮动，浮动元素的左（右）边界不能超出包含块的左（右）边界



* 规则三：浮动元素不能层叠
  + 如果一个元素浮动，另一个浮动元素已经在那个位置了，后浮动的元素将**紧贴着前一个浮动元素**（左浮找左浮，右浮找右浮）
  + 如果水平方向剩余的空间不够显示浮动元素，浮动元素将向下移动，直到有充足的空间为止



* 规则四：浮动元素会将行内级元素内容推出
  + 浮动元素**不能与行内级内容层叠**，行内级内容将会被浮动元素推出
  + 比如**行内级元素**、**inline-block元素**、块级元素的**文字内容**
  + 图文环绕效果



* 规则五：浮动只能左右浮动, 不能超出本行的高度
  + 行内级元素、inline-block元素浮动后，其顶部将与所在行的顶部对齐

## 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，下面的文本流将环绕在它左边：

## 实例

img { float:right; }

## 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。

在这里，我们对图片廊使用 float 属性：

## 实例

.thumbnail { float:left; width:110px; height:90px; margin:5px; }

## 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; }

## 以前技术笔记落后，float被用来做很复杂的界面，现在很多界面效果都不需要使用float了，现在只有你需要文字环绕图片是时候才使用float

# 12.如何制作响应式网站

## 1.所谓是响应式网站就是能够根据窗口的宽度或者高度的变化来自动调整外观的网站

## 2.有4种方式可以实现响应式网站功能

|  |
| --- |
|  |

### 1）媒体查询

|  |
| --- |
|  |

#### 如果不写，默认的媒体是屏幕，条件必须写在括号里面，又如

|  |  |
| --- | --- |
| |  | | --- | |  | |

#### 上面的css的意思是当屏幕宽度大于600px，设置div的宽高位200px标签背景色位蓝色，如果屏幕宽度小于等于600px，设置div的宽高为100px ，背景色不变，条件可以组合起来用

|  |
| --- |
|  |

### 2）css 网格

|  |
| --- |
|  |

### 3）弹性盒子，flex布局

|  |
| --- |
|  |

### 弹性盒子在布局的时候，会根据每一个元素的flex属性的值来计算该元素的占比然后根据这个占比\*容器的总宽度来获取元素的宽度，这个是动态的当元素的flex值改变了，它的宽度或者高度就会改变

### 4）bootstrap

|  |
| --- |
|  |

### 注意：需要使用bootstrap的CDN或者本地安装bootstrap