## 定位

### 浮动

浮动问题：一个父元素包含多个浮动元素，但浮动元素不会影响父元素的宽度，使得副元素坍塌，高度变为0.

🡪需要清除浮动

#### Clear

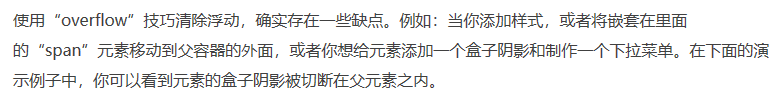
方法：在容器结束标签之前添加一个空标签，样式设置“clear：both”

缺点：导致空标签的迅速堆积

#### Overflow

方法：在具有浮动元素的父容器中设置“overflow：auto”

缺点：



#### Clearfix

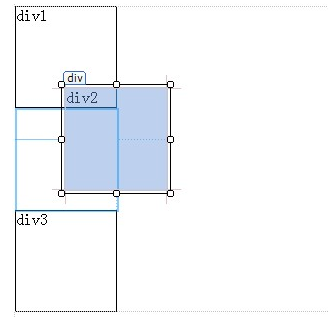
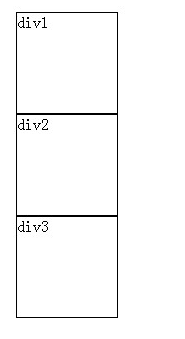
方法：在父元素上使用“：before”和“：after”两个伪类。通常定义一个类名，把这个类名加到需要清除浮动的容器上。



### 定位

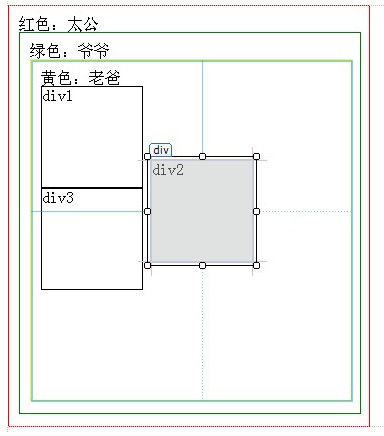
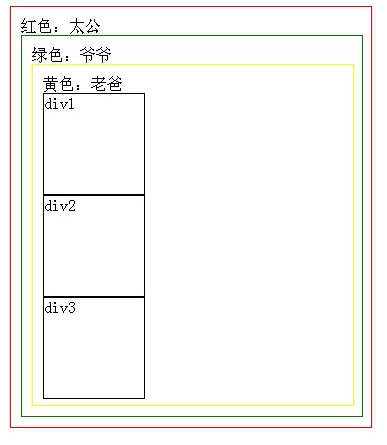
#### Relative

* 相对于默认位置移动，top，right，bottom，left设置位置，top与left优先于bottom与right。
* 不脱离文档流，所以元素本身所占的位置会保留。



#### Absolute

* 需要至少一个祖先元素设置了相对定位，不然相对于页面主体进行定位。
* 脱离文档流，重新定位后元素是不会占着原来的位置的



#### Fixed

#### z-index

* position为上述三个之一才能使用。
* 值越大，显示在越上层。

## 居中

https://css-tricks.com/centering-css-complete-guide/

### 水平居中

#### Inline/inline-\* 元素

在父元素设置”text-align:center”

#### Block 元素

设置元素“margin:?px auto”

Faking”float:center”链接：<https://css-tricks.com/float-center/>

#### 多个block元素

将多个block元素设置”display:inline-block”

将父元素设置”text-align:center”

### 垂直居中

#### Inline/inline-\*元素

单行：

1. Padding-top=padding-bottom
2. 设置元素”line-height=height”

多行

1. Padding-top=padding-bottom
2. .center-table设置”display:table”,.center-table p设置”display:table-cell;vertical-align:middle”。
3. .flex-center-vertically设置”display:flex;justify-content:center;flex-direction:column”(容器有具体高度时)

#### Block元素

有具体高度：

.parent{

position:relative;

}

.child{

position:absolute;

top:50%;

height:100px;

margin-top:-50px; /\* account for padding and border if not using box-sizing: border-box; \*/

}

无具体高度：

将上文margin-top的值设置为translateY(-50%)

### 水平垂直居中

#### 有具体宽度和高度

.parent{

position:relative;

}

.child{

position:absolute;

top:50%;

left:50%;

height:100px;

width:200px;

margin-top:-50px; /\* account for padding and border if not using box-sizing: border-box; \*/

margin-left:-100px;

}

#### 无具体宽度和高度

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

#### Flexbox

.parent {

display: flex;

justify-content: center;

align-items: center;

}

#### Grid

body, html {

height: 100%;

display: grid;

}

span { /\* thing to center \*/

margin: auto;

}