

[笔记][LIKE-H5][15]

前端

[笔记][LIKE-H5][15]

- 121. 网页布局方式(掌握)
- 122. 浮动流基本概念(掌握)
- 123. 浮动元素脱标(理解)
- 124. 浮动元素排序规则(理解)
- 125. 浮动元素贴靠现象(掌握)
- 126. 浮动元素字围现象(理解)
- 127. 浮动练习上(理解)
- 128. 浮动练习中(理解)
- 129. 浮动练习下(理解)

121. 网页布局方式(掌握)

什么是网页的布局方式？

- 网页的布局方式其实就是指浏览器是如何对网页中的元素进行排版的

标准流(文档流/普通流)排版方式

- 其实浏览器默认的排版方式就是标准流的排版方式
 - 在CSS中将元素分为三类, 分别是块级元素/行内元素/行内块级元素
 - 在标准流中有两种排版方式, 一种是垂直排版, 一种是水平排版
- 垂直排版**, 如果元素是块级元素, 那么就会垂直排版
- 水平排版**, 如果元素是行内元素/行内块级元素, 那么就会水平排版

122. 浮动流基本概念(掌握)

浮动流排版方式

- 浮动流是一种“半脱离标准流”的排版方式
- 浮动流只有一种排版方式, 就是**水平排版**. 它只能设置某个元素左对齐或者右对齐

注意点:

- 浮动流中没有居中对齐, 也就是没有 `center` 这个取值
- 在浮动流中是不可以使用 `margin: 0 auto;`

特点:

- 在浮动流中是不区分块级元素/行内元素/行内块级元素的
- 无论是块级元素/行内元素/行内块级元素都可以水平排版
- 在浮动流中无论是块级元素/行内元素/行内块级元素都可以设置宽高
- 综上所述, 浮动流中的元素和标准流中的行内块级元素很像

`float: left;` 要跟父元素的左边对齐

`float: right;` 要跟父元素的右边对齐

123. 浮动元素脱标(理解)

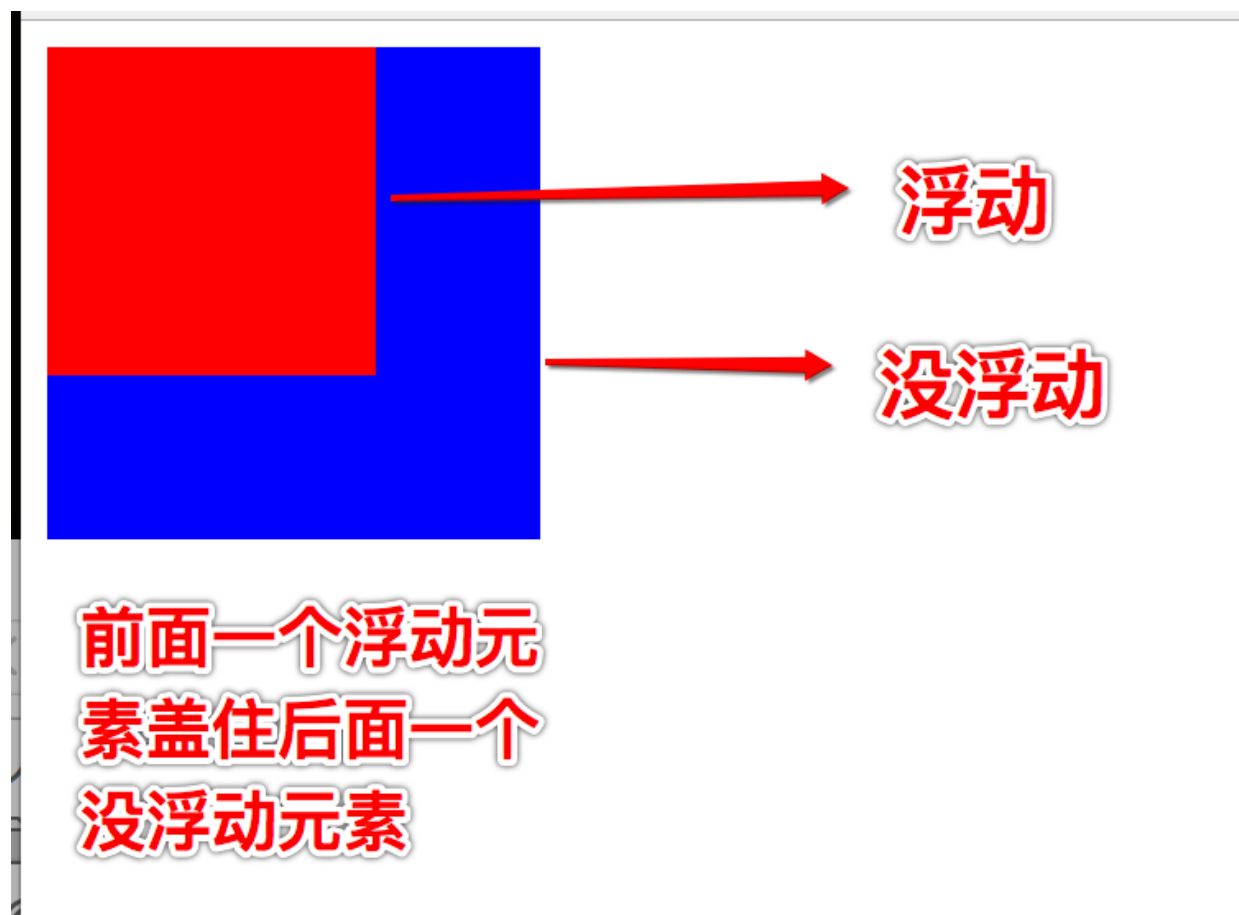
什么是浮动元素的脱标?

脱标: 脱离标准流

当某一个元素浮动之后, 那么这个元素看上去就像被从标准流中删除了一样, 这个就是浮动元素的脱标

浮动元素脱标之后会有什么影响?

如果前面一个元素浮动了, 而后面一个元素没有浮动, 那么这个时候前面一个元就会盖住后面一个元素



124. 浮动元素排序规则(理解)

浮动元素排序规则

- 相同方向上的浮动元素, 先浮动的元素会显示在前面, 后浮动的元素会显示在后面
- 不同方向上的浮动元素, 左浮动会找左浮动, 右浮动会找右浮动
- 浮动元素浮动之后的位置, 由浮动元素浮动之前在标准流中的位置来确定

125. 浮动元素贴靠现象(掌握)

什么是浮动元素贴靠现象?

- 如果父元素的宽度能够显示所有浮动元素, 那么浮动的元素会并排显示

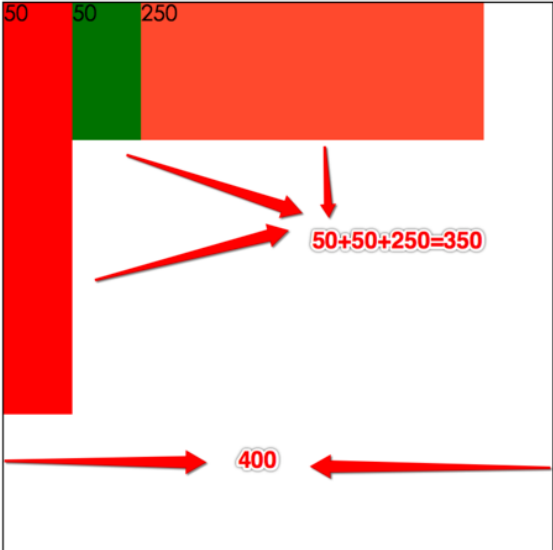


Diagram illustrating the float element wrapping phenomenon. A red vertical bar (width 50) and a green square (width 50) are shown. A red horizontal bar (width 250) is shown below them. Arrows indicate the total width of the floating elements is $50+50+250=350$, which is greater than the container width of 400. This causes the elements to wrap.

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <!--
      1. 什么是浮动元素贴靠现象?
      1. 如果父元素的宽度能够显示所有浮动元素, 那么浮动的元素会并排显示
      2. 如果父元素的宽度不能显示所有浮动元素, 那么会从最后一个元素开始往前贴靠
      3. 如果贴靠了前面所有浮动元素之后都不能显示, 最终会贴靠到父元素的左边或者右边
    -->
    <div class="father">...</div>
  </body>
</html>
```

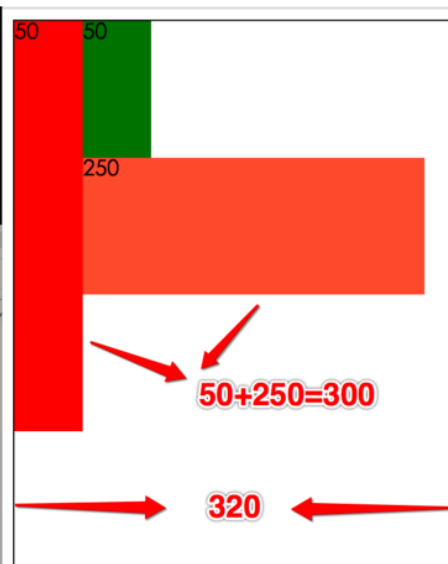
Styles

```
element.style {
}

.father {
  width: 400px;
  height: 400px;
  border: 1px solid #000;
}

div {
  display: block;
}
```

- 如果父元素的宽度不能显示所有浮动元素, 那么会从最后一个元素开始往前贴靠



Elements Console Sources Network Timeline Profiles Application Security Audits

```

<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <!--
      1.什么是浮动元素贴靠现象?
      1.如果父元素的宽度能够显示所有浮动元素,那么浮动的元素会并排显示
      2.如果父元素的宽度不能显示所有浮动元素,那么会从最后一个元开始往前贴靠
      3.如果贴靠了前面所有浮动元素之后都不能显示,最终会贴靠到父元素的左边或者右边
    -->
    <div class="father">...</div>
  </body>
</html>
  
```

html body div.father

Styles Computed Event Listeners >>

Filter :hov .cls +

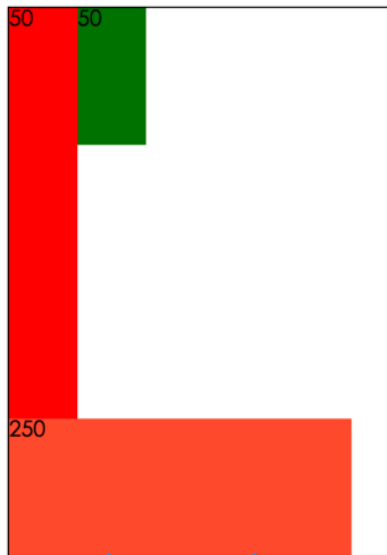
```

element.style {
}

.father {
  width: 320px;
  height: 400px;
  border: 1px solid #000;
}

div {
  display: block;
}
  
```

- 如果贴靠了前面所有浮动元素之后都不能显示,最终会贴靠到父元素的左边或者右边



Elements Console Sources Network Timeline Profiles Application Security Audits

```

<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <!--
      1.什么是浮动元素贴靠现象?
      1.如果父元素的宽度能够显示所有浮动元素,那么浮动的元素会并排显示
      2.如果父元素的宽度不能显示所有浮动元素,那么会从最后一个元开始往前贴靠
      3.如果贴靠了前面所有浮动元素之后都不能显示,最终会贴靠到父元素的左边或者右边
    -->
    <div class="father">...</div>
  </body>
</html>
  
```

html body div.father

Styles Computed Event Listeners >>

Filter :hov .cls +

```

element.style {
}

.father {
  width: 280px;
  height: 400px;
  border: 1px solid #000;
}

div {
  display: block;
}
  
```

126. 浮动元素字围现象(理解)

什么是浮动元素字围现象？

- 浮动元素不会挡住没有浮动元素中的文字, 没有浮动的文字会自动给浮动的元素让位置, 这个就是浮动元素字围现象

应用场景

- 图文混排

127. 浮动练习上(理解)

浮动只能做水平方向上的布局。

以后在开发的时候，如果发现水平方向上有多个盒子，那么就要想到使用浮动。

规律：

- 垂直方向上使用标准流布局
- 水平方向上布局使用浮动流布局

128. 浮动练习中(理解)

浏览器会尽量保证浮动元素在它的父元素中完整显示，如果不能完整显示会出现贴靠现象

开发中会把相关的样式写在一起，比如都属于 `header` 的内容，就写在一起。

129. 浮动练习下(理解)

在企业开发中什么时候使用标准流，什么时候使用浮动流？

垂直方向使用标准流，水平方向使用浮动流

拿到一个很复杂的界面该如何入手？

- 从上至下布局
- 从外向内布局
- 水平方向可以先划分一左一右再对左边或者右边进行进一步布局

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <style>
    body,div,dl,dt,dd,ul,ol,li,h1,h2,h3,h4,h5,h6,pre,code,form,fields
et,legend,input,textarea,p,blockquote,th,td{
      margin:0;padding:0
    }

    .header{
      width: 980px;
      height: 100px;
      /*background-color: red;*/
      margin: 0 auto;
    }
    .header .logo{
      width: 250px;
      height: 100px;
      background-color: pink;
      float: left;
    }
    .header .language{
      width: 150px;
      height: 50px;
      background-color: skyblue;
      float: right;
    }
    .header .nav{
      width: 650px;
      height: 50px;
      background-color: purple;
      float: right;
    }

    .content{
      width: 980px;
      height: 400px;
      /*background-color: green;*/
      margin: 0 auto;
      margin-top: 10px;
    }
    .content .aside{
      width: 320px;
      height: 400px;
      background-color: yellow;
      float: left;
    }
    .content .article{
      width: 650px;
      height: 400px;
      /*background-color: pink;*/
      float: right;
```

```

    }
    .content .article .articleTop{
        width: 650px;
        height: 350px;
        /*background-color: blue;*/
    }
    .content .article .articleTopLeft{
        width: 400px;
        height: 350px;
        /*background-color: red;*/
        float: left;
    }
    .content .article .articleTopLeft .new1{
        width: 400px;
        height: 200px;
        background-color: deepskyblue;
    }
    .content .article .articleTopLeft .new2{
        width: 400px;
        height: 140px;
        background-color: tomato;
        margin-top: 10px;
    }
    .content .article .articleTopRight{
        width: 240px;
        height: 350px;
        background-color: green;
        float: right;
    }
    .content .article .articleBottom{
        width: 650px;
        height: 40px;
        background-color: darkgoldenrod;
        margin-top: 10px;
    }
    .footer{
        width: 980px;
        height: 40px;
        background-color: tomato;
        margin: 0 auto;
        margin-top: 10px;
    }
</style>
</head>
<body>
    <div class="header">
        <div class="logo"></div>
        <div class="language"></div>
        <div class="nav"></div>
    </div>
    <div class="content">
        <div class="aside"></div>

```

```
<div class="article">
  <div class="articleTop">
    <div class="articleTopLeft">
      <div class="new1"></div>
      <div class="new2"></div>
    </div>
    <div class="articleTopRight"></div>
  </div>
  <div class="articleBottom"></div>
</div>
</div>
<div class="footer"></div>
</body>
</html>
```

完成于 201811051021