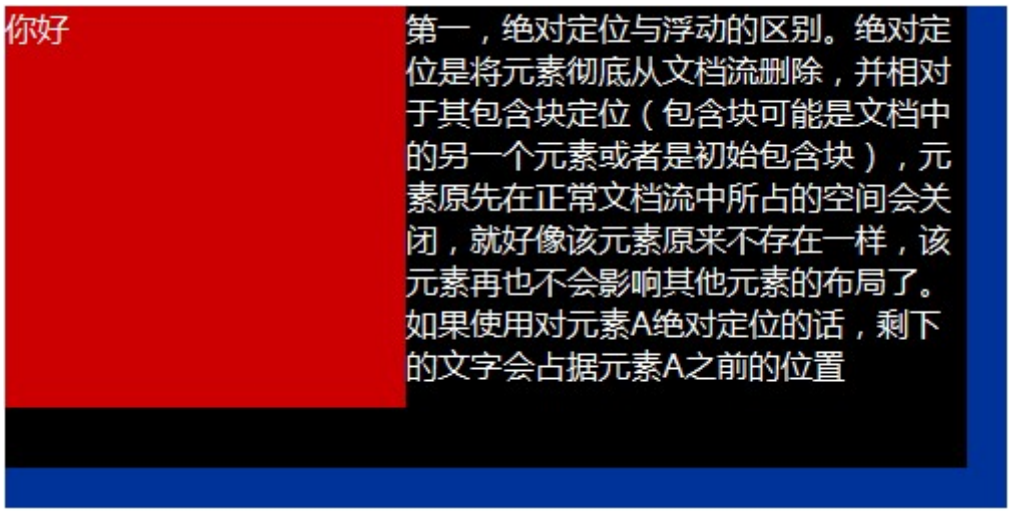
浮动（float）

为什么需要浮动

- 最初浮动是为了像word有文字环绕，图片嵌入的功能出现的



- 让多个块级元素水平排列成一行

- 实现两个块级盒子的左对齐和右对齐

浮动的本质

1. 浮动的包裹性

   抛开浮动的破坏性先不谈，浮动就是个带有方位的display:inline-block的属性。
   display:inline-block 某种意义上的作用就是包裹，浮动也有类似的效果

   但是 float和display:inline-block 无法完全等同

   原因：

- float可以自定义方向（从左往右或从右往左） inline-block只有从左到右一个水平排列的方向
- float会使周围文字环绕元素 inline-block不会

2. 浮动的破坏性

flex破坏性指的是，父元素高度塌陷 。

把一张图片放在div里显示：

```
<div class="outer">
    <img src="./images/2.jpg" width="150px" height="100px">x
</div>
```

```
.outer{
    width: 400px;
    background-color: lightblue;
}
```

这时候如果给图片加上float:left ,父元素蓝色框就只有x占据的宽度

```
.outer{
    width: 400px;
    background-color: lightblue;
    float:left;
}
```

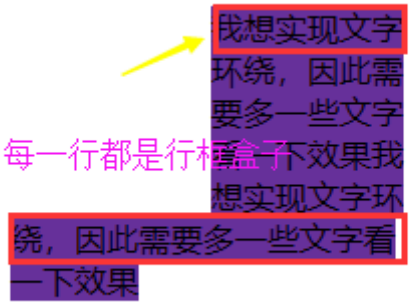
这时候如果把x去掉 那么父元素（蓝色框） 就直接没了

探究float实现文字环绕：

css设计师在设计float属性的时候定义了float 两个特性：

- “破坏文档流”，使父元素高度塌陷
- 禁止行框盒子与浮动元素进行重叠

所以在文字环绕图片的过程中 第一步破坏文档流使父元素高度塌陷 这时候文字标签跟图片在一个水平线上。然而还没有出现环绕的效果 这时候 第二个特征也就是float元素禁止与行框盒子发生重叠（有人认为这是float与绝对定位最大的区别之一）



float 使用

```
float:left  左浮动

float:right  右浮动
```

float:left 实现块级元素水平排列

```
<!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>
        *{
            margin:0;
            padding: 0;
        }
        .box{
            width: 1000px;
            height: 300px;
            background: gold;
            margin: 20px;
        }
        .box1{
            /* width:60px;
            height: 60px; */
            background: red;
```

```
            /* background: rgba(255, 0, 0, 0.363); */
            /* float: left; */


        }
        .box2{
            width: 120px;
            height:80px;
            background: gray;
            border: 1px solid pink;
            /* float:right */
            float: left;
        }
        .a{
            /* width: 70px;
            height: 70px;
            border: 1px solid blue;
            background: pink; */
            clear: right;
        }
    </style>
</head>
<body>
    <div class="box">
        <div class="box1">asaaaa</div>
        <div class="box2"></div>
        <div class="box2"></div>
        <div class="box2"></div>
        <div class="box2"></div>
        <div class="box2"></div>
        <div class="box2"></div>


    </div>
</body>
</html>
```

清除浮动

```
clear:left(清除左浮动)/right(清除右浮动)/both(清除左浮动+右浮动)
```

哪边不允许有浮动元素，clear就是对应方向的值，两边都不允许就是`both`

方案1：给需要清浮动的元素加clear属性

```
  .box4{
            width: 120px;
            height:80px;
            background: blue;
            border: 1px solid pink;
            clear: left;
        }
```

方案2：

html:

```
<div class="box">
        <div class="box1">asaaaa</div>
        <div class="box2"></div>

        <div class="box2">4</div>
        <div class="box2">5</div>
        <!-- <div class="box2" style="height: 180px;">6</div> -->
        <div class="blank"></div>
        <div class="box4"></div>
    </div>
```

css:

```
 .blank{
            clear: left;
        }
```

方案3：使用伪元素清除浮动

css:

```
.boxes:after {
            content: '.';
            height: 0;
            display: block;
            clear: both;
        }
```

html:

```
<div class="box">
        <div class="boxes clearfix">
            <div class="box1">asaaaa</div>
            <div class="box2"></div>

            <div class="box2">4</div>
            <div class="box2 blankdiv">5</div>
            <!-- <div class="box2" style="height: 180px;">6</div> -->
            <div class="blank"></div>
        </div>
        <div class="box4"></div>

    </div>
```

方案4：overflow 清除浮动

html：

```
<div class="box">
        <div class="boxes clearfix">
            <div class="box1">asaaaa</div>
            <div class="box2"></div>

            <div class="box2">4</div>
            <div class="box2 blankdiv">5</div>
            <!-- <div class="box2" style="height: 180px;">6</div> -->
            <div class="blank"></div>
        </div>
        <div class="box4"></div>

    </div>
```

css:

```
   .boxes{
            overflow: auto;
        }
```

其他内容

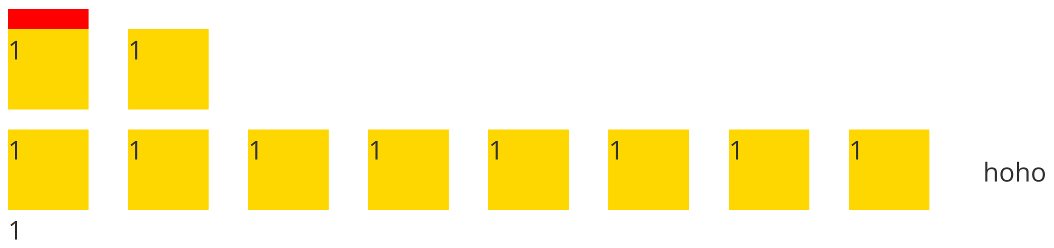
这里有很多文字，且要超出一行且要超出一行且要超出一行 更多

这里有很多文字，且要超出一更多行且要超出一行且要超出一行

浮动元素参考的是离他最近的行框盒子

下面这段代码 有什么效果？为什么

```
<div>
    <div style="width:40px;height:40px;background:gold;float:left;margin-
left:20px;margin-top:10px">1</div>
    <div style="width:40px;height:40px;background:gold;float:left;margin-
left:20px;margin-top:10px">1</div>
    <div style="width:40px;height:40px;background:gold;float:left;margin-
left:20px;margin-top:10px;clear:left">1</div>
    <div style="width:40px;height:40px;background:gold;float:left;margin-
left:20px;margin-top:10px">1</div>
    <div style="width:40px;height:40px;background:gold;float:left;margin-
left:20px;margin-top:10px">1</div>
    <div style="width:40px;height:40px;background:gold;float:left;margin-
left:20px;margin-top:10px">1</div>
    <div style="width:40px;height:40px;background:gold;float:left;margin-
left:20px;margin-top:10px">1</div>
    <div style="width:40px;height:40px;background:gold;float:left;margin-
left:20px;margin-top:10px">1</div>
    <div style="width:40px;height:40px;background:gold;float:left;margin-
left:20px;margin-top:10px">1</div>
    <div style="width:40px;height:40px;background:gold;float:left;margin-
left:20px;margin-top:10px">1</div>
</div>
```



效果如上面的实例所示，会将里面的小方块分成两行

原因：在行内样式中，第三个元素被赋予clear:left (清除左浮动)，这个时候根据标准 他不能跟前面浮动的元素相邻，也就是不能跟前面的元素做朋友了 要分开去下一行 只能换行咯

但是标准没有规定不能跟下一个元素做朋友，所以后面的元素依然可以依据左浮动的规则继续靠在一起.

因为第一行和第二行都保持了浮动的特性，所以父级元素的高度是0，再最后面插入文本的时候，会飘到第一行去,如果是文字的话因为浮动元素不能和行框元素发生重叠，所以文字会出现在最右侧 而且最后定义的盒子里面的字会跟盒子本体分离

因为盒子占了空间所以hoho被挤到下一行的右边去了

看完了上面的例子，再来简单了解下clear的四个属性，分别是none(默认，就是没有)，left（清左浮动），right（清右浮动）以及我们最常用的both(全清)。作者这里给出了clear的基本使用方式就是clear:both。left和right属性根本没有软用，让CSS自己判断就好了，因为不可能有一个元素既是left又是right浮动的，因此无需考虑是清左浮动还是右浮动，全清就完事了。

由于clear只能确保和前面的元素发生关系，因此我们最常使用的是after伪类清浮动，而不是before，因为before生成的元素根本没法和后面的元素交流clear的事情。最后我们放上我们最喜欢使用的after伪类清浮动的方法，注意clear属性只有块级元素才有效，而伪类的默认属性是内联值，不要忘了display:block声明。

通过BFC 彻底清除浮动

BFC是block formatting context的缩写，中文名为“块级格式化上下文”。前面也多次提到了这个听起来十分拗口的属性，那么CSS设计这个属性的初衷是什么呢？

记住一句话：拥有BFC特性的元素会形成类似“结界”的封闭内部空间，该元素内部的元素以及该元素本身都不会影响外部元素的表现。要理解这句话，还得通过一些例子来证明，在举例证明之前，我们必须得知道一件事，就是CSS规定了哪些属性可以生成BFC属性，常见的情况如下：

- flaot 不为none的元素
- overflow为 auto scroll 或hideen 的元素
- display的值为inline-block，table-cell或table-caption的元素
- position的值不为relative和static

我的父元素有BFC,我是左浮动 我的父元素有BFC,我是右浮动

我的父元素有BFC,我是左浮动 我的父元素有BFC,我是右浮动