

A N E X C E R P T F R O M

The Hound of the Baskervilles

HOLMES STRETCHED OUT HIS HAND FOR THE MANUSCRIPT AND FLATTENED IT UPON HIS KNEE. "YOU WILL OBSERVE, WATSON, the alternative use of the long s and the short. It is one of several indications which enabled me to fix the date." At the head was written: "Baskerville Hall," and below in large, scrawling figures: "1742."

"It appears to be a statement of some sort."

"Yes—it is a statement of a certain legend which runs in the Baskerville family."

"Of the origin of the Hound of the Baskervilles there have been many statements, yet as I come in a direct line from Hugo Baskerville, and as I had the story from my father..."

When Dr. Mortimer had finished reading this singular narrative he pushed his spectacles up on his forehead and stared across at Mr. Sherlock Holmes.

float

让人又爱又恨的~浮动

float history

在很久很久很久以前（多么熟悉的旋律），浮动就已经出现了。拿来干嘛的呢？

one：最初仅仅为了实现传统报刊出版物上边的 **文字环绕** 效果。

two：让上下排列的元素不在堆叠，变成左右的排列，从而实现布局的很多的效果，例如分栏（不乖乖的呆着我要挣脱束缚 --- **文档流**，还tm有这种操作，我什么操作要你管）。

tip：其实简单的来说就是在原有的空间支持的情况下，浮起来脱离常规的文档流，那么其后边的元素就会在空间支持的情况下向上提升与其他元素平起平坐。这里的话遵循一个原则就是能浪到哪就是哪，除非有东西挡住了或者说到了父元素的边界。要不然我是不会停止的（毕竟挣脱了束缚）。

文字环绕

float的最初目的

example

```
<style>
  .example1 span{float: left;background: red;color: #fff;}
  .example2 span{float: left;background: red;color: #fff;}
</style>
<body>
  <!-- 通过理解行内元素换行产生空字符来理解 -->
  <!-- 1. 去空格化 -->
  <!-- 2. 会让元素block化 添加高度宽度有效果 自行测试 忘记说了 -->
  <div class="example1">

    <span>1</span>
```

```
<span>2</span>
<span>3</span>
</div>

<!-- html解析的回车是一个空字符 本质是  span浮动之后，空字符去哪里了。最后面 -->
<!-- 那么其实就是文字环绕的效果 -->
<div class="example2">
  <span>1</span>&nbsp;<span>2</span>&nbsp;<span>3</span>
</div>
</body>
```

tip：实在不行，请脑补皇帝后宫的故事，香艳的乾清宫，姿色万分的嫔妃，皇帝也是辛苦为了做到雨露均沾，从原来的厮守交欢独宠一人，到现在的百花齐放。不得以使用了邪恶的浮动。。。此处省略1w字。但是要注意一个点浮动非图片的时候一样的效果千万不要以为只有图片可以，若不是非图片元素记得给宽高，因为图片是有宽高的。

float现代意义

- 分列布局
- 导航栏、商品列表等等
- 注意点：浮动的时候可能会被高度不一的元素挡住，注意调整元素的高度到合适的位置，已经父元素的空间是否足够。请记住能用浮动办到的事情请不要使用定位，因为不利于页面的后期维护，迫不得已才去使用定位。

包裹与浮动

浮动的元素脱离了文档流，必然会导致父元素高度丢失。因此不会包裹原有文档流中的子元素。这种情况不是我们想要的。

原本的元素在正常文档流中是会受父元素限制（包裹）里边的改变应该只是在内部不会去影响到外部（内部翻江倒海，外部风平浪静。或者也可以是你和一个妹妹玩角色扮演。“你叫啊，叫破喉咙也不会有人来救你”）---块级格式化上下文。当然这个不是重点。

当元素被添加了浮动之后包裹就消失父元素塌陷，并且元素还会按照浮动的方向去移动。遭到了破坏。（想那个画的图）。没办法原本的作用就是环绕，所以忍了吧。

浮动元素的margin:auto失效，没有为什么，本来就相互违背为什么要生效。

解决

清除浮动 回归文档流

明白清除浮动的目的带来的影响，浮动还是存在的，只是浮动元素对周围元素的影响以及父元素对其的包裹性。

- 插入元素添加属性clear：left / right / both。注意点
 1. 插入的元素必须是块级属性的元素。clear会令插入元素的margin-top/bottom失效。
 2. 伪元素的使用在于低版本的IE不支持，并且一定是给父元素添加（也一定是有浮动的父元素）

```
/* 都可以 */
.clearfix::after{
  content: '';
  display: block;
  overflow: hidden;
  clear: both;
}

/* 推荐 代码少 */
.clearfix::after{
  content: '';
  display: table;
  clear: both;
}

/* IE8- */
.clearfix{
  display: table;
  clear: both;
}

/* IE7 */
.clearfix{*zoom: 1;}
```

- 对父元素使用overflow: hidden
- 解决包裹问题对父元素使用浮动（没有清除的效果，请区别对待）

last

有什么问题请留言！ *Good night~ (^_^)*。