

网页编程基础 CSS3基础

WebBasic CSS3BASIC

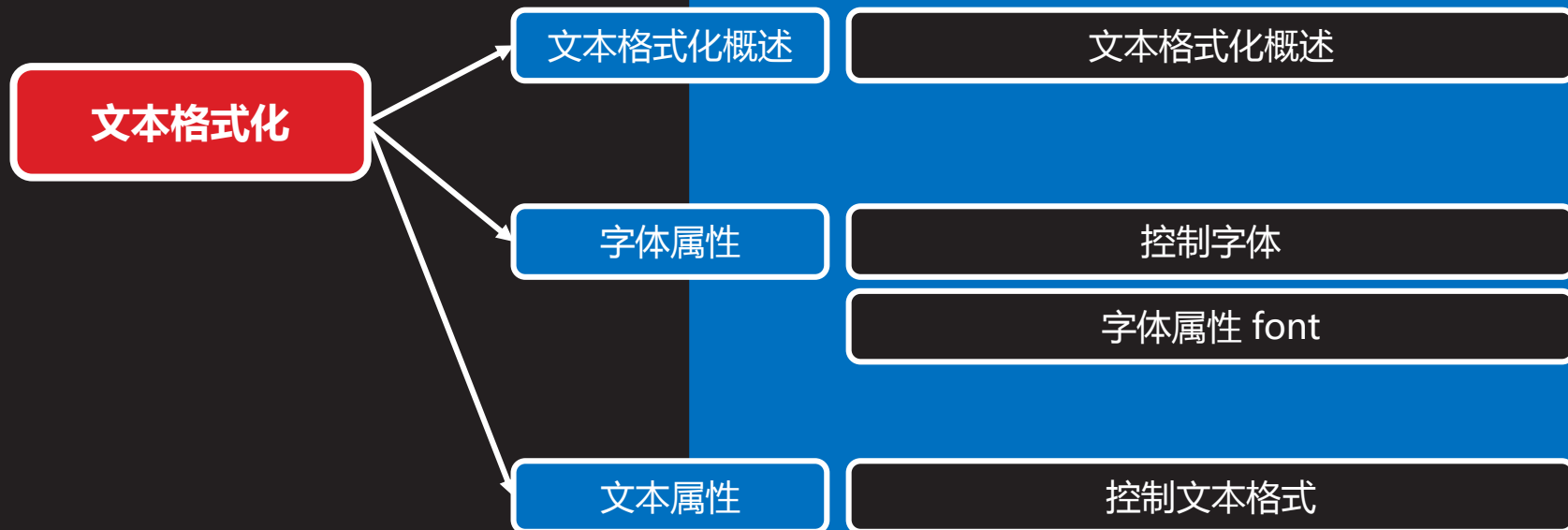
Unit03

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	文本格式化
	10:30 ~ 11:20	
	11:30 ~ 12:00	表格
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	浮动
	16:00 ~ 16:50	
	17:00 ~ 17:30	总结和答疑



文本格式化



文本格式化概述



文本格式化概述

- 主要包括
 - 控制字体：如字体大小、字体加粗、字体系列等
 - 设置文本格式：如文本颜色、文本排列和文本缩进等
- 建议使用文本格式化相关的样式，取代加粗（）、倾斜（<i>）等 HTML 元素



字体属性



控制字体

- 指定字体
 - font-family : value1,value2;
- 字体大小
 - font-size : value;
- 字体加粗
 - font-weight : normal/bold/value;
- 字体样式
 - font-style : normal/italic;
- 小型大写字母显示
 - font-variant : normal/small-caps;



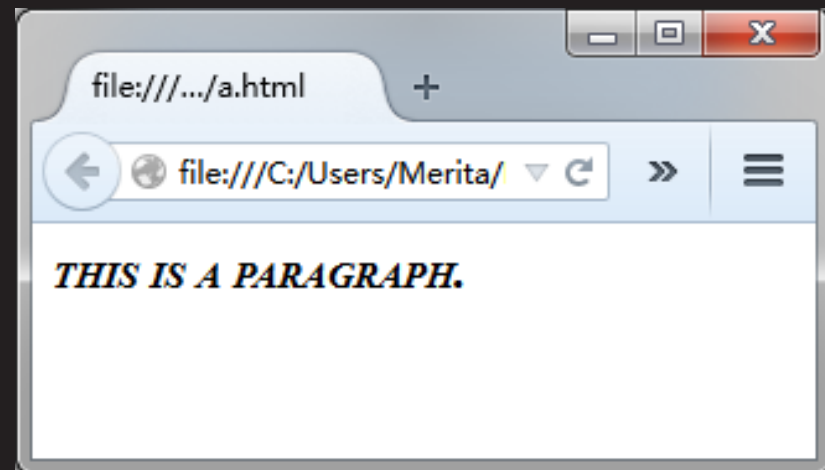
控制字体（续1）

样式表中：

```
p {
    font-family : Times, 'New York', serif ;
    font-size : 14pt ;
    font-weight : bold ;
    font-style : italic;
    font-variant : small-caps;
}
```

html 文档中：

<p>this is a paragraph.</p>



字体属性 font

- 字体属性 font 用于把所有针对字体的属性设置在一个声明中
- 为简写属性，包含6个值，可以按顺序设置
 - font : font-style font-variant font-weight font-size font-family;
 - 不设置的值，则使用默认设置



文本属性



控制文本格式

- 文本颜色 `color : value;`
- 文本排列 `text-align : left/right/center;`
- 文字修饰 `text-decoration : none/underline;`
- 行高 `line-height : value;`
- 首行文本缩进 `text-indent : value;`
- 文本阴影
 - `text-shadow : h-shadow v-shadow blur color;`



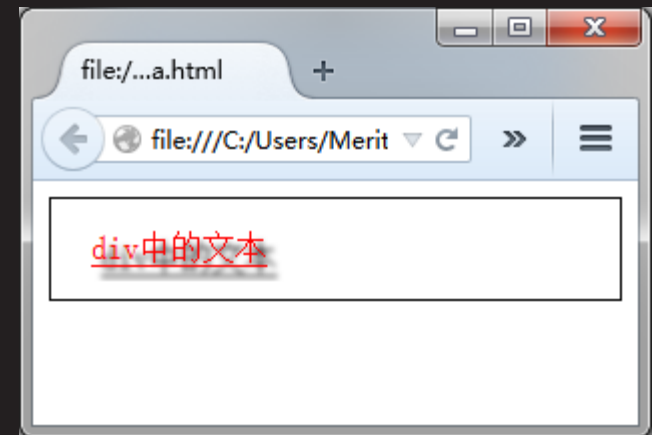
控制文本格式（续1）

```
div {
    color : #FF0000 ;
    text-align : left ;
    text-decoration : underline ;
    height:50px;
    text-indent : 20px ;
    border : 1px solid black;
    text-shadow : 5px 5px 2px black;
}
```



```
div {
    color : #FF0000 ;
    text-align : left ;
    text-decoration : underline ;
    height:50px;
    text-indent : 20px ;
    border:1px solid black;
    text-shadow : 5px 5px 2px black;
    line-height:50px;
}
```

<div>div中的文本</div>



文本格式化

- 【参见 COOKBOOK】
 - 使用文本格式化属性为页面元素定义样式



表格

表格

表格常用样式属性

表格常用样式属性

垂直方向对齐

表格特有样式属性

边框合并 border-collapse

边框边距 border-spacing

标题位置 caption-side

显示规则 table-layout

表格常用样式属性

表格常用样式属性

- 边距属性：padding
- 尺寸属性：width、height
 - 设置表格或者单元格的尺寸
- 文本格式化属性
- 背景属性：设置表格或者单元格的背景色或者背景图像
- border 属性：设置表格的边框



表格常用样式属性（续1）

CSS 样式

table

```
{
  width:200px;
  border:1px solid black;
}
```

td

```
{
  height:50px;
  border:3px solid red;
}
```

html 文档

```
<table>
  <tr>
    <td>aa</td>
    <td>bb</td>
  </tr>
  <tr>
    <td>cc</td>
    <td>dd</td>
  </tr>
</table>
```



aa	bb
cc	dd

垂直方向对齐

- vertical-align 属性
 - 在表单元格中，设置单元格框中的单元格内容的对齐方式
- 取值
 - vertical-align : top/middle/bottom;



表格特有样式属性



边框合并 border-collapse

- 如果设置了单元格的边框，相邻单元格的边框会单独显示，类似于双线边框
- border-collapse 属性：合并相邻的边框
 - 设置是否将表格边框合并为单一边框
- border-collapse : separate/collapse;

```
table {border : 2px dotted black;}
td {
```

```
    width : 200px ;
    height : 50px;
    border : 3px solid gray;
```

```
}
```

```
table.separate {border-collapse : separate;}
```

```
table.collapse {border-collapse : collapse;}
```




边框合并 border-collapse (续1)

```
<table class="separate">
  <tr>
    <td>第1行第1列</td>
    <td>第1行第2列</td>
  </tr>
</table>
```



第1行第1列	第1行第2列
--------	--------

```
<table class="collapse">
  <tr>
    <td>第1行第1列</td>
    <td>第1行第2列</td>
  </tr>
</table>
```



第1行第1列	第1行第2列
--------	--------

边框边距 border-spacing

- border-spacing 属性
 - 设置相邻单元格的边框间的距离
 - 仅限于分隔单元格边框，即border-collapse 属性为 separate 值的情况下，也称为边框分离模式
- 取值为长度值，可以为该属性指定一个或者两个值
 - 指定一个值：该值同时应用于水平和垂直间距
 - 指定两个值：第一个值指定水平间距，第二个值指定垂直间距，且两个值之间用空格隔开



边框边距 border-spacing (续1)

```
table
{
    border:2px dotted gray;
    border-spacing:10px 20px;
}
td
{
    background-color:#f0f0f0;
    width:200px;
    height:50px;
    border-collapse:separate;
    border:1px solid black;
}
```



标题位置 caption-side

- caption-side 属性设置表格标题的位置，指定表标题相对于表框的放置位置
- 可取值
 - top：表格标题定位在表格之上，为默认值
 - bottom：表格标题定位在表格之下



标题位置 caption-side (续1)

```
td
{
    background-color:#f0f0f0;
    width:150px;
    height:50px;
    border-collapse:separate;
    border:1px solid black;
}
caption
{
    border:2px dotted gray;
}
caption.bottom
{
    caption-side:bottom;
    height:50px;
}
```

第1行第1列	第1行第2列
第2行第1列	第2行第2列
表格标题	



显示规则 table-layout

- table-layout 属性用来帮助浏览器如何显示或者布局一张表，即用来设置显示表格单元格、行、列的算法规则
- 可取值
 - auto：列宽度由单元格内容设定，为默认值，即自动表格布局
 - fixed：列宽由表格宽度和列宽度设定，即固定表格布局



显示规则 table-layout (续1)


CSS 样式

```
table { width:400px;}
td { border:1px solid black;}
table.one { table-layout: auto;}
```

1000000000000000000	100000000	100
---------------------	-----------	-----

html 文档

```
<table class="one">
  <tr>
    <td width="20%">1000000000000000000</td>
    <td width="40%">100000000</td>
    <td width="40%">100</td>
  </tr>
</table>
```



显示规则 table-layout (续2)


CSS 样式

```
table { width:400px;}  
td { border:1px solid black;}  
table.two { table-layout: fixed;}
```

10000000000000000000	1000000000	100
----------------------	------------	-----

html 文档

```
<table class="two">  
  <tr>  
    <td width="20%">10000000000000000000</td>  
    <td width="40%">100000000</td>  
    <td width="40%">100</td>  
  </tr>  
</table>
```



显示规则 table-layout (续3)

- 自动表格布局
 - 单元格的大小会适应内容的大小
 - 在表格复杂时会比较慢
 - 适用于不知道每一列的确定大小
- 固定表格布局
 - 取决于表格宽度、列宽度、表格边框宽度、单元格间距，而与单元格的内容无关
 - 会加速表的显示，因为浏览器在接收到第一行后就可以显示表格
- 固定布局算法比较快，但是不太灵活
- 自动算法比较慢，不过更能反映传统的 HTML 表



表格样式的应用

- 【参见 COOKBOOK】
 - 使用表格样式为页面元素定义样式



浮动

浮动

定位概述

定位概述

普通流定位

浮动定位

浮动概述

浮动定位

float 属性

clear 属性

float 与 overflow

定位概述



定位概述

- 定位：定义元素框相对于其正常位置应该出现的位置，或者相对于父元素、另一个元素甚至浏览器窗口本身的位置
- 普通流定位
- 浮动定位
- 相对定位
- 绝对定位



普通流定位

- 页面中的块级元素框从上到下一个接一个地排列
 - 每一个块级元素都会出现在一个新行中（比如 `<p>` 元素、`<div>` 元素）
 - 元素框之间的垂直距离是由框的垂直外边距计算出来的
- 内联元素将在一行中从左到右排列水平布置
 - 不需要从新行开始
 - 可以使用水平内边距、边框和外边距调整它们的间距



浮动定位



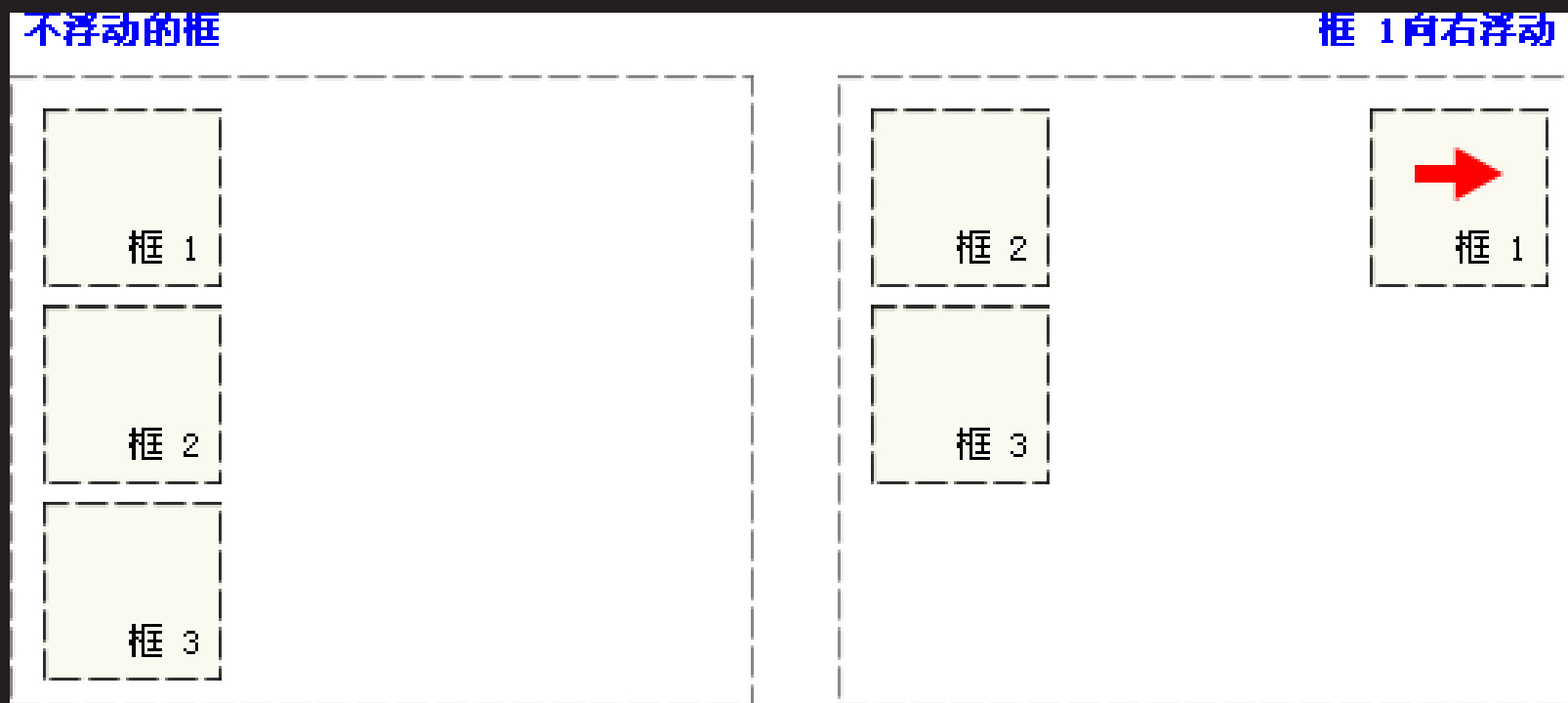
浮动概述

- 浮动定位是指
 - 将元素排除在普通流之外，即元素将脱离标准文档流
 - 元素将不在页面占用空间
 - 将浮动元素放置在包含框的左边或者右边
 - 浮动元素依旧位于包含框之内
- 浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止
 - 经常使用它来实现特殊的定位效果



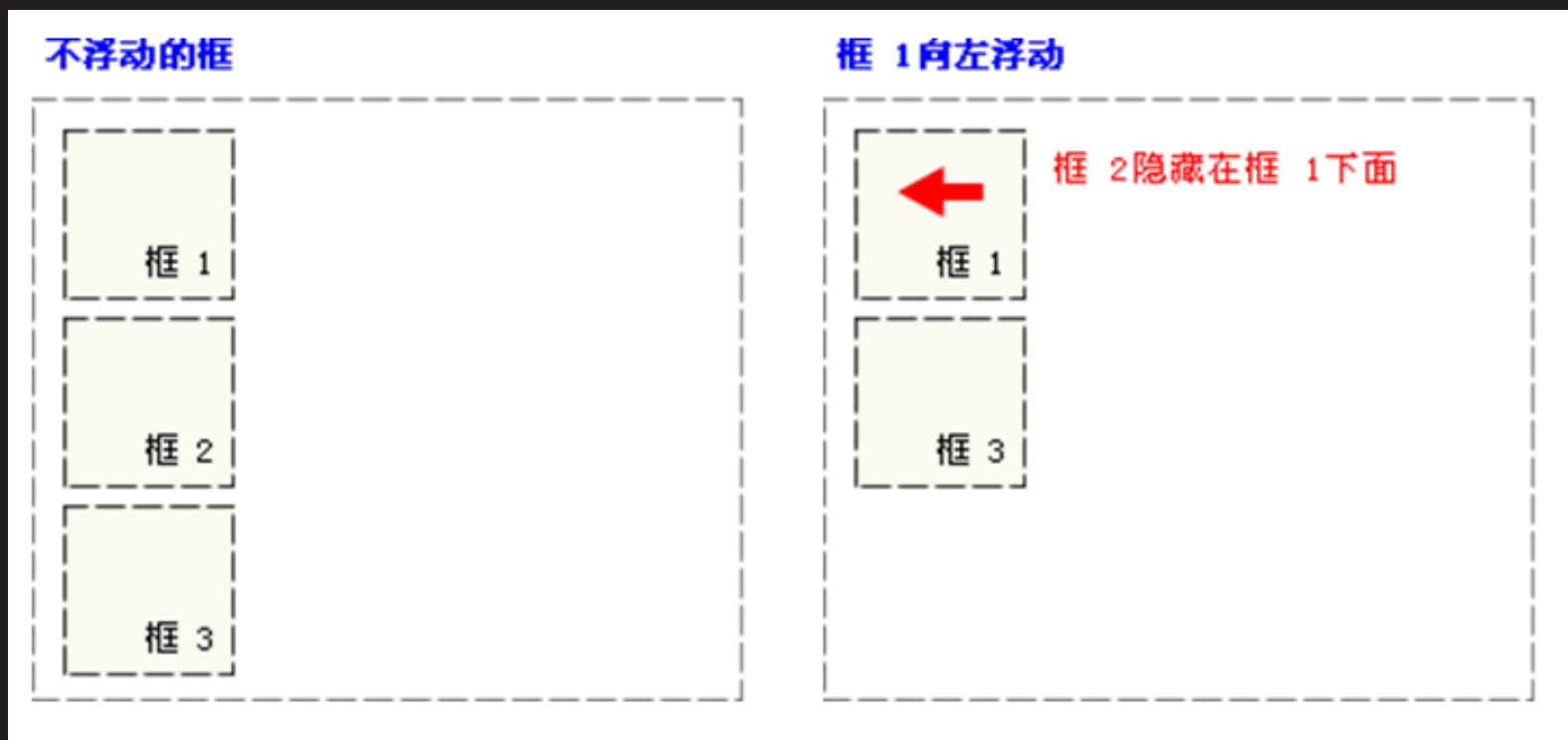
浮动定位

- 包含框中有三个元素框，如果把框 1 向右浮动，则它脱离文档流并且向右移动，直到它的右边缘碰到包含框的右边缘



浮动定位（续1）

- 当框 1 向左浮动时，它脱离文档流并且向左移动，直到它的左边缘碰到包含框的左边缘
- 因为框 1 不再处于文档流中，所以它不占据空间，实际上覆盖住了框 2，使框 2 从视图中消失

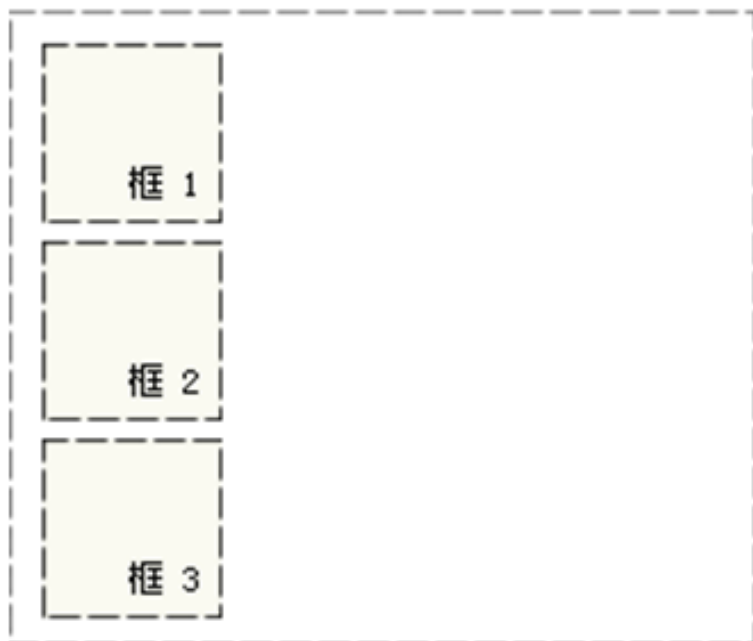


浮动定位（续2）

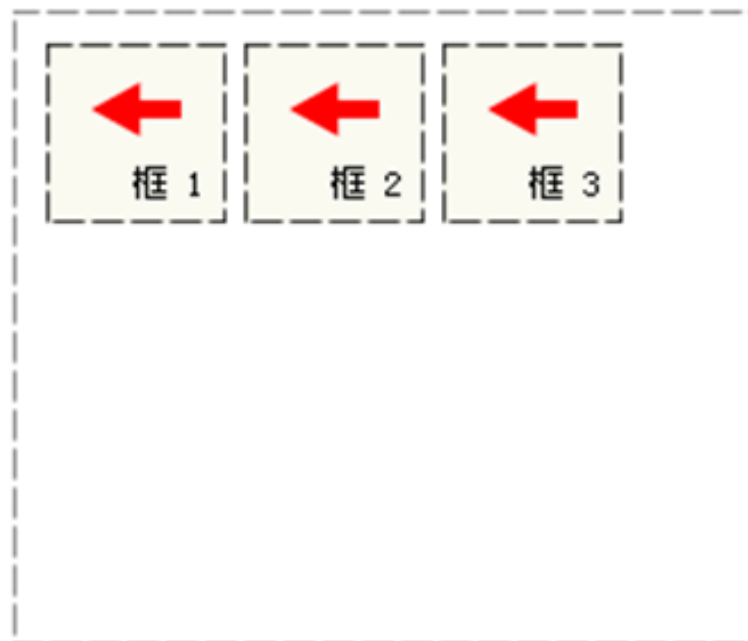
- 如果把所有三个框都向左移动，那么框 1 向左浮动直到碰到包含框，另外两个框向左浮动直到碰到前一个浮动框：三个框在同一行上显示

在需要设置多个块级元素同行排列的情况下会非常有用

不浮动的框



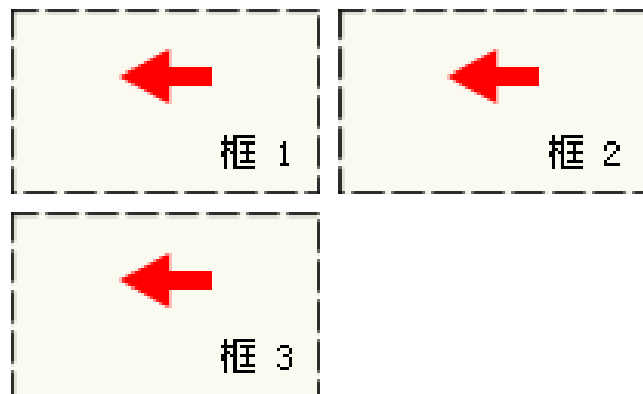
所有三个框向左浮动



浮动定位（续3）

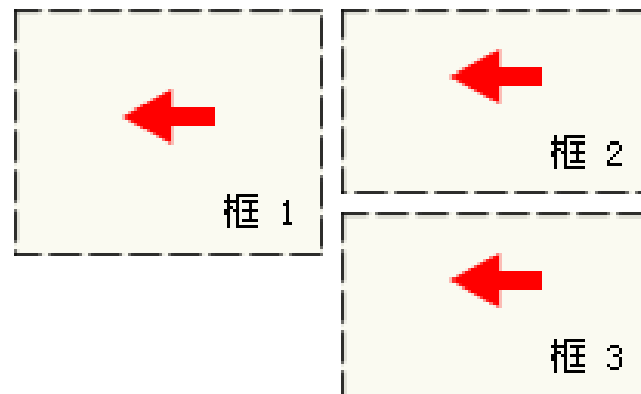
- 如果包含框太窄，那么其它浮动块会自动向下移动，直到有足够的空间（如左图所示）
- 如果浮动元素的高度不同，那么当它们向下移动时可能被其它浮动元素“卡住”（如右图所示）

框 1 向左浮动



框 3 下降

所有三个框向左浮动



框 3 被框 1 “卡住了”

浮动定位（续4）

- 浮动元素的外边缘不会超过其父元素的内边缘
- 浮动元素不会互相重叠
- 浮动元素不会上下浮动



float 属性

- 如果需要设置框浮动在包含框的左边或者右边，可以通过 float 属性来实现
- float 属性定义元素在哪个方向浮动
 - 在 CSS 中，任何元素都可以浮动
- float : none/left/right ;

CSS 样式：

```
div
{
    border : 1px solid black;
    width : 100px;
    height : 50px;
    float : left;
}
```

HTML 文档：

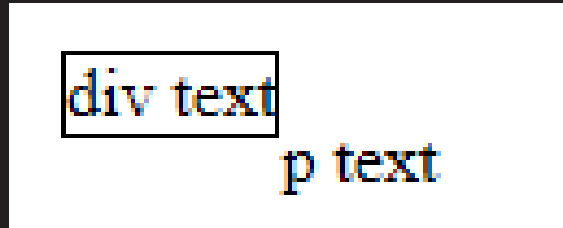
```
<div>1</div>
<div>2</div>
<div>3</div>
```



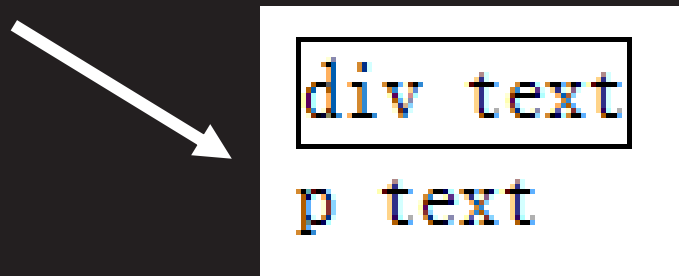
clear 属性

- clear 属性用于清除浮动所带来的影响
- clear : none/left/right/both ;
 - 定义了元素的哪边上不允许出现浮动元素

```
<div style="float:left;border:1px solid black;">div text</div>
<p>p text</p>
```



```
<div style="float:left;border:1px solid black;">div text</div>
<p style="clear:left;">p text</p>
```



float 与 overflow

- 包含框内的元素被应用了float之后，包含框的高度会发生变化

CSS 样式：

```
div.parent
{
    width:150px;
    border:1px solid black;
}
div.parent div
{
    border:1px solid red;
    background-color:#fdedd2;
    float:left;
    width:50px;
    height:50px;
}
```

HTML 文档：

```
<div class="parent">
    <div></div>
    <div></div>
</div>
```



包含框的高度不再随着内容自适应

float 与 overflow (续1)

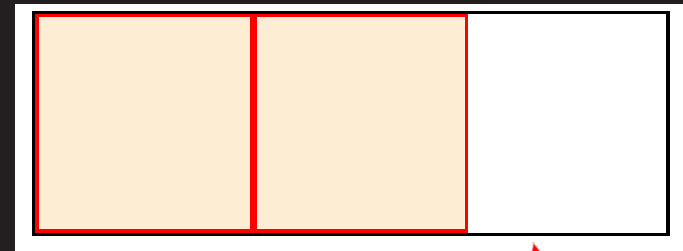
- 设置包含框内的 overflow 属性

CSS 样式 :

```
div.parent
{
    width:150px;
    border:1px solid black;
    overflow:hidden;
}
div.parent div
{
    border:1px solid red;
    background-color:#fdedd2;
    float:left;
    width:50px;
    height:50px;
}
```

HTML 文档 :

```
<div class="parent">
    <div></div>
    <div></div>
</div>
```



包含框的高度正常

浮动属性的应用

- 【参见 COOKBOOK】
 - 使用浮动属性为页面元素定义样式



总结和答疑

