

第3讲

CSS

层叠样式表

信息学院 孙辉



内容概要

- CSS的概念与实例
- CSS基本语法
- 选择器
- 样式
- CSS深入



CSS 的概念

- 层叠样式表
- 定义如何显示HTML元素，样式通常存储在样式表中
- 解决内容与表现分离的问题
- 外部样式表可以极大提高工作效率
- 外部样式表通常存储在 CSS 文件中。

CSS实例1

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> HTML Sample </title>
5     <style type="text/css">
6       <! li {font-size: 24px; color: blue} >
7       <! a {font-size: 24px; display:inline;} >
8     </style>
9   </head>
10  <body>
11    <p>I study in
12      <a href="http://www.ruc.edu.cn">
13        Renmin Univ
14      </a>
15    </p>
16    <p>Let's begin to learn HTML!</p>
17    <ul>
18      <li><font size="3" color="red">这是第1项</font></li>
19      <li>这是第2项</li>
20    </ul>
21    <br/>
22  </body>
23 </html>
```

CSS实例2

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
5     <title> HTML Sample </title>
6     <link type="text/css" rel="stylesheet" href="demo2.css" />
7   </head>
8   <body>
9     <p>I study in
10      <a href="http://www.ruc.edu.cn">
11        Renmin Univ
12      </a>
13    </p>
14    <p>Let's begin to learn HTML!</p>
15    <ul>
16      <li><font size="3" color="red">这是第:
17      <li>这是第2项</li>
18      <li style="display:none">ITEM 3</li>
19    </ul>
20    <br/>
21  </body>
22 </html>
```

```
1 li
2 {
3   font-size: 24px;
4   color: blue
5 }
6 a
7 {
8   font-size: 24px;
9   display:inline;
10 }
11
```

内容概要

- CSS的概念与实例

- CSS基本语法

- 选择器

- 样式

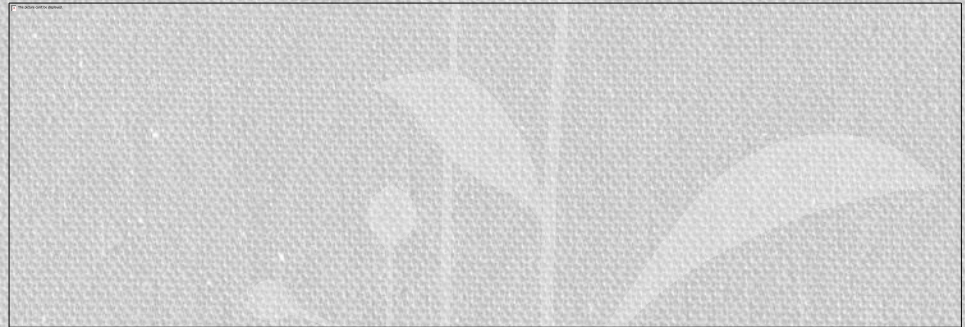
- CSS深入



CSS 语法

▶ 基本语法

- CSS 规则由两个主要的部分构成：选择器，以及一条或多条声明。
- `selector {declaration1; declaration2; ... declarationN }`
- **选择器**通常是您需要改变样式的 HTML 元素。
- 每条声明由一个属性和一个值组成。
 - `selector {property: value}`
 - `h1 {color:red; font-size:14px;}`
 - ```
p {
 text-align: center;
 color: black;
 font-family: arial;
}
```



在CSS中，注释以“/\*”开始，以“\*/”结束

# 内容概要

- CSS的概念与实例
- CSS基本语法
- 选择器
- 样式





# 简单选择器

- `html-tag-name`
  - HTML 代码中找到的任何标记
  - 前面的例子中用的就是这种选择器

```
html-tag-name {
 css-property-key-1: css-value-1;
 css-property-key-2: css-value-2;
}
```

- 如果我们不想把所有的`<li>`都设成某种格式，怎么办？
  - `id`选择器/`class`选择器

- `#id-of-html-tag`
  - 前面带有井号#的 ID 引用

```
#id-of-html-tag {
 ...
}
```

- `.class-of-html-tag`
  - 前面带有点. 的类引用

```
.class-of-html-tag {
 ...
}
```

# id 选择器

- ▶ id 选择器可以为标有特定 id 的 HTML 元素指定特定的样式
- ▶ id 选择器以 "#" 来定义。

CSS: #red {color:red;} #green {color:green;}

HTML:

<p id="red">这个段落是红色。</p>

<p id="green">这个段落是绿色。</p>

# CSS 类选择器

- 类选择器以一个点号显示
- `.center {text-align: center}`

```
<h1 class="center"> This heading will be center-aligned </h1>
```

```
<p class="center">
```

```
 This paragraph will also be center-aligned.
```

```
</p>
```

# id选择器 or 类选择器

- class和id看起来除了一个.和#的区别之外，别无差异??
- 同一id在一个页面内只能应用一次
- class则是用于描述多次出现的元素。

# CSS 派生选择器

- 通过依据元素在其位置的上下文关系来定义样式。

- `li strong { font-style: italic; font-weight: normal; }`

`<p><strong>我是粗体字，不是斜体字，因为我不在列表当中，所以这个规则对我不起作用</strong></p> <ol>`

`<li><strong>我是斜体字。这是因为 strong 元素位于 li 元素内。`

`</strong></li>`

`<li>我是正常的字体。 </li>`

`</ol>`

只要strong标签在li标签内部就会起作用，不需要紧挨着。

## ▶ id 选择器和派生选择器 结合

```
#sidebar p { font-style: italic; text-align: right; margin-top: 0.5em; }
```

```
div#sidebar { border: 1px dotted #000; padding: 10px; }
```

# CSS 属性选择器

- 对带有指定属性的 HTML 元素设置样式

- 格式

- `[title] { color:red; }` 单纯的属性选择
- `[title=FirstLevel] { border:5px solid blue; }` 属性值选择
- `[title~=hello] { color:red; }` 属性多值选择-空格分隔
- `[lang|=en] { color:red; }` 属性多值选择-连字符分隔

[Demo](#)

# CSS创建的三种方式

## ▶ 外部样式表

HTML: `<head> <link rel="stylesheet" type="text/css" href="mystyle.css" />  
</head>`

CSS: `hr {color: sienna;}  
p {margin-left: 20px;}  
body {background-image:  
url("images/back40.gif");}`

## ▶ 内部样式表

`<head> <style type="text/css"> hr {color: sienna;} p {margin-left: 20px;}  
body {background-image: url("images/back40.gif");} </style> </head>`

## ▶ 内联样式

`<p style="color: sienna; margin-left: 20px"> This is a paragraph </p>`

等级覆盖关系

# 内容概要

- CSS的概念与实例
- CSS基本语法
- 选择器
- 样式
- CSS深入





# 样式——文字、文本属性

- 文字属性 (font)
  - font-family: 字体
  - font-size: 大小
  - font-style: 样式, 比如斜体
  - font-weight: 粗细
- 文本属性 (text)
  - 例如颜色、间距和首行缩进
  - text-indent: 首行缩进
  - color: 文本颜色
  - text-align: 文本对齐属性
  - text-decoration: 文本修饰, 如下划线

# 样式——背景、列表

- 背景

- background-color

- background-image

- background-repeat: 不重复、水平重复、.....

- background-position

- background-attachment: 背景图片是否随着滚动条滚动而改变位置

- 列表

- list-style-type

- list-style-image

# 样式——表格

- 表格属性可以极大地改善表格的外观
- border 属性：设置表格边框
- width、height 属性定义表格
- text-align 和 vertical-align 属性设置表格中文本的对齐方式
  - text-align 属性设置水平对齐方式，比如左对齐、右对齐或者居中：
  - vertical-align 属性设置垂直对齐方式，比如顶部对齐、底部对齐或居中对齐

```
table, th, td
{
border: 1px solid blue;
}
```

制作一个漂亮的表格

# 样式——轮廓

- 绘制元素周围的一条线，位于边框边缘的外围，起到突出元素的作用
- **outline** 属性规定元素轮廓的样式、颜色和宽度

## CSS 轮廓

属性	描述
<u><a href="#">outline</a></u>	在一个声明中设置所有的轮廓属性。
<u><a href="#">outline-color</a></u>	设置轮廓的颜色。
<u><a href="#">outline-style</a></u>	设置轮廓的样式。
<u><a href="#">outline-width</a></u>	设置轮廓的宽度。

# CSS 举例

- 图片显示的控制
  - 设置图片显示位置大小等
    - [http://www.w3school.com.cn/tiy/t.asp?f=csse\\_dim\\_height](http://www.w3school.com.cn/tiy/t.asp?f=csse_dim_height)

# 内容概要

- CSS的概念与实例

- CSS基本语法

- 选择器

- 样式

- CSS深入



# CSS深入

- HTML标准流 (normal flow)
- Box Model (框模型)
- CSS定位



# 标准流

- normal flow，译为标准流、普通流、文档流
  - 将窗体自上而下分成一行行，并在每行中按从左至右的顺序排放元素
  - 多数情况下元素呈现在 web 页面上的方式
  - 所有 HTML 都在块框(block boxes, 块级元素)或者行内框(inline boxes, 行内元素)中。
  - 每个新的块级元素渲染为新行。行内元素则按照顺序被水平渲染直到当前行遇到了边界，然后换到下一行垂直渲染
- CSS样式float会使元素脱离标准流。

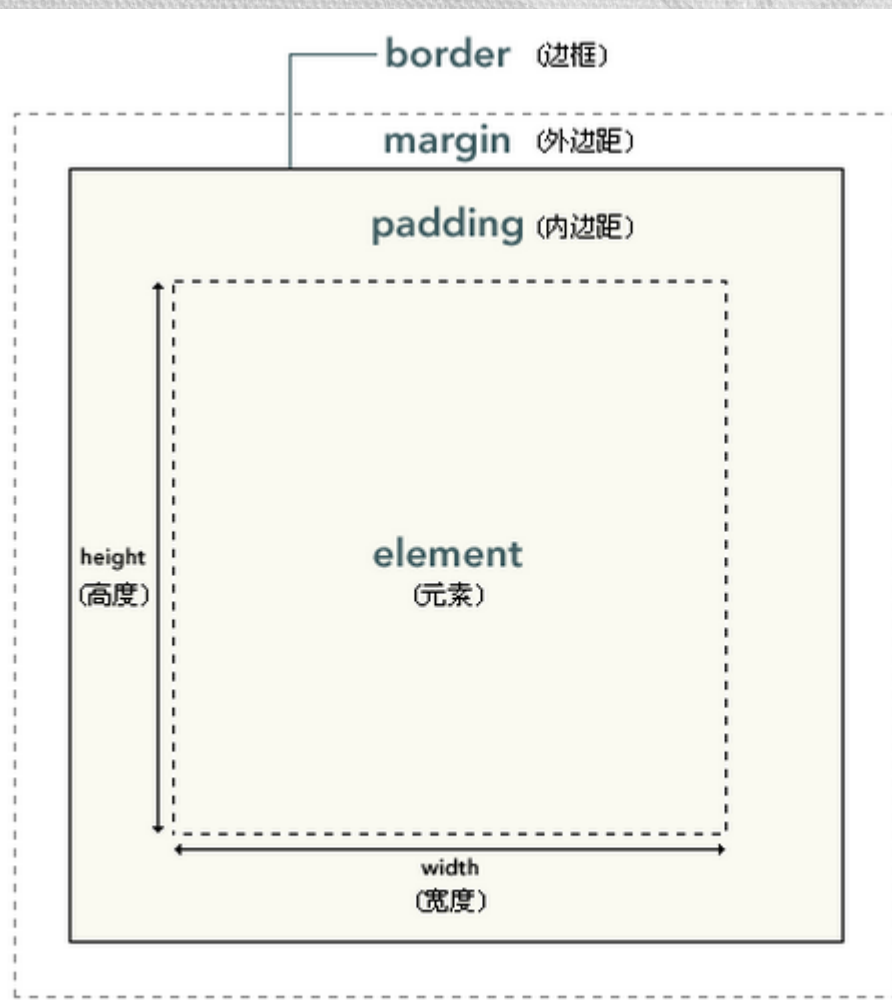


# 块元素&内联元素

- 大多数 HTML 元素被定义为块级元素(block level element)或内联元素(inline element)
  - 显示块级元素时，通常会以新行来开始（和结束）
    - 例如：<h1>, <p>, <ul>, <table>, <form>
  - 内联元素在显示时通常不会以新行开始
    - 例子：<b>, <td>, <a>, <img>, <em>
    - 任何不是块级元素的可见元素都是内联元素
- <div> 元素是块级元素，它是可用于组合其他 HTML 元素的容器。
- <span> 元素是内联元素，可用组合行内元素的容器。

# 框模型 Box Model

- 框模型规定了元素框处理元素内容、内边距、边框和外边距
- 盒子里由外至里依次是：
  - margin 边距
  - border 边框
  - padding 间隙 (也有人称做补丁)
  - content (内容, 比如文本, 图片等)



# Box Model(2)

- margin
  - 用来设置一个元素所占空间的边缘到相邻元素之间的距离。
- Border
  - 用来设定一个元素的边线。
- padding
  - 用来设置元素内容到元素边框的距离。
- CSS背景属性指的是 content 和 padding 区域。
- CSS 属性中的 width 和 height 指的是 content 区域的宽和高。

# Box Model示意

```
h2 {
 margin: 15px 0 15px 0;
 padding: 15px 15px 15px 15px;
}
```

- 按照上、右、下、左（顺时针）的顺序分别设置
- 也可通过四个单独的属性来设定
  - padding-top
  - padding-right
  - padding-bottom
  - padding-left



# CSS定位 (Positioning)

- CSS的定位（浮动）功能很强大，利用它你可以做出各种各样的网页布局
- 可以建立列式布局
- 将布局的一部分与另一部分重叠，
- 完成以往使用多个表格才能完成的任务

# CSS定位的基本思想

- 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
- 一切皆为框
- 提出了浮动的概念
  - 浮动不完全是定位，不是标准流布局

# CSS定位机制

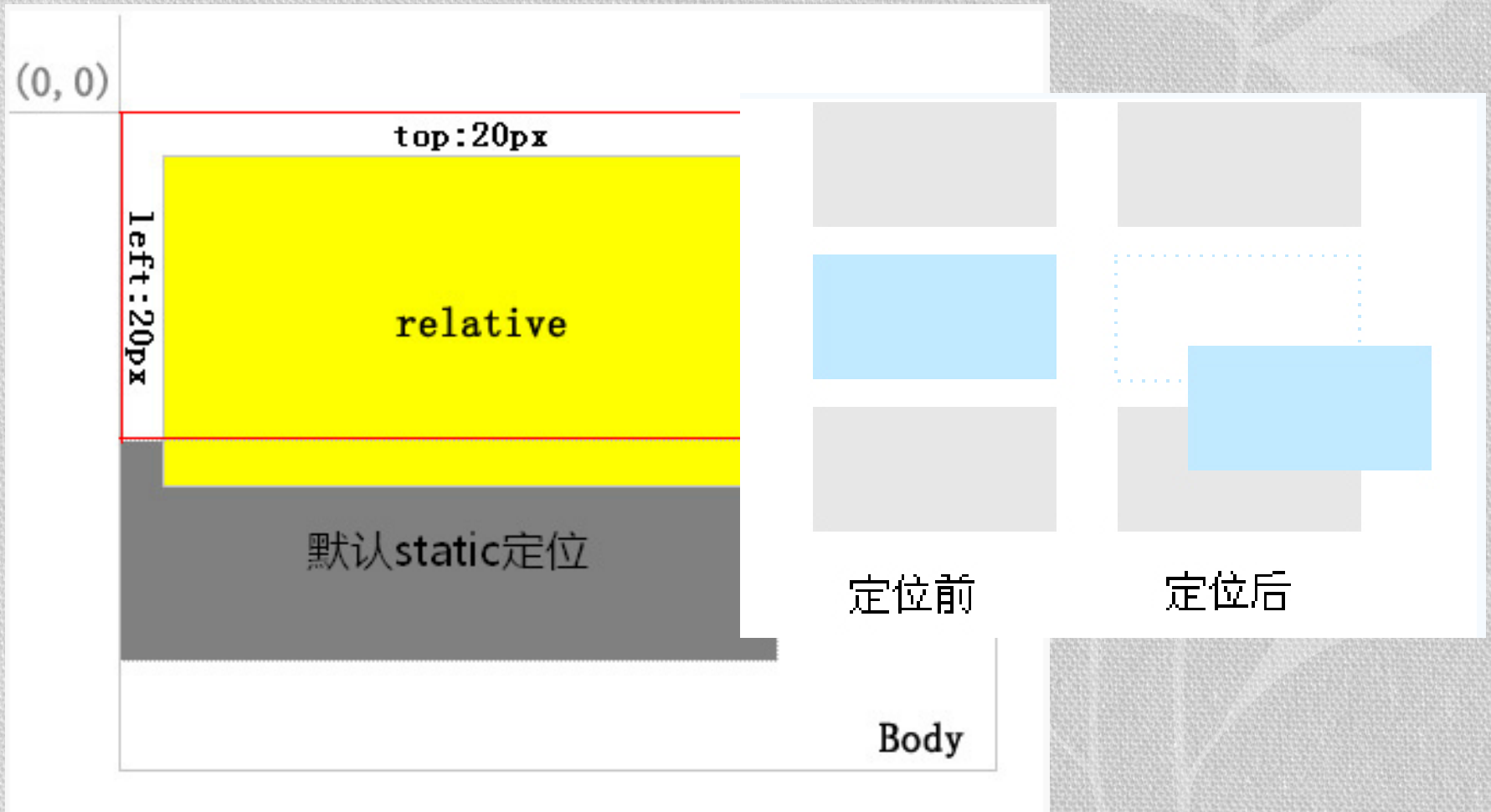
- CSS定位的机制：**普通流、浮动、绝对定位**
- 除非专门指定，否则所有框都在普通流中定位。即元素的位置由元素在HTML中的位置决定。
- 块级框从上到下一个接一个地排列，框之间的垂直距离是由框的垂直外边距计算出来。
- 行内框在一行中水平布置，可以使用水平内边距、边框和外边距调整它们的间距。

# 通过position属性可以设定4种定位

- static
  - 默认值。没有定位，元素出现在正常的流中。
- relative(相对定位)
  - 相对它本来应处的位置做的移动，但在文本流中的位置还在
- absolute (绝对定位)
  - 元素框从文档流完全删除
  - 绝对定位使元素的位置与文档流无关，因此不占据空间
  - 绝对定位的元素位置是相对于距离它最近的那个已定位的祖先(相对/绝对)元素决定的。如果元素没有已定位的祖先元素，那么它的位置相对于初始包含块。
- fixed
  - 生成绝对定位的元素，相对于浏览器窗口进行定位。

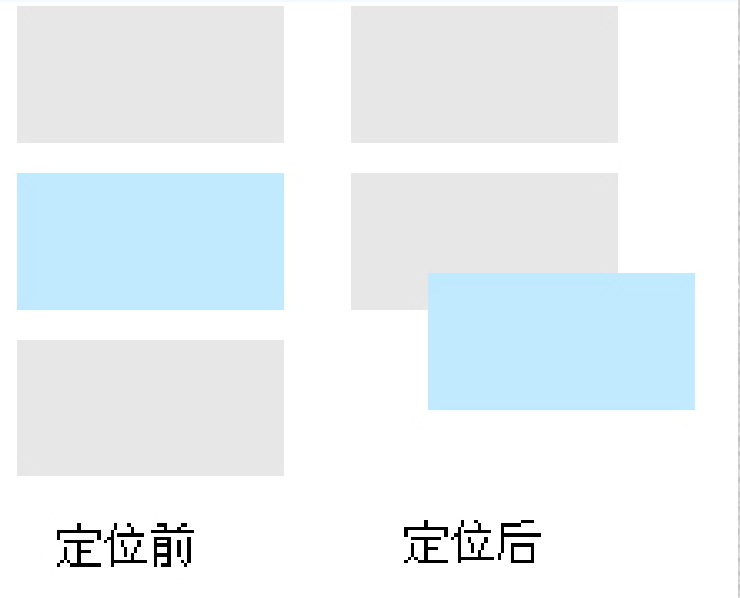
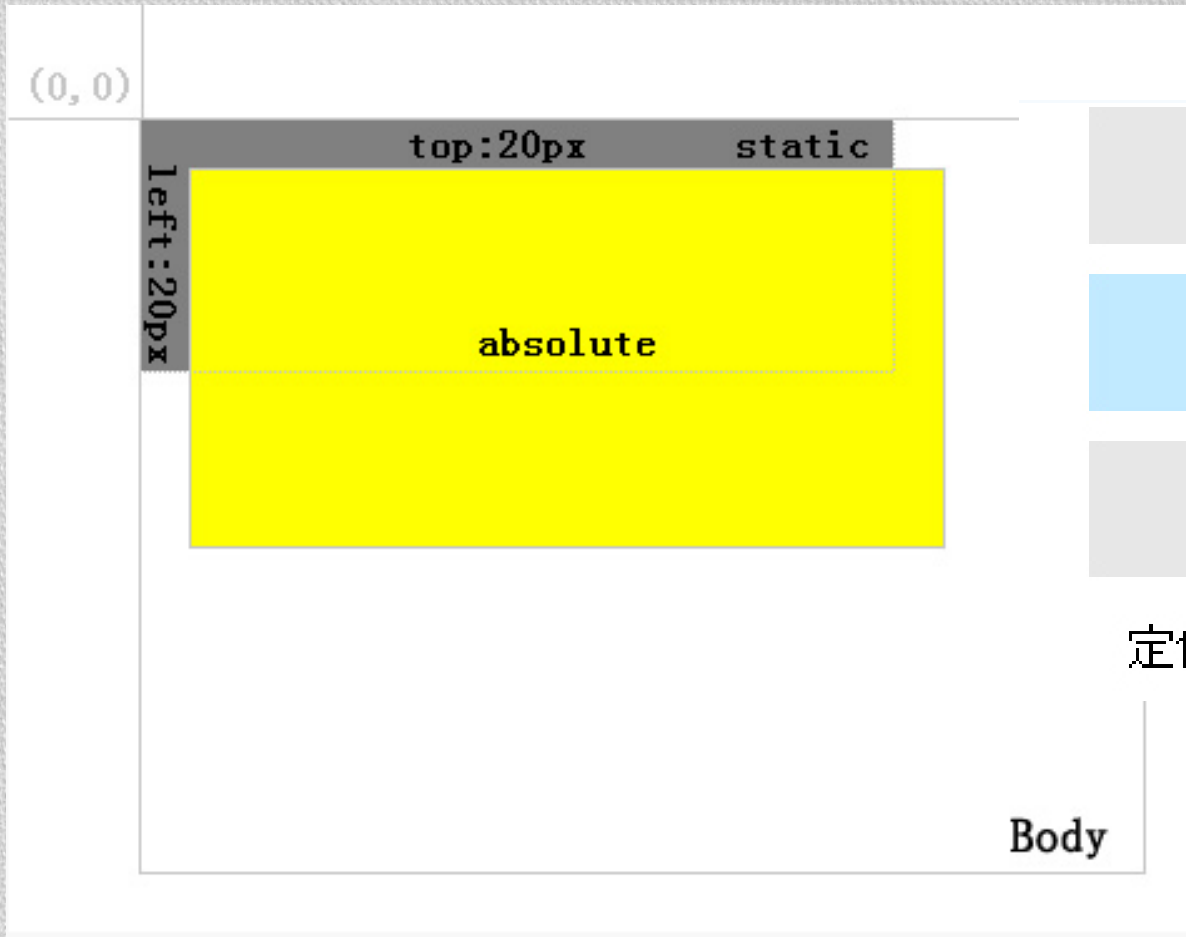


# relative



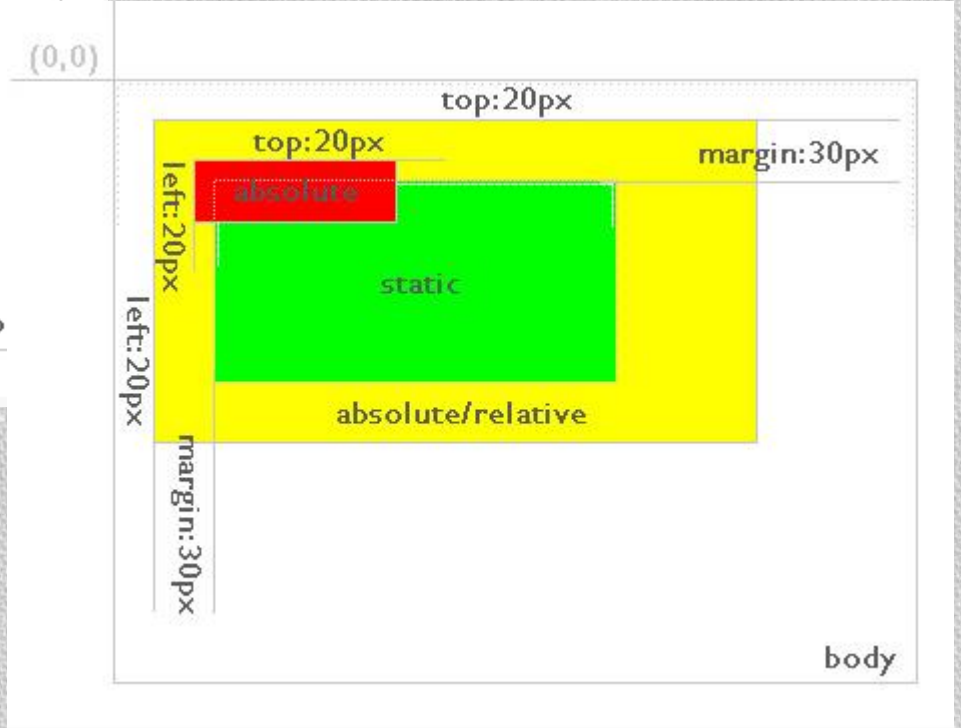
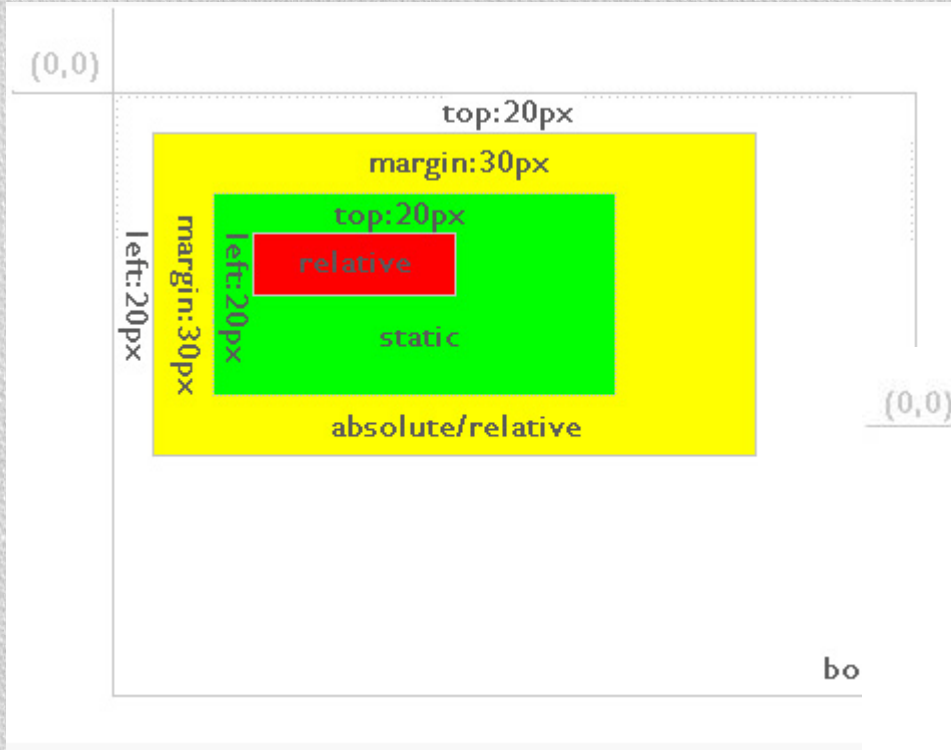
相对定位仍然属于普通流定位模型

# absolute



与相对定位不同，绝对定位使元素与普通流无关

# 区别



# Z-index属性

- 绝对定位的框与文档流无关，所以它们可以覆盖页面上的其它元素。可以通过设置 [z-index](#) 属性来控制这些框的堆放次序。

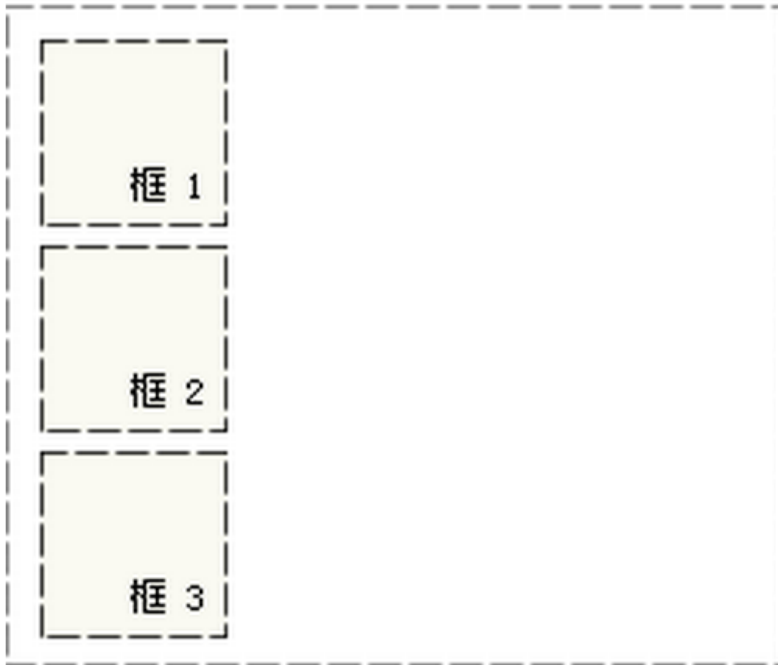
# CSS浮动

- floats是建立基于css的网站的一个基本元素，它可以用来调整图像和列的对齐方式
- 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
- 浮动框不在文档的普通流中，所以文档的普通流中的块框“看不见”浮动框
- CSS中通过float属性实现元素的浮动

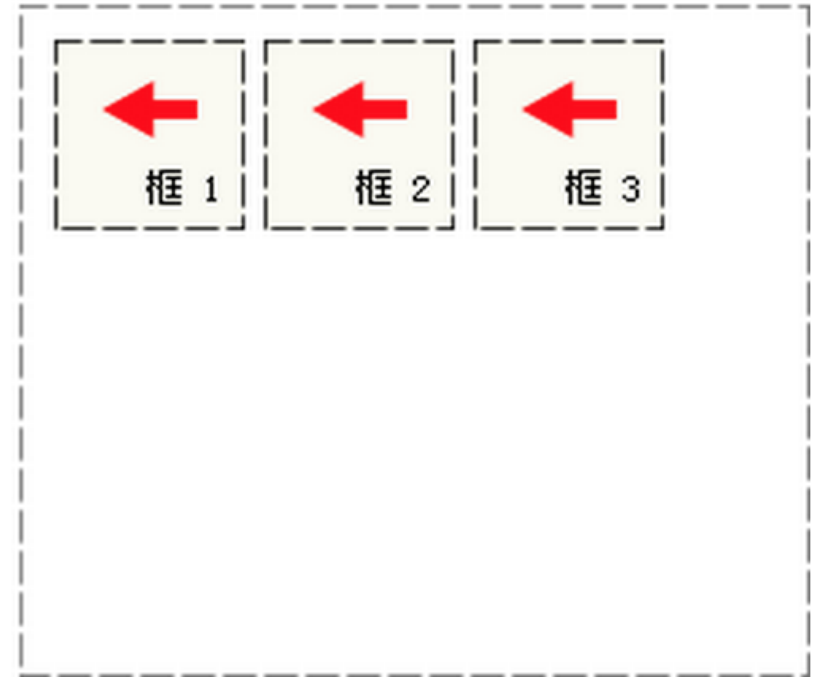
# CSS浮动示意

- 框1、框2、框3向左浮动

不浮动的框



所有三个框向左浮动

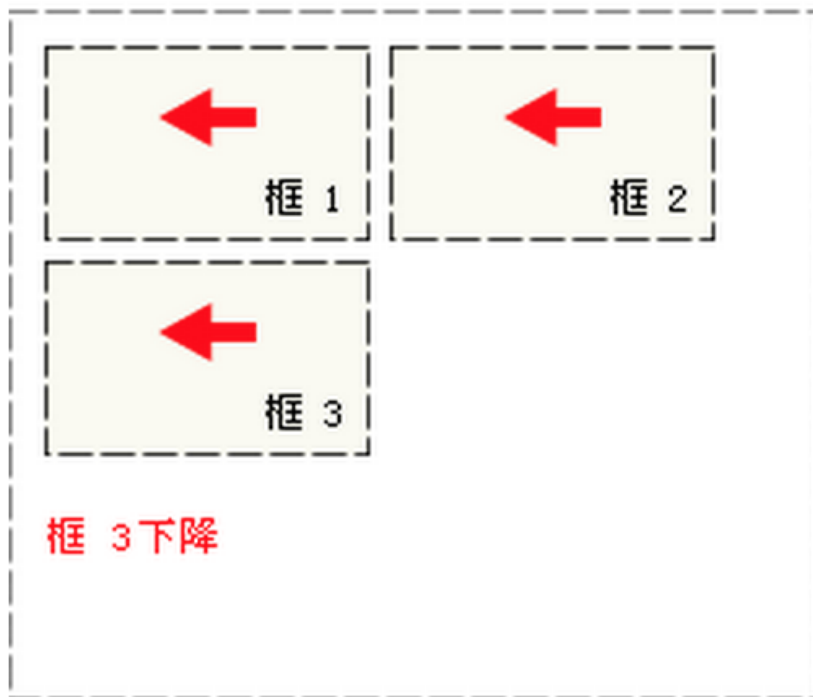


demo

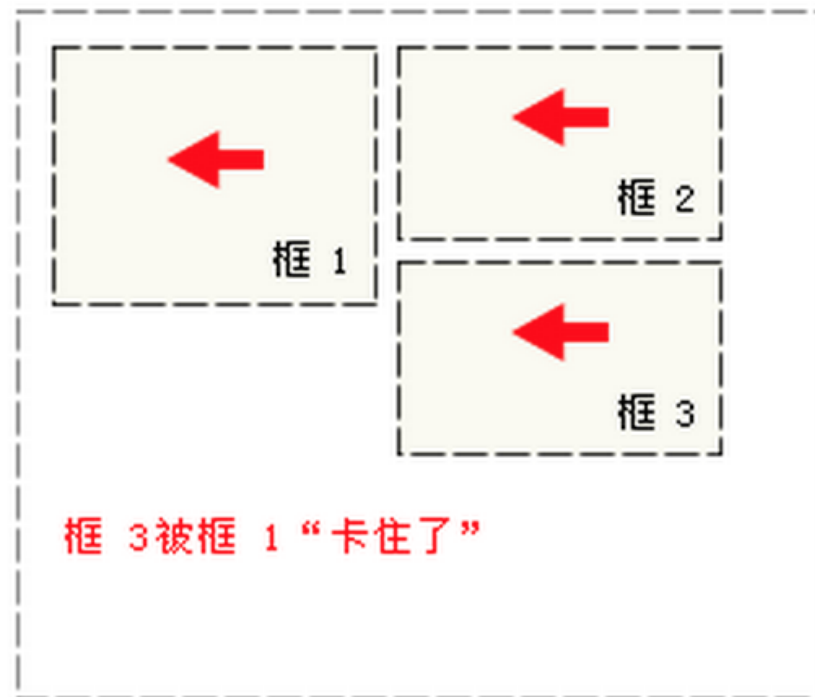
# CSS浮动 (2)

- 当包含框太窄，无法容纳水平排列的三个浮动元素时，那么其它浮动块向下移动，直到有足够的空间。
- 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”：

框 1 向左浮动



所有三个框向左浮动



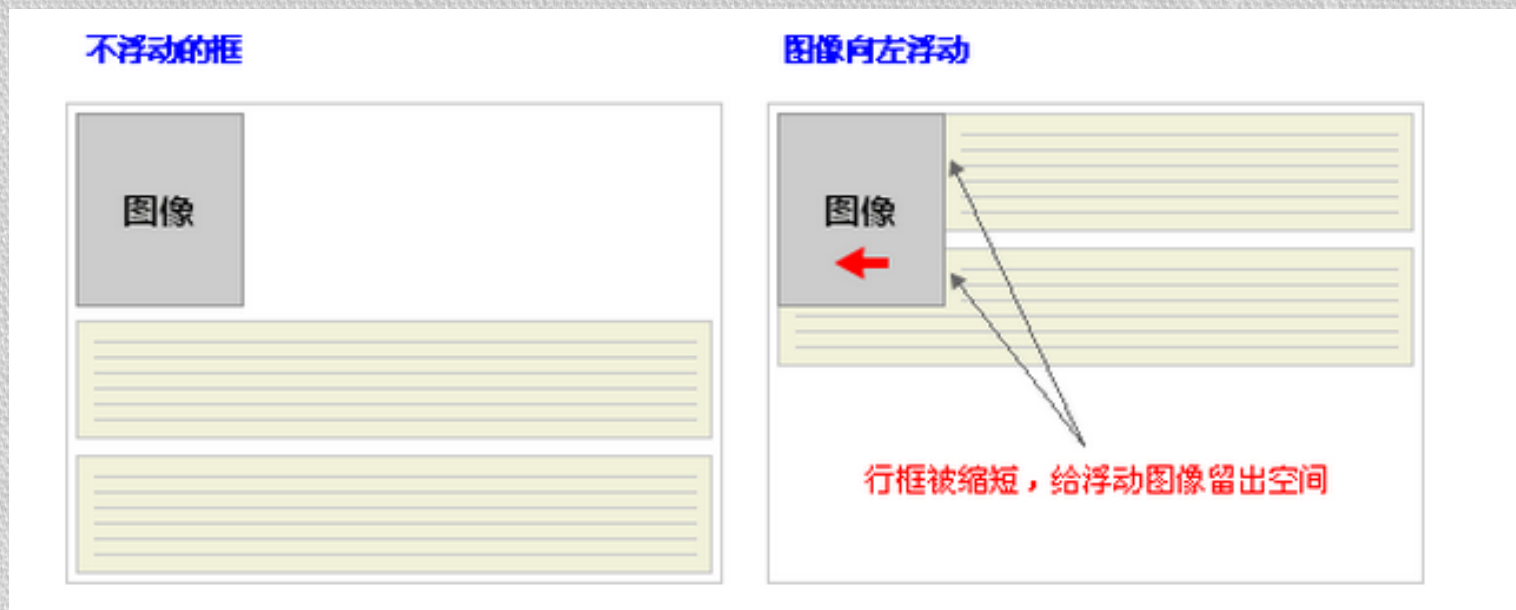
# 浮动举例

- 创建无表格的首页
- [http://www.w3school.com.cn/tiy/t.asp?f=csse\\_float6](http://www.w3school.com.cn/tiy/t.asp?f=csse_float6)



# 行框

- 浮动框旁边的行框被缩短，从而给浮动框留出空间，行框围绕浮动框
- 创建浮动框可以使文本围绕图像



demo

# 清除浮动clear

- 阻止行框围绕浮动框，需要对该框应用 clear 属性。
- clear 属性的值可以是 left、right、both 或 none，表示当前框的哪些边不能应该挨着浮动框

# 清除浮动clear

- 在标准流中，元素是竖向排列的，而浮动之后可以理解为横向排列。
- 清除浮动可以理解为打破横向排列

注意：CSS的清除浮动，只能影响使用清除的元素本身，不能影响其他元素

语法：

`clear : none | left | right | both`

取值：

`none` : 默认值。允许两边都可以有浮动对象

`left` : 不允许左边有浮动对象

`right` : 不允许右边有浮动对象

`both` : 不允许有浮动对象

# CSS布局学习参考

- <http://zh.learnlayout.com/>