# CSS

## 行内元素、块级元素

### 元素的类型

CSS中将元素分为行内元素和块级元素，他们都各自有各自的特点和行为习惯。

div、h1 或 p 元素常常被称为块级元素。这意味着这些元素显示为一块内容，即“块框”，也称为块级元素。在布局页面时，块级元素独占一行。

与之相反，span 和 strong 等元素称为“行内元素”，这是因为它们的内容显示在行中，即“行内框”，也称为行内元素。在布局页面时，行内元素不会独占一行。同时垂直内边距、边框和外边距不影响行内框的高度。

可以通过display属性修改框的类型。

## CSS概述

### CSS是什么

CSS是层叠样式表（Cascading Style Sheets）用来定义网页的显示效果。可以解决html代码对样式定义的重复，提高了后期样式代码的可维护性，并增强了网页的显示效果功能。简单一句话：CSS将网页内容和显示样式进行分离，提高了显示功能。

### 组织网页的两种常用方式

组织网页结构主要有两种方式：

表格嵌套表格

DIV+CSS

早期的网页都是表格套表格的方式实现的，这种方式对于比较简单的页面效果尚可，一旦页面稍微复杂一些，使用起来就十分的麻烦，现阶段的大部分网页都主要使用DIV+CSS的方式，只在局部使用表格嵌套方式

## 引入CSS的四种方式

### Style属性方式

利用标签中style属性来改变每个标签的显示样式。

例：

<p style="background-color:#FF0000; color:#FFFFFF">

p标签段落内容。

</p>

该方式比较灵活，但是对于多个相同标签的同一样式定义比较麻烦，适合局部修改。

### Style标签方式

在head标签中加入style标签，对多个标签进行统一修改。

<head>

<style type=”text/css”>

p { color:#FF0000;}

</style>

</head>

该方式可以对单个页面的样式进行统一设置，但对于局部不够灵活。

### 导入方式

前提是已经存在一个定义好的CSS文件。网页的一部分样式需要用到，那么就用到这种方式。

例：

<style type="text/css">

@import url(div.css);

div { color:#FF0000;}

</style>

注：url括号后面必须要用分号结束。如果导入进来的样式与本页面定义的样式重复，以本页定义样式为准。

### 链接方式

通过head标签中link标签来实现，前提也是先要有一个已定好的CSS文件。

例：

<link rel="stylesheet"type="text/css" href="css\_3.css" media="screen" />

注：可以通过多个link标签链接进来多个CSS文件。重复样式以最后链接进来的CSS样式为准。

### 四种方式的优先级

由上到下，由外到内。优先级由低到高。

## CSS代码格式

### css代码格式

选择器名称 { 属性名：属性值1 ; 属性值2 属性值3 ；属性名：属性值；…….}

属性与属性之间用 分号 隔开

属性与属性值直接按用 冒号 连接

如果一个属性有多个值的话，那么多个值用 空格 隔开。

## CSS基本选择器

### html标签名选择器

html标签名选择器。使用的就是html的标签名

div｛

color:red;

｝

\*{

Color:red;

}

### class选择器

class选择器。其实使用的标签中的class属性。格式为 .class名

.aaa{

color:red;

}

<div class=”aaa”>hahahaha</div>

### id选择器

id选择器。其实使用的是标签的中的id属性。格式为#id名

#bbb{

color:red;

}

<div id=”bbb”>hehehe</div>

## CSS扩展选择器

### 后代选择器

选择父选择器选中的元素中符合子选择器的子孙元素，可以用此选择器。

例：

p { color:#00FF00;}

p b { color:#FF000;}

<p>P标签<b>刘德华</b><a href=”#”><b>段落</b>样式</a></p>

<p>P标签段落</p>

### 子元素选择器

选择父选择器选中的元素中符合子选择器的子元素，可以用此选择器。

与后代选择器相比，子元素选择器（Child selectors）只能选择作为某元素子元素的元素。

h1 > strong {color:red;}

这个规则会把第一个 h1 下面的两个 strong 元素变为红色，但是第二个 h1 中的 strong 不受影响：

<h1>

This is <strong>very</strong> <strong>very</strong> important.

</h1>

<h1>

This is <em>really <strong>very</strong></em> important.

</h1>

### 分组选择器

对多个不同选择器进行相同样式设置的时候应用此选择器。

例：

p,div { color:#FF0000;}

<p>P标签显示段落。</p>

<div>DIV标签显示段落</div>

注：多个不同选择器要用逗号分隔开。

### 属性选择器

**属性选择器可以根据元素的属性及属性值来选择元素。**

如果您希望把包含属性（title）的所有元素变为红色，可以写作：

\*[title] {color:red;}

与上面类似，可以只对有 href 属性的锚（a 元素）应用样式：

a[href] {color:red;}

为了将同时有 href 和 title 属性的 HTML 超链接的文本设置为红色，可以这样写：

a[href][title] {color:red;}

假设只希望选择 moons 属性值为 1 的那些 planet 元素：

planet[moons="1"] {color: red;}

上面的代码会把以下标记中的第二个元素变成红色，但第一个和第三个元素不受影响：

<planet>Venus</planet>

<planet moons="1">Earth</planet>

<planet moons="2">Mars</planet>



### 相邻兄弟选择器

如果需要选择紧接在另一个元素后的元素，而且二者有相同的父元素，可以使用相邻兄弟选择器。

h1 + p {margin-top:50px;}

### 伪元素选择器

其实就在html中预先定义好的一些选择器。称为伪元素。是因为CSS的术语。

格式：标签名：伪元素。类名 标签名。类名：伪元素。都可以。

a:link 超链接未点击状态。

a:visited 被访问后的状态。

a:hover 光标移到超链接上的状态（未点击）。

a:active 点击超链接时的状态。

## 盒子模型

### 盒子模型概述

CSS盒子模型就是在网页设计中经常用到的CSS技术所使用的一种思维模型。

如图-2所示：

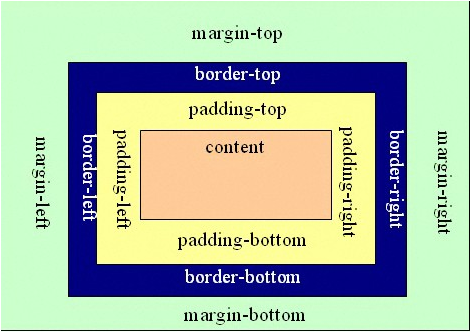


图-2

### Padding

padding 属性定义元素的内边距。padding 属性接受长度值或百分比值，但不允许使用负值。

如果设置一个值，则上下左右内边距都取该值。

如果设置两个值，则第一个值为上下边距，第二个值为左右边距。

如果设置三个值，则第一个值为上边距，第二个为左右边距，第三个为下边据。

如果设置四个值，则按照上右下左边距取值，即按顺时针方向取值。

可以为元素的内边距设置百分数值。百分数值是相对于其父元素的 width 计算的。所以，如果父元素的 width 改变，它们也会改变。

也可以通过如下属性单独设置上右下左内边距。

[padding-top](http://www.w3school.com.cn/cssref/pr_padding-top.asp)：上内边距

[padding-right](http://www.w3school.com.cn/cssref/pr_padding-right.asp)：右内边距

[padding-bottom](http://www.w3school.com.cn/cssref/pr_padding-bottom.asp)：下内边距

[padding-left](http://www.w3school.com.cn/cssref/pr_padding-left.asp)：左内边距

### Margin

围绕在元素边框的空白区域是外边距。设置外边距会在元素外创建额外的“空白”。

设置外边距的最简单的方法就是使用 margin 属性，这个属性接受任何长度单位、百分数值甚至负值。

属性接受任何长度单位，可以是像素、英寸、毫米或 em。

可以设置为 auto。

百分数是相对于父元素的 width 计算的。

多值设置规则和Padding相同。

也可以通过如下属性当杜设置上右下左外边距。

[margin-top](http://www.w3school.com.cn/cssref/pr_margin-top.asp" \o "CSS margin-top 属性)：上外边距

[margin-right](http://www.w3school.com.cn/cssref/pr_margin-right.asp" \o "CSS margin-right 属性): 右外边距

[margin-bottom](http://www.w3school.com.cn/cssref/pr_margin-bottom.asp" \o "CSS margin-bottom 属性): 下外边距

[margin-left](http://www.w3school.com.cn/cssref/pr_margin-left.asp" \o "CSS margin-left 属性): 左外边距

### Border

元素的边框 (border) 是围绕元素内容和内边距的一条或多条线。

CSS border 属性允许你规定元素边框的样式、宽度和颜色。

Border-style:定义边框样式

[border-top-style](http://www.w3school.com.cn/cssref/pr_border-top_style.asp" \o "CSS border-top-style 属性)

[border-right-style](http://www.w3school.com.cn/cssref/pr_border-right_style.asp" \o "CSS border-right-style 属性)

[border-bottom-style](http://www.w3school.com.cn/cssref/pr_border-bottom_style.asp" \o "CSS border-bottom-style 属性)

[border-left-style](http://www.w3school.com.cn/cssref/pr_border-left_style.asp" \o "CSS border-left-style 属性)

border-width：边框宽度

[border-top-width](http://www.w3school.com.cn/cssref/pr_border-top_width.asp" \o "CSS border-top-width 属性)

[border-right-width](http://www.w3school.com.cn/cssref/pr_border-right_width.asp" \o "CSS border-right-width 属性)

[border-bottom-width](http://www.w3school.com.cn/cssref/pr_border-bottom_width.asp" \o "CSS border-bottom-width 属性)

[border-left-width](http://www.w3school.com.cn/cssref/pr_border-left_width.asp" \o "CSS border-left-width 属性)

border-color：边框颜色

[border-top-color](http://www.w3school.com.cn/cssref/pr_border-top_color.asp" \o "CSS border-top-color 属性)

[border-right-color](http://www.w3school.com.cn/cssref/pr_border-right_color.asp" \o "CSS border-right-color 属性)

[border-bottom-color](http://www.w3school.com.cn/cssref/pr_border-bottom_color.asp" \o "CSS border-bottom-color 属性)

[border-left-color](http://www.w3school.com.cn/cssref/pr_border-left_color.asp" \o "CSS border-left-color 属性)

注意，如果边框样式设置为none，那么无论设置多少宽度真实值都为0，即不可见也不占用空间。

## 三种定位方式

### 三种定位方式

CSS 有三种基本的定位机制：普通流、浮动和绝对定位。

## 文档流定位

### 文档流定位概述

除非专门指定，否则所有框都在普通流中定位。也就是说，普通流中的元素的位置由元素在 (X)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。但是，垂直内边距、边框和外边距不影响行内框的高度。

## Float\_浮动

### 浮动概述

浮动可以改变元素的布局方式。

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。

由于浮动框不在文档的普通流中，所以文档的普通流中的块框表现得就像浮动框不存在一样。

浮动元素会生成一个块级框，而不论它本身是何种元素。

如图-3所示：

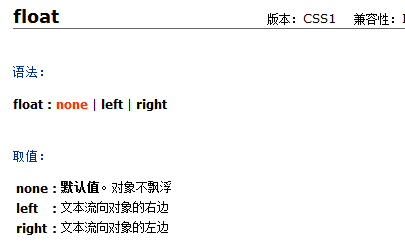


图-3

### 浮动演示

请看下图，当把框 1 向右浮动时，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：

如图-4所示：

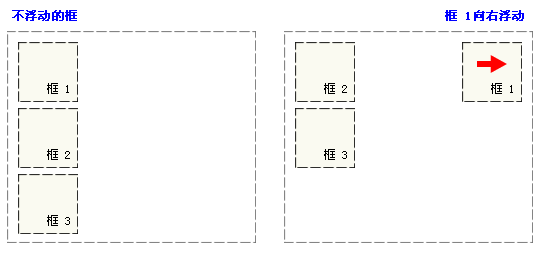


图-4

再请看下图，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。因为它不再处于文档流中，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。

如图-5所示：

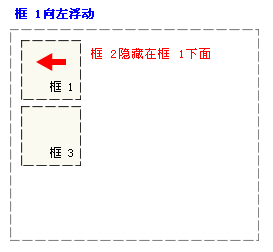


图-5

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。

如图-6所示：

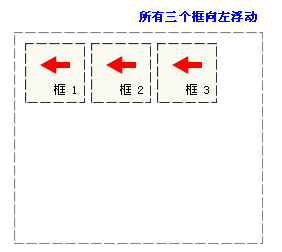


图-6

如下图所示，如果包含框太窄，无法容纳水平排列的三个浮动元素，那么其它浮动块向下移动，直到有足够的空间。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”，如图-7所示：

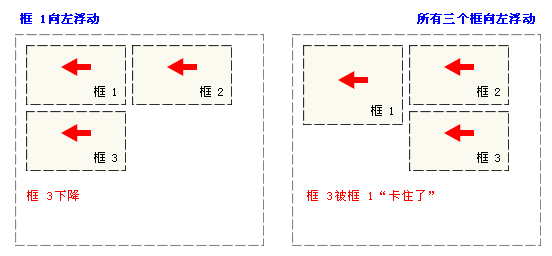


图-7

### Clear\_清除浮动

由于浮动的元素脱离了文档流，对其他元素的产生了影响，可以通过clear禁止元素左右具有浮动元素，从而阻止这种影响。

Clear可以指定元素的左/右/左右边上不允许有浮动元素。

如图-8所示：



图-8

## 定位

### CSS position 属性

通过使用 [position 属性](http://www.w3school.com.cn/cssref/pr_class_position.asp" \o "CSS position 属性)，我们可以选择不同类型的定位，这会影响元素框生成的方式。

position 属性值的含义：

static

元素框正常生成。块级元素生成一个矩形框，作为文档流的一部分，行内元素则会创建一个或多个行框，置于其父元素中。

relative

元素框偏移某个距离。元素仍保持其未定位前的形状，它原本所占的空间仍保留。

absolute

元素框从文档流完全删除，并相对于其包含块定位。包含块可能是文档中的另一个元素或者是初始包含块。元素原先在正常文档流中所占的空间会关闭，就好像元素原来不存在一样。元素定位后生成一个块级框，而不论原来它在正常流中生成何种类型的框。

fixed

元素框的表现类似于将 position 设置为 absolute，不过其包含块是视窗本身。

提示：相对定位实际上被看作普通流定位模型的一部分，因为元素的位置相对于它在普通流中的位置。

### TOP　bottom left right

在绝对定位时，通过这些属性指定当前元素基于最近的有定位的祖先元素定位位置关系。

在相对定位时，通过这些属性指定当前元素基于当前元素的初始位置的位移进行定位。

### z-index

指定元素的堆叠顺序，数值越大离用户越近，数值越小离用户越远，此值可以为负数。

## Display

### Display属性详解

设置元素的显示方式。

我们知道元素分为行内元素和块级元素，而display属性可以修改元素的显示方式。

如图-1所示：



图-1

\*visibility也可以实现隐藏，不同的是display不显示，也不占用空间，而visibility隐藏，但是空间仍然占用。

## 其他CSS属性

### 其他CSS属性

对齐方式

vertical-align

设置或检索对象内容的垂直对其方式。

text-align

设置或检索对象中文本的对齐方式。

列表属性

list-style

设置列表项目相关样式

list-style-position

设置或检索作为对象的列表项标记如何根据文本排列。

list-style-type

设置或检索对象的列表项所使用的预设标记。

字体设置

Font

字体设置

Color

字体颜色

Font-family

字体设置

Font-size

字体大小

Font-style

斜体设置

font-weight

字体粗细

text-decoration

字体下划线

text-underline-position

下划线位置

line-height

行高。即字体最底端与字体内部顶端之间的距离。

宽高设置

Height

高度

Width

宽度

背景设置

Background

背景设置

background-color

背景色设置

background-image

背景图片设置

background-position

设置或检索对象的背景图像位置

background-repeat

设置或检索对象的背景图像是否及如何铺排

# EasyMall页面实现

## 代码实现

### 代码实现参照课堂案例

主页的页面实现

用户注册页面实现的页面实现

用户登录页面实现的页面实现

商品列表实现的页面实现

商品信息页面实现的页面实现

购物车实现的页面实现

支付页面的页面实现

订单列表页面的页面实现

增加订单页面的页面实现