

网页编程基础 CSS3基础

WebBasic CSS3BASIC

Unit04

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	显示
	10:30 ~ 11:20	
	11:30 ~ 12:00	列表
下午	14:00 ~ 14:50	定位
	15:00 ~ 15:50	
	16:00 ~ 16:50	
	17:00 ~ 17:30	总结和答疑



显示

显示

显示方式

显示方式

display 属性

显示效果

visibility 属性

opacity 属性

vertical-align 属性

光标

光标

cursor 属性

显示方式



显示方式

- “一切皆为框”：页面上所有的元素都可以显示为框
- 块级元素
 - `<div>`、`<h1>` 或 `<p>` 元素等
 - 这些元素显示为一块内容，即“块框”
- 内联元素 / 行内元素
 - ``、`<a>` 元素等
 - 内容显示在行中，即“行内框”

除了默认的显示效果之外，可以使用 `display` 属性来修改元素框的显示方式，即改变生成的框的类型



display 属性

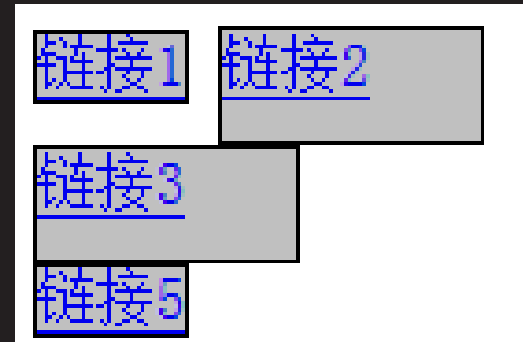
- 取值
 - display : none/block/inline/inline-block;
- none
 - 让生成的元素根本没有框
 - 该框及其所有内容就不再显示，不占用文档中的空间
- block
 - 让行内元素（比如 <a> 元素）表现得像块级元素一样
- inline
 - 让块级元素（比如 <p> 元素）表现得像内联元素一样
- inline-block
 - 行内块元素



display 属性 (续1)

CSS 样式 :

```
a
{
    border:1px solid black;
    width:70px;
    height:30px;
    background-color:silver;
}
```



HTML 文档 :

```
<a href="#">链接1</a>
<a href="#" style="display:inline-block;">链接2</a>
<a href="#" style="display:block;">链接3</a>
<a href="#" style="display:none;">链接4</a>
<a href="#">链接5</a>
```

显示效果



visibility 属性

- visibility 属性规定元素是否可见
- 可取值
 - visible : 默认值, 元素可见
 - hidden : 元素不可见, 但是依然占据空间
 - collapse : 用在表格元素时, 可删除一行或一列, 且不影响表格的布局

即使不可见的元素也会占据页面上的空间
可以使用 "display" 属性来创建不占据页面空间的不可见元素



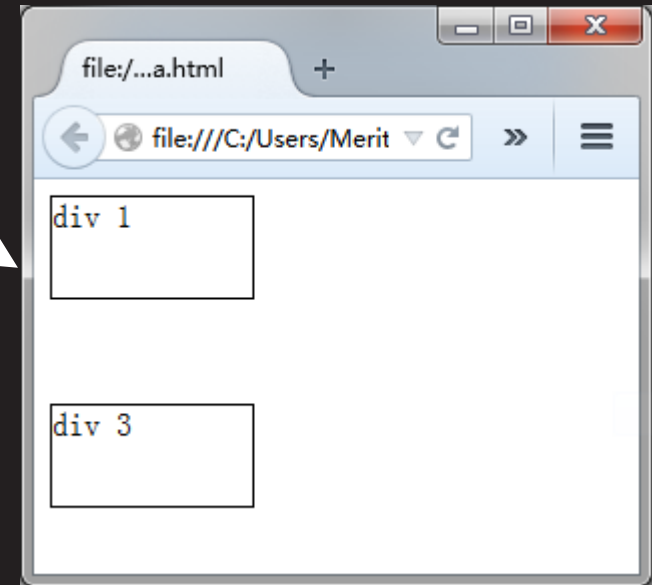
visibility 属性 (续1)

CSS 样式 :

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

HTML 文档 :

```
<div>div 1</div>
<div style="visibility:hidden;">div 2</div>
<div>div 3</div>
```



opacity 属性

- opacity 属性设置元素的不透明级别
- 取值 opacity : value;
 - value 规定不透明度，从 0.0（完全透明）到 1.0（完全不透明）



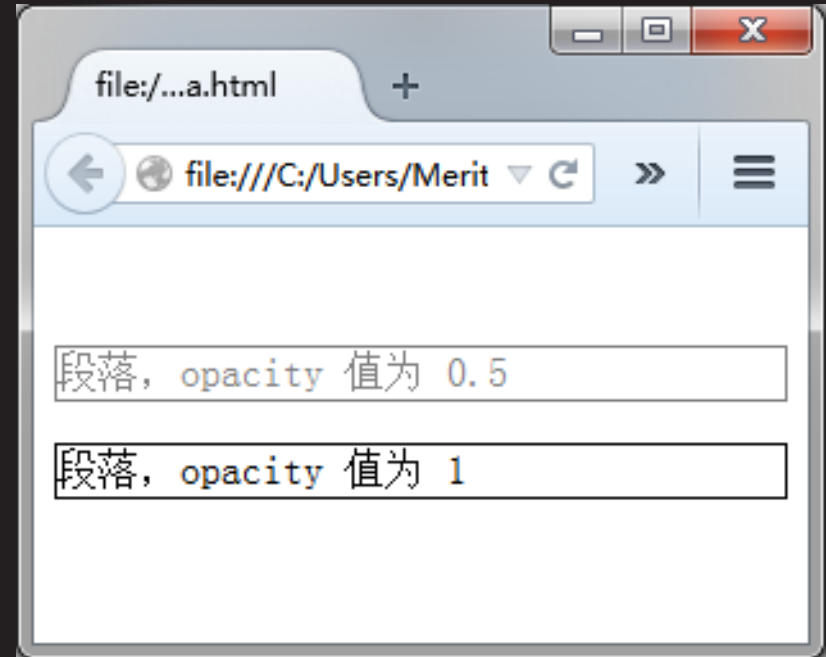
opacity 属性 (续1)

CSS 样式 :

```
p {border:1px solid black;}
#p1 {opacity:0;}
#p2 {opacity:0.5;}
#p3 {opacity:1;}
```

HTML 文档 :

```
<p id="p1">段落 , opacity 值为 0</p>
<p id="p2">段落 , opacity 值为 0.5</p>
<p id="p3">段落 , opacity 值为 1</p>
```



vertical-align 属性

- 设置单元格框中的单元格内容的垂直对齐方式
- 对于行内块级元素，如 ``，可设置垂直对齐方式
 - 定义行内元素的基线相对于该元素所在行的基线的垂直对齐
- 常用取值
 - baseline：默认，元素放置在父元素的基线上
 - top：把元素的顶端与行中最高元素的顶端对齐
 - bottom：把元素的顶端与行中最低的元素顶端对齐
 - middle：把此元素放置在父元素的中部



vertical-align 属性 (续1)

CSS 样式 :


```
div {
    width:400px;
    height:100px;
    border:1px solid black;
}
img{border:1px solid red;}
```

```
#img1{vertical-align:middle;}
```

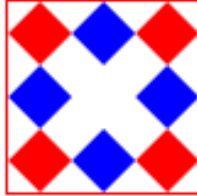
```
#img2{vertical-align:top;}
```

```
#img3{vertical-align:bottom;}
```

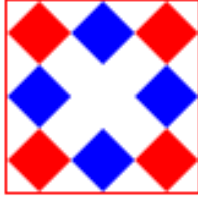
没有设置 img 的垂直对齐



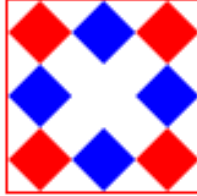
img 的vertical-align 为 middle



img 的vertical-align 为 top



img 的vertical-align 为 bottom



光标



光标

- 默认情况下，光标会根据用户的操作发生改变
 - 当鼠标悬停在一个链接上时，光标将从指针形状变为手状形状
 - 当鼠标悬停在文本区域时，会显示 I 形状
 - 而当鼠标悬停在一个按钮上时，光标会显示为箭头
- 可以使用 `cursor` 属性指定显示给用户的鼠标光标类型（形状）
 - 可以为用户提供一种可视化的暗示，提示可以进行的操作



cursor 属性

- cursor 属性定义了鼠标指针放在一个元素边界范围内时所用的光标形状
- 可取值
 - default
 - pointer
 - crosshair
 - text
 - wait
 - help 等



显示属性的应用

- 【参见 COOKBOOK】
 - 使用显示属性为页面元素定义样式



列表

列表

列表样式

列表项标志 list-style-type

列表项图像 list-style-image

列表项位置 list-style-position

列表属性 list-style

列表样式



列表项标志 list-style-type

- list-style-type 属性用于控制列表中列表项标志的样式
 - 无序列表：出现在各列表项旁边的圆点
 - 有序列表：可能是字母、数字或另外某种计数体系中的一个符号
- 无序列表取值
 - none：无标记
 - disc：实心圆，为默认值
 - circle：空心圆
 - square：实心方块



列表项标志 list-style-type (续1)

- 有序列表取值
 - none : 无标记
 - decimal : 数字 (如 1,2,3,4,5) , 为默认值
 - lower-roman : 小写罗马数字 (如 i, ii, iii, iv, v)
 - upper-roman : 大写罗马数字 (如 I, II, III, IV, V)
 - 等



列表项图像 list-style-image

- list-style-image 属性使用图像来替换列表项的标记
 - 取值为：url()，指定图像作为有序或无序列表项的标志

HTML 文档：

```
<ul class="parent">
  <li>账户
    <ul>
      <li><a href="#">基本信息</a></li>
      <li><a href="#">修改密码</a></li>
    </ul>
  </li>
  <li>相册
    <ul>
      <li><a href="#">上传照片</a></li>
      <li><a href="#">我的相册</a></li>
    </ul>
  </li>
</ul>
```

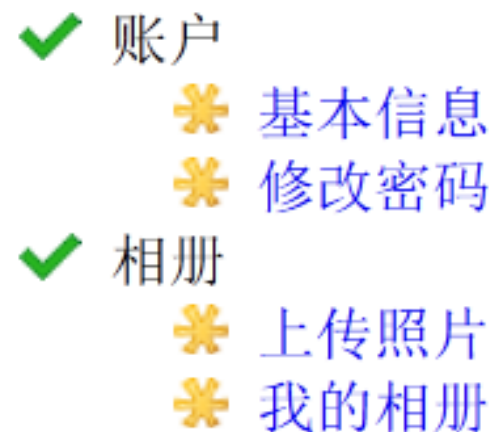


列表项图像 list-style-image (续1)

CSS 样式 :

```
li {list-style-type:none;}  
li a {text-decoration:none;}
```

```
ul.parent li  
{  
    list-style-image:url(dh.gif);  
}  
ul.parent li ul li  
{  
    list-style-image:url(ph.gif);  
}
```



列表项位置 list-style-position

- list-style-position 属性可以控制列表标记的位置
- 可取值
 - outside : 标记位于文本的左侧，且放置在文本以外，为默认值
 - inside : 标记放置在文本以内



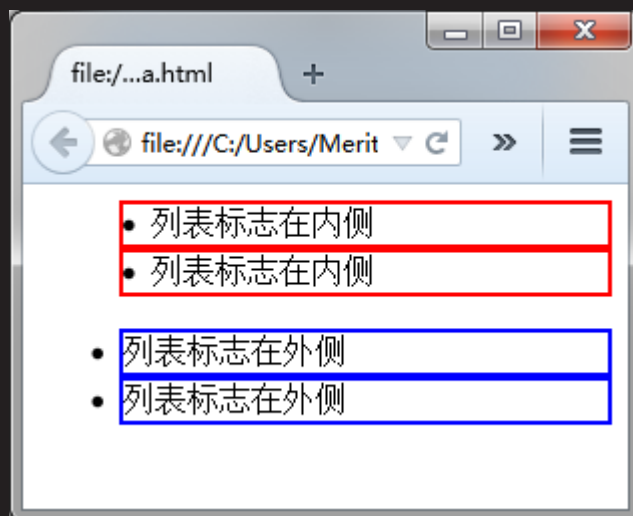
列表项位置 list-style-position (续1)

CSS 样式 :

```
.list1 li{  
    list-style-position:inside;  
    border:2px solid #f00;  
}  
.list2 li{  
    list-style-position:outside;  
    border:2px solid #00f;  
}
```

HTML 文档 :

```
<ul class="list1">  
    <li>列表标志在内侧</li>  
    <li>列表标志在内侧</li>  
</ul>  
<ul class="list2">  
    <li>列表标志在外侧</li>  
    <li>列表标志在外侧</li>  
</ul>
```



列表属性 list-style

- list-style 属性为简写属性，用于控制列表项的样式
- 语法为
 - list-style: type url(图像URL) position;
- 一般情况下，直接将list-style属性设置为 none



列表属性的应用

- 【参见 COOKBOOK】
 - 使用列表属性为页面元素定义样式



定位

定位

定位概述

什么是定位

定位属性

定位方式

相对定位

绝对定位

堆叠顺序

固定定位

定位概述



什么是定位

- 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
 - 内联元素将在一行中从左到右排列水平布置
- 浮动
- 相对定位
- 绝对定位
- 固定定位



定位属性

属性	说明
position	规定元素的定位类型，可取值： static/relative/absolute/fixed
偏移属性	top、bottom、left、right属性，用于定义元素框的偏移位置
z-index	设置元素的堆叠顺序
float/clear	浮动定位属性

使用position 属性和偏移属性

实现普通流定位、相对定位、绝对定位和固定定位

使用float 属性实现浮动定位

其他属性为辅助属性



定位属性（续1）

- position 属性：更改定位模式为相对定位、绝对定位或者固定定位
 - position : static/relative/absolute/fixed;
- 偏移属性：实现元素框位置的偏移
 - top/bottom/right/left: value;
- 堆叠顺序
 - z-index : value;



定位方式



相对定位

- 元素仍保持其未定位前的形状
- 元素原本所占的空间仍保留
- 元素框会相对于它原来的位置偏移某个距离
 - 设置垂直或水平位置，让元素相对于它的起点进行移动
- 设置元素为相对定位
 - 首先需要设置position 属性的值为 relative
 - 然后使用left 属性或者right 属性设置水平方向的偏移量
 - 也可以使用top 属性或者bottom 属性设置垂直方向的偏移量



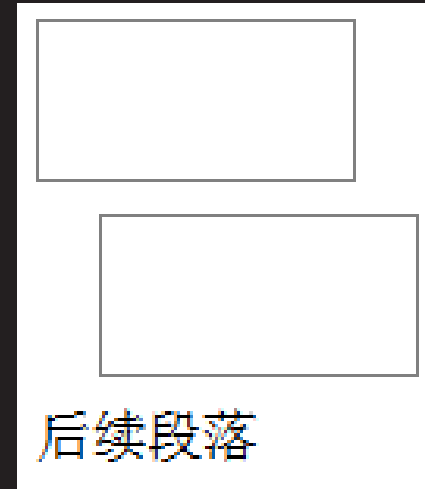
相对定位（续1）

CSS 样式：

```
div
{
    width : 100px;
    height : 50px;
    border : 1px solid gray;
}
```

HTML 文档：

```
<body>
    <div> </div>
    <div style="position:relative;left:20px;top:10px;"> </div>
    <p>后续段落</p>
</body>
```



元素原本所占的空间仍保留

绝对定位

- 将元素的内容从普通流中完全移除，不占据空间
- 并使用偏移属性来固定该元素的位置
 - 相对于最近的已定位祖先元素
 - 如果元素没有已定位的祖先元素，那么它的位置相对于最初的包含块，比如 body 元素
- 设置元素为绝对定位
 - 首先需要设置position 属性的值为 absolute
 - 然后使用left 属性或者right 属性设置元素的水平位置
 - 也可以使用top 属性或者bottom 属性设置元素的垂直位置



绝对定位 (续1)

CSS 样式 :

```
div.parent
{
    width:200px;
    height:150px;
    border:1px solid gray;
}
```

```
div.child
{
    width:80px;
    height:30px;
    border:1px solid black;
    background-color:silver;
}
```

HTML 文档 :

```
<body>
    <div class="parent">
        <div class="child">1</div>
        <div class="child">2</div>
    </div>
</body>
```

没有设置定位属性时



绝对定位 (续2)

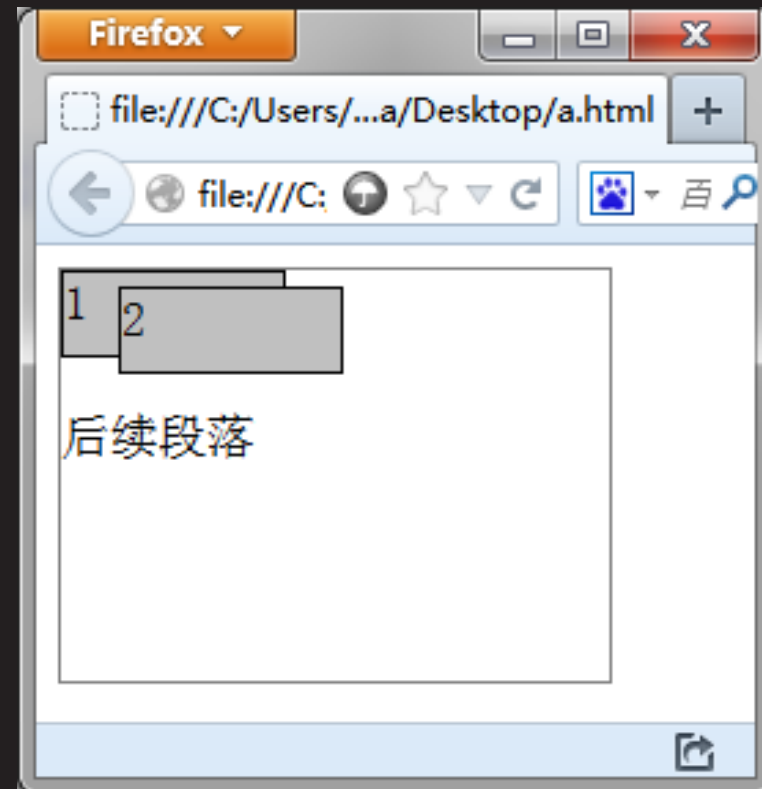
HTML 文档：

```
<body>
  <div class="parent">
    <div class="child">1</div>
    <div class="child"
      style="position : absolute ; top : 15px ; left : 30px ;">2</div>
  </div>
</body>
```

设置为绝对定位

知识讲解

元素框不再占用原有位置，会产生重叠
相对于最近的已定位祖先元素定位
如果没有已定位的祖先元素，那么它的位置
相对于最初的包含块，比如 body 元素



绝对定位 (续3)

CSS 样式 :

```
div.parent  
{  
    width : 200px;  
    height : 150px;  
    border : 1px solid gray;  
    position : relative;  
}
```

知识讲解

会根据父元素 <div> 的位置进行定位
因为父元素设置了定位属性



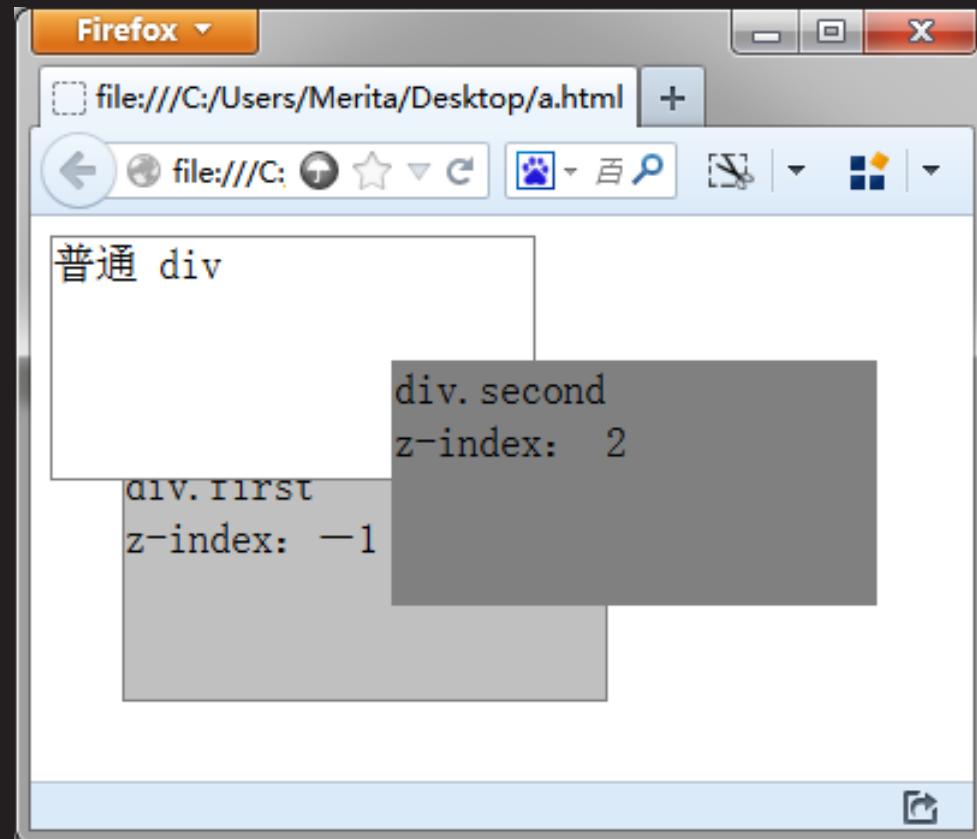
堆叠顺序

- 一旦修改了元素的定位方式，则元素可能会发生堆叠
- 可以使用z-index 属性来控制元素框出现的重叠顺序
- z-index 属性
 - 值为数值：数值越大表示堆叠顺序更高，即离用户更近
 - 拥有更高堆叠顺序的元素总是会处于堆叠顺序较低的元素的前面，即元素的显示会接近页面表面
 - 可以设置为负值：表示离用户更远



堆叠顺序（续1）

```
div {
  z-index:100;
  width:200px;
  height:100px;
  border:1px solid gray;
  background-color:White; }
div.first {
  position:relative;
  top:-10px;
  left:30px;
  z-index:-1;
  background-color:Silver; }
div.second {
  position:absolute;
  top:60px;
  left:150px;
  z-index:2;
  background-color:gray; }
```



固定定位

- 将元素的内容固定在页面的某个位置
 - 元素从普通流中完全移除，不占用页面空间
 - 当用户向下滚动页面时元素框并不随着移动
- 设置固定定位
 - 首先需要设置position 属性的值为 fixed
 - 通过 left、top、right 以及bottom 这些偏移属性来定义元素的位置



定位属性的应用

- 【参见 COOKBOOK】
 - 使用定位属性为页面元素定义样式



总结和答疑

