

mui组件通用CSS类

color(颜色)

mui中组件以iOS 7为基础，补充部分Android特有控件，颜色值主要有以下五种：

```
primary => #007aff; //蓝色(blue)
success => #4cd964; //绿色(green)
warning => #f0ad4e; //黄色(yellow)
danger  => #dd524d; //红色(red)
royal   => #8a6de9; //紫色(purple)
```

注：上述primary等颜色值关键词在引用的时候可以替换成对应的颜色单词，如 `.mui-btn-primary` 等效于 `.mui-btn-blue`；

badge (数字角标)：默认背景为灰色，另外mui内置了蓝色 (primary/blue)、绿色 (success/green)、黄色(warning/yellow)、红色(danger/red)、紫色 (royal/purple) 五种颜色，通过给button或者a标签添加 `.mui-badge-*` 类添加背景颜色。如：

```
<span class="mui-badge">1</span>
<span class="mui-badge mui-badge-primary">2</span>
<span class="mui-badge mui-badge-success">3</span>
<span class="mui-badge mui-badge-warning">4</span>
<span class="mui-badge mui-badge-danger">5</span>
<span class="mui-badge mui-badge-royal">6</span>
```

button (按钮)：默认背景为白色，另外mui内置了蓝色 (primary/blue)、绿色 (success/green)、黄色(warning/yellow)、红色(danger/red)、紫色 (royal/purple) 五种颜色，通过给button或者a标签添加 `.mui-btn-*` 类添加背景颜色。

```
<button type="button" class="mui-btn">默认</button>
<button type="button" class="mui-btn mui-btn-primary">蓝色</button>
<button type="button" class="mui-btn mui-btn-success">绿色</button>
<button type="button" class="mui-btn mui-btn-warning">黄色</button>
<button type="button" class="mui-btn mui-btn-danger">红色</button>
<button type="button" class="mui-btn mui-btn-royal">紫色</button>
```

switch(开关)：默认值为绿色，可选颜色为蓝色，添加 `.mui-switch-blue` 类即可。

```
<div class="mui-switch mui-active">
  <div class="mui-switch-handle"></div>
</div>
<div class="mui-switch mui-switch-blue mui-active">
  <div class="mui-switch-handle"></div>
</div>
```

任意元素：可以给任意元素添加.mui-bg-*类添加背景颜色，可选参数为primary/positive/negative。

```
<div class="mui-bg-primary">我是蓝色背景</div>
<div class="mui-bg-positive">我是绿色背景</div>
<div class="mui-bg-negative">我是红色背景</div>
```

文字(font)

文字位置：可以给文字添加 .mui-text-left、.mui-text-center、.mui-text-right 设置文字位置。

```
<div class="mui-text-left">我在左边</div>
<div class="mui-text-center">我在中间</div>
<div class="mui-text-right">我在右边</div>
```

溢出隐藏：当文字内容超过一行或者多行时，溢出文本用省略号代替。

显示单行：

```
<p class="mui-ellipsis">我在一行内，当我超出了会显示省略号。我在一行内，当我超出了会显示省略号。我在一行内，当我超出了会显示省略号。我在一行内，当我超出了会显示省略号。</p>
```

显示两行：

```
<p class="mui-ellipsis-2">我在两行内，当我超出了会显示省略号。我在两行内，当我超出了会显示省略号。我在两行内，当我超出了会显示省略号。我在两行内，当我超出了会显示省略号。我在两行内，当我超出了会显示省略号。</p>
```

若需要设置显示多行，通过设置-webkit-line-clamp属性，如显示三行添加style="-webkit-line-clamp:3":

`<p class="mui-ellipsis-2" style="-webkit-line-clamp:3">`我在三行内，当我超出了会显示省略号。我在三行内，当我超出了会显示省略号。我在三行内，当我超出了会显示省略号。`</p>`

基本整体布局

```
<header class="mui-bar mui-bar-nav">
  <a class="mui-action-back mui-icon mui-icon-left-nav mui-pull-left"></a>
  <h1 class="mui-title">标题</h1>
</header>
<footer class="mui-bar mui-bar-footer">
  底部
</footer>
<div class="mui-content">
  主体
</div>
```

```

<nav class="mui-bar mui-bar-tab">
  <a class="mui-tab-item mui-active">
    <span class="mui-icon mui-icon-home"></span>
    <span class="mui-tab-label">首页</span>
  </a>
  <a class="mui-tab-item">
    <span class="mui-icon mui-icon-phone"></span>
    <span class="mui-tab-label">电话</span>
  </a>
  <a class="mui-tab-item">
    <span class="mui-icon mui-icon-email"></span>
    <span class="mui-tab-label">邮件</span>
  </a>
  <a class="mui-tab-item">
    <span class="mui-icon mui-icon-gear"></span>
    <span class="mui-tab-label">设置</span>
  </a>
</nav>

```

在使用时需要注意一下两个细节：

固定栏靠前：所谓的固定栏，也就是带有 `.mui-bar` 属性的节点，都是基于fixed定位的元素；常见组件包括：顶部导航栏（`.mui-bar-nav`）、底部工具条（`.mui-bar-footer`）、底部选项卡（`.mui-bar-tab`）；这些元素使用时需遵循一个规则：放在 `.mui-content` 元素之前，即使是底部工具条和底部选项卡，也要放在 `.mui-content` 之前，否则固定栏会遮住部分主内容。

一切内容都要包裹在mui-content中：除了固定栏之外，其它内容都要包裹在 `.mui-content` 中，否则就有可能被固定栏遮罩，原因：固定栏基于Fixed定位，不受流式布局限制，普通内容依然会从 `top:0` 的位置开始布局，这样就会被固定栏遮罩，mui为了解决这个问题，定义了如下css代码：

```

.mui-bar-nav ~ .mui-content {
  padding-top: 44px;
}
.mui-bar-footer ~ .mui-content {
  padding-bottom: 44px;
}
.mui-bar-tab ~ .mui-content {
  padding-bottom: 50px;
}

```

你当然可以通过自定义CSS的方式实现如上类似效果，但为了使用简便，建议将除固定栏之外的所有内容，全部放在 `.mui-content` 中。

局部元素布局

设置边距：可以给块元素设置 `.mui-content-padded` 类添加边距，默认为10px外边距。

设置元素为内联块对象：可以给元素设置 `.mui-inline` 类，将元素呈递为内联对象，但是对象的内容作为块对象呈递。

```
<div class="mui-content-padded">
  <div class="mui-inline">我是A</div>
  <div class="mui-inline">我是B</div>
  <div class="mui-inline">我是C</div>
</div>
```

设置元素为块元素对象：可以给元素设置元素 `.mui-block`，将元素设置为块元素。

浮动：可以给元素设置 `.mui-pull-left` 或者 `.mui-pull-right` 类，从而设置浮动效果，可以使用 `.mui-clearfix` 清除浮动。

```
<div class="mui-clearfix">
  <div class="mui-pull-left">我在左边</div>
  <div class="mui-pull-right">我在右边</div>
</div>
```

块级表格：mui中使用CSS表格的方式定义了 `.mui-table` 块级表格布局方式，会作元素为块级表格来显示。

```
<div class="mui-table">
  <div class="mui-table-cell mui-text-center">
    我是A
  </div>
  <div class="mui-table-cell mui-text-center">
    我是B
  </div>
  <div class="mui-table-cell mui-text-center">
    我是C
  </div>
</div>
```

列表去掉默认样式：可以给ul元素设置 `.mui-list-unstyled` 类去掉ul标签在浏览器中默认的原点和内边距。

```
<ul class="mui-list-unstyled">
  <li>我是第1条</li>
  <li>我是第2条</li>
  <li>我是第3条</li>
</ul>
```

列表显示为内联元素：可以给ul元素设置 `.mui-list-inline` 类将li标签元素设置为内联块元素。

```
<ul class="mui-list-inline">
  <li>我是第1条</li>
  <li>我是第2条</li>
  <li>我是第3条</li>
</ul>
```

设置元素显示与隐藏： `.mui-visibility` 默认为设置元素可见，`.mui-hidden` 默认为设置元素隐藏。

```
<div id="loading">
  loading...
</div>

<script type="text/javascript">
  setTimeout(function(){
    document.getElementById("loading").classList.add('mui-hidden');
  },3000)
</script>
```

OS环境判断多平台适配：

mui会通过 `mui.os.*` 方法判断环境，将 `.mui-plus`，`.mui-plus-stream`，`.mui-ios`，`.mui-android`，`.mui-wechat`，`.mui-ios-version`，`.mui-android-version`，`.mui-wechat-version` 绑定在 `document.body.classList` 中，我们可以通过这些样式类判断当前的运行判断，于是可以做出一些适配，如：

```
<div class="mui-input-row mui-plus-visible">
  <label>mui-plus-visible</label>
  <input type="text" class="mui-input-speech mui-input-clear" placeholder="我在web环境下隐藏5+环境下显示">
</div>
<div class="mui-input-row mui-plus-hidden">
  <label>mui-plus-hidden</label>
  <input type="text" class="mui-input-clear" placeholder="我在web环境下显示5+环境下隐藏">
</div>
```

mui中默认在plus环境下和微信环境下设置了样式几个样式：

- .mui-plus-visible：在plus环境下显示，非plus环境下隐藏
- .mui-wechat-visible：在wechat环境下显示，非wechat环境下隐藏
- .mui-plus-hidden：在plus环境下隐藏，非plus环境下显示
- .mui-wechat-hidden：在wechat环境下隐藏，非wechat环境下显示

栅格系统布局：mui中定义了一个简单适用的栅格系统，将每一行宽度平均分为12份，每一份作为一个子栅格，每一行的内容置于 `.mui-row` 行容器中，通过 `.mui-col-xs-*` 和 `.mui-col-sm-*` 将行分成若干行。使用以下媒体查询（media query）将 `.mui-row` 像素宽度400px作为分界，`.mui-row` 像素宽度低于400px的使用 `.mui-col-xs-*`，当 `.mui-row` 像素宽度高于400px使用 `.mui-col-sm-*`。`.mui-row` 宽度若不设置，默认为屏幕像素宽度。

通过给每一行中的列设置1~12的数值，相应列的宽度会随着 `.mui-row` 像素宽度变化。若一行中列的数值和大于12，多余的列所在元素会作为一个整体另起一行排列。

```
<style type="text/css">
  .mui-row{
    height: 50px;
    line-height: 50px;
    text-align: center;
  }
  .mui-col-sm-3{
    border: 1px solid #aaa;
  }
  .mui-col-sm-9{
    border: 1px solid #aaa;
  }
</style>
<div class="mui-row">
  <div class="mui-col-xs-4 mui-col-sm-3">
    .mui-col-sm-3
  </div>
  <div class="mui-col-xs-8 mui-col-sm-9">
    .mui-col-sm-9
  </div>
</div>
```

这样我们得到了两列元素，当 `.mui-row` 像素宽度低于400px时，左侧宽度为4份子栅格宽度，右侧宽度为8份子栅格宽度；当 `.mui-row` 像素宽度高于400px时，左侧宽度为4份子栅格宽度，右侧宽度为8份子栅格宽度。