全局CSS样式

地址： <address>

<strong>加粗字体</strong>

</addresss>

无序列表：<ul class="list-unstyled">==>没有样式（圆点）

<li></li>

<ul>

省略 <abbr title="标识">内容</abbr>

p标签: <p class="lead"></p>字体大小是14px,行高是10px的外边距 lead==>可以让字体突出显示

mark: <mark>内容</mrak>拥有背景效果

del: <del></del>带有删除线的效果

h1-h6: <h1><small>得到一个颜色相对来说颜色浅的，字体小的字体</small></h1>

文本的居中：class="text-align"

文本的居右：class="text-right"

文本的居左：class="text-left"

文本的大写：class="text-uppercase"

文本的小写：class="text-lowercase"

文本的首字母大写：class="text-capitalize"

Bootstrap CSS栅格

Containers:栅格系统是依赖于系统存在的

<div class="container">：大部分使用

</div>

或

<div class="container-fluid">100%的宽度占据视口的主要容器

</div>

栅格：分成12列 行row/列

手机端<768px ipad>=768px 小的显示器>=992px 大的显示器>=1200px

.col-xs- .col-sm- .col-md- .col-lg-

.col-xs-1~12 .col-sm-1~12 .col-md-1~12 .col-lg-1~12

偏移：col-md-offset-数字大小

位置进行调换：cd-md-push-数字大小

cd-md-pull-数字的大小

code在网页上面展示代码

内联代码：在网页上面显示出来<内容> 列子：<code>&lt;section&gt;</code>

输入的样式：<kbd>内容</kbd>黑色包裹起来的边框

代码块进行包裹：<pre>内容</pre>灰色的边框

数学字符：<var>x</var>=<var>y</var>+<var>z</var>==>页面上x=y+z

输出：<samp>内容</samp>-->直接输出内容

表格 Bootstrap CSS表格

格式： <table class="table(表格) table-striped(斑马线表格) table-bordered(边框) table-hover(鼠标经过背景颜色发生变化) table-condensed(让每个边框的距离变得小一点)">

<thead>

<tr class="active(鼠标悬停的)">

<th></th>

</tr>

</thead>

<tbody>

<tr class="success">

<td></td>

<td></td>

<td></td>

</tr>

<tr class="info">

<td></td>

<td></td>

<td></td>

</tr>

<tr class="danger">

<td></td>

<td></td>

<td></td>

</tr>

</tbody>

</table>

如果想要是响应式的在最外面那层<div class="table-responsize">

form表单

<form class="form-inline(内联格式显示在一排默认是竖着排列//form-horizontal水平的)" role="form(扮演的角色)">

<div class="form-group">

<lable class="sr-only(进行label的隐藏)/想要文字对齐选择12栅格进行分化位置col-sm-2/label-control">姓名/密码</lable>===>必须的

<div class="col-sm-offset-2(移动两个栅格从最左面开始) col-sm-10(占12栅格的10份)">

<input type="password/file/email" class="form-control(表单的控制-->为了让表单的长度更长还有原角的效果)" placeholder="提示性文字"/>

<button type="submit" class="btn btn-default">提交</button>

</div>

</div>

</form>

<p class="help-block">内容文字</p>==>用于指引式文字

文本域：<textarea class="form-control" row=5></textarea>

复选框：<div class="checkbox">

<lable>

<input type="checkbox" value=""/>

文字

</label>

</div>

单选框：<div class="radio">

<label>

<input type="radio" name="" checked--默认选中状态/>

文字

</label>

</div>

下拉列表：<select class="form-control/在整个列表中出现multiple">

<option>1</option>

<option>2</option>

<option>3</option>

</select>

纯静态的文本：<p class="form-control-static">描述性文字</p>

禁用状态：<input type="text" class="form-control" placeholder="请输入" disabled(禁用) readonly(只读)/>

如果是整个表单全部禁用的情况：<fieldset disabled></fieldset>包裹全部

如果想要input的边框有颜色在（父级上面添）加：class="has-success"

Bootstrap图片和按钮

按钮：<button type="button" class="btn-->代表它是按钮/btn-default-->颜色/btn-lg/sm/xs(按钮的大小)"></button>

a链接：<a href="#" class="btn btn-default btn-lg active--激活的状态" role="button" disabled="disabled"---禁用的状态></a>

图片：<img src="" alt="" class="img-circle(圆角)/img-rounded(正方形)img-thumbnail(外面还有一个边框)"/>

下拉菜单

定义：<div class="dropdown(下拉菜单)">

<button class="btn btn-default dropdown-toggle(可以点击多次) data-toggle="dropdown(显示下拉列表)"></button>

</div>

下拉列表的内容：class="dropdown-menu" role="menu" role="menuitem(菜单的子项)" dropdown-menu-right(在右侧出现下拉菜单)

头文件进行分割：<li class="dropdown-header"></li>

分割线：<li class="divider"></li>

想要让三个按钮充满整个屏幕：(在父类添加)btn-group-justified

8.分页

分页（Pagination）

下表列出了 Bootstrap 提供的处理分页的 class。

Class 描述 示例代码

.pagination 添加该 class 来在页面上显示分页。

<ul class="pagination">

<li><a href="#">&laquo;</a></li>

<li><a href="#">1</a></li>

.......

</ul>

.disabled

.active

您可以自定义链接，通过使用 .disabled 来定义不可点击的链接，通过使用 .active 来指示当前的页面。

<ul class="pagination">

<li class="disabled"><a href="#">&laquo;</a></li>

<li class="active"><a href="#">1<span class="sr-only">(current)</span></a></li>

.......

</ul>

.pagination-lg

.pagination-sm

使用这些 class 来获取不同大小的项。

<ul class="pagination pagination-lg">...</ul>

<ul class="pagination">...</ul>

<ul class="pagination pagination-sm">...</ul>

翻页（Pager）

与分页链接一样，翻页也是无序列表。默认情况下，链接是居中显示。下表列出了 Bootstrap 处理翻页的 class。

Class 描述 示例代码

.pager 添加该 class 来获得翻页链接。

<ul class="pager">

<li><a href="#">Previous</a></li>

<li><a href="#">Next</a></li>

</ul>

.previous, .next 使用 class .previous 把链接向左对齐，使用 .next 把链接向右对齐。

<ul class="pager">

<li class="previous"><a href="#">&larr; Older</a></li>

<li class="next"><a href="#">Newer &rarr;</a></li>

</ul>

.disabled 添加该 class 来设置对应按钮禁止使用。

<ul class="pager">

<li class="previous disabled"><a href="#">&larr; Older</a></li>

<li class="next"><a href="#">Newer &rarr;</a></li>

</ul>

1. 三角符号  
   caret //css实现向下小三角

边框

增加边框

border //默认：1px solid #dee2e6!important

border-top

border-left

border-right

border-bottom

边框颜色

border-primary

border-secondary

border-success

border-danger

border-warning

border-info

border-light

border-dark

border-white

删除边框

border-0 //去除边框或者除去某一边的边框

border-top-0

border-left-0

border-right-0

border-bottom-0

圆角边框

rounded //border-radius: .25rem!important;

rounded-top //只有顶部的两边有圆角

rounded-right

rounded-bottom

rounded-left

rounded-circle 类可以设置椭圆形图片

rounded-0

display显示

d-none//隐藏

d-inline//内联

d-inline-block//内联块

d-block//显示

d-table//表格

d-table-cell//表格

d-table-row

d-flex

d-inline-flex

flex布局

弹性布局  
.d-flex //将对象作为弹性伸缩盒显示

.d-inline-flex //将对象作为内联块级弹性伸缩盒显示

.d-sm-flex //对应尺寸加载

d-sm-inline-flex

d-md-flex

d-md-inline-flex

d-lg-flex

d-lg-inline-flex

d-xl-flex

d-xl-inline-flex

方向 水平

.flex-row设置在水平方向（在浏览器默认值）

flex-row-reverse //相反方向

方向垂直

.flex-column设定垂直方向

flex-column-reverse

响应式变体flex-direction

flex-row//水平排列

flex-row-reverse//反向的水平排列

flex-column//垂直排列

flex-column-reverse//反向的垂直排列

flex-sm-row//小屏的水平排列

flex-sm-row-reverse//小的屏水平反向排列

flex-sm-column//小屏垂直

flex-sm-column-reverse//小的屏垂直反向

flex-md-row

flex-md-row-reverse

flex-md-column

flex-md-column-reverse

flex-lg-row

flex-lg-row-reverse

flex-lg-column

flex-lg-column-reverse

flex-xl-row

flex-xl-row-reverse

flex-xl-column

flex-xl-column-reverse

对齐方式

justify content //可匹配到不同尺寸

justify-content-start

justify-content-end

justify-content-center

justify-content-between

justify-content-around

align items //子元素设置 支持不同尺寸匹配

align-items-start

align-items-end

align-items-center

lign-items-baseline

align-items-stretch

Align self//子元素设置 支持不同尺寸匹配

align-self-start

align-self-end

align-self-center

align-self-baseline

align-self-stretch

Align content//子元素设置 支持不同尺寸匹配ß

align-content-start

align-content-end

align-content-center

align-content-around

align-content-stretch

.flex-fill//填充