# 2019.4.20

## 防止发生不明不白的错误：

**将所有元素标签内外边距调为0**

\*{

Margin:0px;

Padding:0px;

}

### <!--代替div的结构元素-->

<section></section> <!-- 内容块，章节，页眉，页脚，页面中其它部分 -->

<article></article> <!-- 上下文不相关的独立内容 -->

<aside></aside> <!-- 与article相关的辅助信息 -->

<header></header> <!-- 整个页面标题，我感觉放banner合适 -->

<footer></footer> <!-- 页面底部 -->

<nav></nav> <!-- 导航栏 -->

<main></main> <!-- 网页主要内容区域 -->

<figure> <!-- 独立的流内容,一般表示主体流内容的一个独立单元 -->

<figcaption>GRC</figcaption>

<p>Tje asd asdsadas asdasd asdasdas</p>

</figure>

### <!--具有布尔类型的属性标签,设定真假方式-->

#### <!--true-->

<input type="checkbox" checked > <!-- 只写属性为真 -->

<input type="checkbox" checked="checked"> <!-- 属性="属性"为真 -->

<input type="checkbox" checked=""> <!-- 属性=""空为真 -->

#### <!--false-->

<input type="checkbox"> <!-- 不写为假 -->

## HTML5新增标签

### <!-- HTML5中新增元素 -->

#### <!-- 1.建立播放视频/音频流文件 -->

<!—video,audio>

<video src="E:/迅雷下载/西虹市首富.mp4" controls style=" width:1440px; height:500px; background:#000; " >test</video> <!—video中controls显示控制栏,src索引视频文件-->

<br />

<audio src="music/GILL.mp3" controls >audio元素</audio><!—音频索引-->

<br />

#### <!-- 2.插入多媒体文件,自动播放多媒体文件,视频无法全屏 -->

<!--embed-->

<embed src="E:/迅雷下载/西虹市首富.mp4" /> <!—可以使用controls属性加入控制栏 -->

#### <!-- 3.突显高亮文字,mark代替span -->

<!--mark-->

<mark>我卡啊</mark>

<br />

#### <!-- 4.显示javascript耗费时间的函数进程 -->

<!--progress-->

<progress>

<script type="text/javascript"> var a,b; a=0; b=1; a+b; </script>

</progress>

<br />

#### <!-- 5.规定数值范围 -->

<!--meter-->

<meter><p style="width:200px; height:100px;" ></p></meter>

<br />

#### <!-- 6.时间 -->

<!—time-->

<time></time>

<br />

#### <!-- 7.ruby创建有发音符的字符串,rt发音 -->

<!—ruby,rt,rp-->

<ruby>你<rt>1</rt></ruby>

<rp></rp> <!-- 放在ruby里,定义不支持ruby的浏览器所显示的内容 -->

<br />

#### <!-- 8.软换行,宽度足够时不换行，宽度不够时进行换行,对中文用处不大 -->

<!--wbr-->

<p style="width:50px;" >

123213123<wbr>asdasdasdasd

</p>

<br />

#### <!-- 9.canvas本身没有行为,给javascript提供一片画布，进行绘制图形 -->

<!--canvas-->

<canvas id="myCanvas" width="200" height="100px" >1</canvas>

<br />

#### <!-- 10.建立单选复选框，目前大多浏览器不支持,暂且不谈 -->

<command></command>

<br />

#### <!-- 11.折叠显示信息 -->

<!—details,summary-->

<details>

<summary>HTML5</summary> <!-- 标题 -->

ASDASDASDASDASDSADASDASDASDASDAS ASDASDASDSA ASDASD . <!-- 折叠信息 -->

</details>

<br />

#### <!-- 12.内容框,下拉选项创建 -->

<!—input,datalist,option-->

<form>

<input list="abc" name="abc">

<datalist id="abc" > <!-- 创建下拉选项 -->

<option value="1"> <!-- 选项内容 -->

<option value="2">

</datalist>

<input type="submit">

</form>

<br />

#### <!-- 13.HTML5中已经删除 -->

<datagrid>

　　<ol>

　　　　<li> (datagrid row 0) </li>

　　　　<li> (datagrid row 1)

　　　　　　<ol style="list-style-type:lower-alpha;">

　　　　　　　　<li> (datagrid row 1,0) </li>

　　　　　　　　<li> (datagrid row 1,1) </li>

　　　　　　</ol>

　　　　</li>

　　　　<li> (datagrid row 2) </li>

　　</ol>

</datagrid>

<br />

#### <!-- 14.keygen生成密码,HTML5中已经删除 -->

<keygen>

<br />

#### <!-- 15.output输出计算结果 -->

<!--<form oninput=””>,output-->

<form oninput=" x.value = parseInt(a.value) \* parseInt(b.value) " > <!-- oninput与output联用,parseInt()转换为整数类型,id.value能索引出目标的value值 -->

<input id="a" type="range" value="10" >

<input id="b" type="number" value="10" style="width:50px;" > =

<output name="x" for="a b"></output> <!-- 输出x值,for属性描述计算关系,用空格隔开 -->

</form>

<br />

#### <!-- 16.source设置多个播放源,可以使用到流媒体播放标签,如: audio,video,embed -->

<!-- source -->

<audio controls>

<source src="music/GILL.ogg" type="audio/ogg">

<source src="music/GILL.mp3" type="audio/mp3">

</audio>

<br />

#### <!-- 17.已经删除 -->

<menu>

<li><input type="checkbox">Red</li>

<li><input type="checkbox">Blue</li>

</menu>

<br />

#### <!-- 18.dialog弹窗交互,目前只要Chrome和Safari支持 -->

<!--dialog-->

<dialog open style=" width:90%; height:70%; " > <!-- open属性表示显示弹窗,默认不显示 -->

<div style="width:100px; height:100px; background:#000;" ></div>

</dialog>

<div style="width:100px; height:100px; background:red;" ></div> <!-- dialog弹窗不影响排序 -->

# 2019.4.21

<form>

## <!-- 新增input元素 -->

**<input type="email">**

**<input type="url">**

**<input type="number">**

**<input type="range">**

**<input type="Date Pickers">**

**<br/>**

### <!-- 新增属性 -->

**<input type="text" autofocus> <!-- autofocus打开页面自动获取焦点 --> <br/>**

**<input type="text" placeholder="请输入数字"> <!-- placeholder在文本框显示提示字符串 --> <br/>**

#### <!-- fieldset指向表单，fieldset内容也属于指向的表单，大部分浏览器不支持该属性 -->

**<form id="name1">**

**<input type="text">**

**</form>**

**<br/>**

**<fieldset form="name1">**

**<input type="text">**

**</fieldset>**

#### <!-- required强制填写属性( 必填 ) -->

**<br/>**

**<form action="">**

**<input type="text" required> <!—如果不填写,则无法进行提交表单 -->**

**<textarea rows="4" cols="50" name="comment" required> </textarea><!—文本框-->**

**<input type="submit">**

**</form>**

## <p> ----------- </p>

## <!-- input详细属性解析 -->

### <!--type属性-->

#### <!-- 按钮类型 -->

**<input type="button"> <br/> <!-- 定义可单击按钮 -->**

**<input type="chekbox"> <br/> <!-- 复选框 -->**

**<input type="file"> <br/> <!-- 上传文件 -->**

**<!--**

**accept: 规定上传文件类型**

**accept="audio/\*" 音频**

**accept="image/\*" 图像**

**accept="video/\*" 视频**

**accept="MIMI\_type"**

**multiple: 设定允许多个元素上传**

**-->**

**<input type="image" alt="123"> <br/> <!-- 定义图像为提交按钮 -->**

**<!--**

**alt="无法加载图片时显示"**

**src="索引图像位置"**

**width="设定宽度"**

**-->**

**<input type="radio"> <br/> <!-- 单选框 -->**

**<input type="range"> <br/> <!-- 滑动条控制数字 -->**

**<input type="reset"> <br/> <!-- 重置按钮 -->**

**<input type="submit"> <br/> <!-- 提交按钮 -->**

#### <!-- 时间类型 -->

**<input type="datetime"> <br/> <!-- 年月日时间,基于时区 -->**

**<input type="date"> <br/> <!-- 年月日 -->**

**<input type="datetime-local"> <br/> <!-- 年月日时间,基于本地 -->**

**<input type="mouth"> <br/>**

**<input type="time"> <br/> <!-- 定义输入时间 -->**

**<input type="week"> <br/> <!-- 定义周期和年 -->**

#### <!-- 输入类型 -->

**<input type="email"> <br/>**

**<input type="number"> <br/> <!-- 定义输入数字字段 -->**

**<input type="password"> <br/> <!-- 密码 -->**

**<input type="search"> <br/> <!-- 搜索字符串 -->**

**<input type="tel"> <br/> <!-- 定义电话号码输入的字段 -->**

**<input type="text"> <br/> <!-- 定义单行文本字段 -->**

**<input type="url"> <br/> <!-- 输入链接 -->**

#### <!-- 其它类型 -->

**<input type="hidden"> <br/> <!-- 定义隐藏类型，浏览器不渲染，且不占用布局空间 -->**

**<input type="color"> <br/> <!-- 拾色器 -->**

### <!-- input适用多属性 -->

<!--

#### 1. autocomplete自动补全开启/关闭

**适用: 输入的type类型**

**autocomplete="off"/"on"**

#### 2. autofocus打开页面直接获取焦点

适用: 所有input类型

#### 3. checked预选

适用: type="checkbox"/"radio"

#### 4. disabled禁用input属性

适用: 所有input类型

#### 5. form指定表单id，设定input元素归属

适用: 所有input类型

大多浏览器不支持

#### 6. formaction设定表单提交链接

**formmethod设定提交方法( 会覆盖form的method属性值 )**

**formenctype设定传输数据编码形式**

**适用: type="submit"/"image"**

**formaction="URL": 注意formaction会覆盖掉<form action="URL">action属性值**

**formmethod="get"/"post" 二种提交表单时的传输方式推荐使用post**

#### 7. formnovalidate不验证提交

**适用: type="submit"/"image"**

**<form>**

**<input type="email">**

**<input type="submit"> 验证提交**

**<input type="submit" formNoValidate> 不验证提交**

**</form>**

#### 8. list属性,与datalist配合形成下拉选项

**适用: 输入类型属性**

**<input list="name1">**

**<datalist id="name1">**

**<option value="1">**

**<option value="2">**

**</datalist>**

#### 9. min/max限定数值大小

适用: 时间类型,数子类型

<input type="range" min="1" max="5">

#### 10. maxlength规定字符串长度

适用: 输入类型

<input type="text" maxlength="10"> 设定输入字符串数量不超过10个

#### 11. name设定input元素名称

**有name可以在javascript中引用input值,并且可以提交表单**

**如果没有name则input内容不会被提交**

**<form action="xx.php" method="post">**

**<input type="text" name="name1">**

**<input type="submit">**

**</form>**

-->

<br/>

#### <!-- 12. pattern:内容进行正则验证 -->

**<!-- 适用: 输入类型 -->**

**<!-- title全局属性,当鼠标移动到元素标签时，进行信息提示 -->**

**<form action="xx.php" method="post">**

**<input type="text" required name="test1" pattern="[A-Za-z]{5}" title="5个字母" >**

**<input type="submit">**

**</form>**

**<br/>**

#### <!-- 13. placeholder: 在文本框内显示提示字符串 -->

**<!-- 适用: 输入类型-->**

**<form action="xx.php" method="post" >**

**<input type="text" required placeholder="TEST\_233" >**

**<input type="submit">**

**</form>**

<!--

#### 14. readonly: 设定字段为只读状态

适用: 字段类型的input

#### 15. size: 设置input宽度

适用: 大部分input类型

#### 16. step: 设置数字间隔

适用: 输入类型

<input type="text" step="3"> step为3则能输入的为-3,0,3,6依次类推

-->

<br/>

#### <!-- 17. 关于value属性 -->

**<form action="xx.php" method="post">**

**<!-- value使用在按钮类型，按钮显示的文本如: button,submit,reset -->**

**<input type="button" value="111" >**

**<!-- value使用在输入类型上,为默认值如: text,password,hidden -->**

**<input type="text" value="233">**

**<!-- value在checkbox,radio,image的值会通过action发送到URL中 -->**

**<!-- 注意,checkbox,radio是必须写入value属性的 -->**

**<input type="checkbox" value="男">**

**<input type="radio" value="女">**

**</form>**

<br/>

#### <!-- 关于form属性 -->

<form action="xx.php" method="post"></form> <!-- action="表单提交目标"，method="提交方式" -->

### <!-- 关于a标签与属性 -->

**<!--**

**href="链接的URL"**

**download="指定下载文件路径"**

**rel="写明链接到的URL与本页的关系,方便搜索引擎录取"**

**target="打开方式"**

**media="规定链接的URL媒体类型"**

**type="规定URL,MIME类型"**

**-->**

#### <!-- 下载图片链接 -->

<a href="images/1.jpg" download="images/1.jpg" > image\_download</a>

<br/>

#### <!-- 测试target -->

**<!--**

**\_blank: 新窗口打开**

**\_self: 同窗口打开**

**\_parent/\_top: 不太了解,发现于\_self没有区别**

**-->**

**<a href="http://www.oo7.fun" target="\_blank">塔式甜点\_blank</a>**

**<a href="http://www.oo7.fun" target="\_self">塔式甜点\_self</a>**

**<a href="http://www.oo7.fun" target="\_parent">塔式甜点\_parent</a>**

**<a href="http://www.oo7.fun" target="\_top">塔式甜点\_top</a>**

**111**

## **<!-- 全局属性-任何元素都可以使用的属性 -->**

#### **<!-- 1. contentEditable="true/false"变为可编译模式 -->**

**<!--**

**他可以使标签变为直接可以编辑的模式，但是并没有保存功能**

**并且他具有,继承性**

**-->**

**<p contentEditable="true" >**

**啊实打实的**

**</p>**

#### **<!-- 2. designMode="on/off" 使全页面开启/关闭可编辑模式 -->**

**<!--**

**注意: 只能在js中使用**

**document.designMode="on/off"**

**-->**

**<script type="text/javascript"> document.designMode="off" // 开启可编辑模式 </script>**

**<p> 1111 </p>**

#### **<!-- 3. hidden隐藏标签-->**

**<!--**

**hidden告诉浏览器不进行渲染此标签,在未设定显示时,不占用布局空间**

**通常可以配合js来显示或者不显示**

**-->**

**<p hidden > 隐藏起来 </p>**

**<p> 正常显示 </p>**

#### **<!-- 4. spellcheck专门为input的text,textarea类型设定的,拼写语法错误检查 -->**

**<textarea rows="10" cols="10" spellcheck="true" ></textarea>**

**经测试在chrome不行**

#### **<!-- 5. tabindex获取焦点 -->**

**<!--**

**tabindex="顺序数字",在使用tab键逐个切换焦点顺序**

**通常情况下:**

**在HTML5中tabindex可以在任何标签中使用,并使标签为焦点**

**tabindex="-1": tab键将无法锁定,但是可以通过javascript的focus()语句来控制焦点锁定**

**-->**

**<br/>**

**<input type="text" tabindex="1" >**

**<p tabindex="2" >tabindex\_test</p>**

# **2019.4.26**

### text-decoration控制字体下划线

Text-decoration:none; <!--去除下划线-->

#### Font-weight控制字体粗细

Font-weight:lighter; <!—特别细-->

### CSS3检测屏幕大小变换更换样式

// 注意空格,一定要按照标准写

//当检测到屏幕宽度小于800分辨率时,启动样式

@media screen and (max-device-width:800px){

nav ul li a{

font-size:20px;

font-weight:bold;

}

}

### 设置标签最小宽度

Min-width：620px； // 最小宽度为620px

## Flex弹性布局

**<!doctype html>**

**<html>**

**<head>**

**<title>test</title>**

**<meta charset="utf-8">**

**<!--**

### **Flex布局研究总结( 弹性布局 ):**

#### **1. 开启弹性布局 ( display;flex; | display:-webkit-flex; )**

**display:flex;**

**display:-webkit-flex;**

**display:inline-flex;**

#### **2. 控制x/y排序方向 ( flex-direction:; | -webkit-flex-direction:; )**

**flex-direction:row | row-reverse | column | column-reverse;**

#### **3. x对齐 ( justify-content:; | -webkit-justif-content:; )**

**justify-content:flex-start | flex-end | flex-center | space-between | space-around;**

#### **4. y对齐 ( align-items:; | -webkit-align-items:; )**

**align-items:flex-start | flex-end | center | space-between | space-around | stretch;**

#### **5. 换行模式 ( flex-wrap:; | -webkit-flex-wrap:; )**

**flex-wrap:nowrap | wrap | wrap-reverse;**

#### **6. 换行排序 ( align-content:; | -webkit-align-content:; )**

**align-content:flex-start | flex-end | center | space-between | space-around | stretch;**

#### **7. 自定义排序顺序( order )**

**order:number;**

**-webkit-order:number;**

#### **8. 完美剧中( flex弹性布局下的auto作用 )**

**margin:auto;**

**核心属性: flex模式下auto会沾满容器**

#### **9. y对齐单独自身调控 ( align-self:; | -webkit-align-self:; )**

**align-self: flex-start | flex-end | center | space-between | space-around ;**

#### **10. 常用属性 ( initial | inherit )**

**initial: 默认值**

**inherit: 继承父类属性**

#### **11. 等比放大 ( flex-grow:; | -webkit-flex-grow:; )**

**flex-grow: number | initial | inherit ;( 默认为0 )**

#### **12. 等比缩小 ( flex-shrink:; | -webkit-flex-shrink:; )**

**flex-shrink:number;( 默认为1 )**

#### **13. 固定宽度 ( flex-basis:; | -webkit-flex-basis:; )**

**flex-basis: numberPX | inherit ;**

#### **14. 便捷式-控制缩放( flex:; | -webkit-flex:; )**

**flex: flex-grow flex-shrink flex-basis | auto | initial | inherit;**

**-->**

**<style type="text/css" >**

**body{**

**margin:0px;**

**padding:0px;**

**}**

**/\*\***

### **0. 行内元素开启弹性布局:**

**.box{**

**display:inline-flex;**

**}**

**\*\*/**

#### **/\*\* 1. 块级元素开启弹性布局 \*\*/**

**.flexTest{**

**display:flex; /\*\* 开启弹性布局 \*\*/**

**display:-webkit-flex; /\*\* 开启弹性布局,兼任其它浏览器 \*\*/**

**/\*\*direction:rtl; 使内部元素反方向排序\*\*/**

#### **/\*\* 2. flex-direction属性,控制x/y排序方向 \*\*/**

**flex-direction:row;**

**/\*\***

**flex-direction:row | row-reverse | column | column-reverse ;**

**row: 默认重左到右**

**row-reverse: 反方向**

**column: 垂直方向排序**

**column-reverse: 反方向**

**\*\*/**

#### **/\*\* 3. justify-content沿主轴线(main-aix,水平轴)水平对齐 \*\*/**

**justify-content:space-around;**

**-webkit-justify-content:space-around; /\*\* 为兼容性 \*\*/**

**/\*\***

**justify-content: flex-start | flex-end | center | space-between | space-around ;**

**flex-start: 默认重左排序**

**flex-end: 重右侧排序**

**center: 剧中排序**

**space-between:平均分布排序,左右二边不留空隙;**

**space-around:平均分布排序,左右二边留出空袭;**

**\*\*/**

#### **/\*\* 4. align-items属性,控制垂直排序方向\*\*/**

**/\*\***

**align-items: flex-start | flex-end | center | baseline | stretch ;**

**flex-start: 顶部对齐**

**flex-end: 底部对齐**

**center: 上下剧中对齐**

**baseline: 基于第一行文字对齐**

**stretch: 如果高度为auto,则沾满容器高度,相当于height:100%;**

**\*\*/**

**align-items:baseline; /\*\* 基于第一行文字对齐 \*\*/**

**-webkit-align-items:baseline; /\*\* 为兼容性 \*\*/**

**width:1200px;**

**height:500px;**

**background:pink;**

**}**

**.pic{**

**width:100px;**

**height:100px;**

**margin:10px;**

**background:green;**

**}**

**.flexTest2{**

**width:1200px;**

**height:500px;**

**background:#000;**

**display:flex;**

**display:-webkit-flex-;**

#### /\*\* 5. flex-wrap控制换行模式\*\*/

**/\*\***

**flex-wrap: nowrap | wrap | wrap-reverse ;**

**nowrap: 不换行,大于容器会溢出**

**wrap: 自动换行**

**wrap-reverse: 反转换行,这个挺有意思的**

**\*\*/**

**flex-wrap:wrap; /\*\* 自动换行 \*\*/**

**-webkit-flex-wrap:wrap; /\*\* 为兼容性 \*\*/**

#### **/\*\*6. align-content控制flex-wrap行为,控制"多行垂直"的排布 \*\*/**

**/\*\***

**aligin-content:stretch | flex-start | flex-end | space-between | space-around;**

**stretch: 高度设置为auto时,将沾满容器的高度;**

**flex-start: 上方起始位置堆叠**

**flex-end: 下方底部位置开始堆叠**

**center: 高度中间位置开始堆叠**

**space-around: 上下二边有空隙( 空隙大小,为子元素之间高度空隙的一半 )，平均发布**

**space-between: 没有空隙的平均分布**

**\*\*/**

**align-content:flex-start; /\*\* 上方起始开始堆叠 \*\*/**

**-webkit-align-content:flex-start; /\*\* 为兼容性 \*\*/**

**}**

**.pic2{**

**width:300px;**

**height:100px;**

**background:red;**

**margin:10px;**

**}**

#### **/\*\*7. order设置元素排序顺序,小值在前,大值在后 \*\*/**

**/\*\***

**默认排序初始值,为0**

**order也可以设置负数**

**\*\*/**

**.firstTest{**

**order:-1; /\*\* 设置排序顺序 \*\*/**

**-webkit-order:-1; /\*\* 为兼容性 \*\*/**

**}**

#### **/\*\* 8. 一行代码搞定完美剧中,及底层原理解析 \*\*/**

**/\*\***

**在flex模式下"margin:auto;"直接使元素剧中**

**因为flex状态下,auto会自动沾满容器**

**\*\*/**

**.preCenter{**

**width:1200px;**

**height:500px;**

**background:pink;**

**display:flex;**

**-webkit-display:flex;**

**}**

**.pre{**

**width:500px;**

**height:100px;**

**background:#fff;**

**margin:auto; /\*\* 核心代码,完美剧中 \*\*/**

**}**

#### **/\*\*设置自身排序,控制语句将放在子元素中\*\*/**

**.flexTest3{**

**width:1200px;**

**height:500px;**

**background:yellow;**

**display:flex;**

**-webkit-display:flex;**

**align-items:center;**

**-webkit-align-items:center;**

**}**

**.pic3{**

**width:300px;**

**height:300px;**

**background:red;**

**margin:10px;**

**}**

#### **/\*\* 9. align-self设置自身垂直排序 \*\*/**

**/\*\***

**align-self:flex-start | flex-end | center | baseline | stretch;**

**flex-start: y轴顶部**

**flex-end: y轴底部**

**center: y轴中间**

**baseline: y轴方向文字对齐,默认值其实为flex-start**

**stretch: y轴方向沾满容器**

**\*\*/**

**.selfTest{**

**align-self:flex-start;**

**-webkit-align-self:flex-start;**

**}**

**</style>**

**</head>**

**<body>**

**<div class="flexTest" >**

**<div class="pic" ><p style="margin-top:30px;" >1</p></div>**

**<div class="pic" >2</div>**

**<div class="pic" >3</div>**

**<div class="pic" >4</div>**

**</div>**

**<div class="flexTest2" >**

**<div class="pic2">1</div>**

**<div class="pic2">2</div>**

### **<!-- 多个class简洁写法 -->**

**<!--**

**.firstTest{**

**即可调用**

**}**

**-->**

**<div class="pic2 firstTest">3</div>**

**<div class="pic2">4</div>**

**<input class="pic2" style="border:0px ;" type="text"> <!-- 测试行内元素排序 -->**

**</div>**

**<div class="preCenter" >**

**<div class="pre" >111</div>**

**</div>**

**<div class="flexTest3" >**

**<div class="pic3 selfTest"></div>**

**<div class="pic3"></div>**

**</div>**

**</body>**

**</html>**

**<!doctype html>**

**<html>**

**<head>**

**<title>test</title>**

**<meta charset="utf-8">**

**<style type="text/css">**

**.flexTest{**

**width:1200px;**

**height:500px;**

**display:flex;**

**display:-webkit-flex;**

**background:pink;**

**}**

**.items{**

**flex-basis:500px;**

**height:200px;**

**background:#000;**

**margin:10px;**

**}**

#### **/\*\*10. 常用的属性值解析 \*\*/**

**/\*\***

**initial( 默认值的意思 )**

**inherit( 继承指定父类的属性,**

**如父元素文字大小为font-size:15px;**

**子元素可以直接font-size:inherit;**

**)**

**\*\*/**

#### **/\*\*11. flex-grow: 等比放大,值至占满容器**

**number: 等比放大**

**initial: 默认值( 默认值为0 )**

**inherit: 继承父类属性设置**

**.it1{**

**flex-grow:1;**

**}**

**.it2{**

**flex-grow:2;**

**}**

**.it3{**

**flex-grow:1;**

**}**

**.it4{**

**flex-grow:1;**

**}**

**\*\*/**

#### **/\*\*12. flex-basis设置固定值宽度大小\*\*/**

**/\*\***

**flex-basis:固定宽度值,但是依然受flex-grow/flex-shrink影响;**

### **分类指定选择: div:nth-of-type(2){}**

**选择第二个DIV标签**

**.flexTest div:nth-of-type(2){**

**flex-basis:300px;**

**}**

**\*\*/**

#### **/\*\*13. flex-shrink空间不足时进行项目缩小 \*\*/**

**/\*\***

**flex-shrink默认值为1**

**当容器变小时,子元素也会跟着缩小**

**子元素,flex-shrink设置的值越大则缩小值越大( 指宽度 )**

**\*\*/**

**.it1{**

**flex-shrink:1;**

**}**

**.it2{**

**flex-shrink:1;**

**}**

**.it3{**

**flex-shrink:3;**

**}**

**.it4{**

**flex-shrink:3;**

**}**

#### **/\*\*14. flex便捷式控制缩放问题 \*\*/**

**/\*\***

**flex: flex-grow flex-shrink flex-basis | auto | initial | inherit**

**flex:auto; == flex:1 1 auto;**

**flex:none: == flex:0 0 auto;**

**flex:initial; == flex:0 1 auto; ( initial默认值的意思 )**

**flex:inherit;( 非常好用 )**

**\*\*/**

**</style>**

**</head>**

**<body>**

**<div class="flexTest">**

**<div class="items it1">1</div>**

**<div class="items it2">2</div>**

**<div class="items it3">3</div>**

**<div class="items it4">4</div>**

**</div>**

**</body>**

**</html**

# **2019.4.27**

### CSS3检测屏幕大小来执行样式,移动端的检测

**@media screen and (max-device-width:800px){**

**/\*\* 注意空格,一定要写完整,检测到屏幕小于800px时执行此样式 \*\*/**

**nav ul li a{**

**font-size:20px;**

**font-weight:bold;**

**}**

**}**

# **2019.4.29**

### 取消浏览器选中效果

**Outline: none ;**

### **:hover动态联动**

**/\*\* 动态联动 \*\*/ 当鼠标移动到标签时，多个标签同时发生动态效果**

#### **CSS**

**.chooseProduct:hover h3 a{**

**border-bottom:1px dashed #bbb;**

**}**

**.chooseProduct:hover .chooseProductImg{**

**background:url(product\_1g.png) center center no-repeat ;**

**padding-top:1px;**

**}**

#### **HTML**

**<article class="chooseProduct" >**

**<a href="#" class="chooseProductImg" ></a>**

**<h3><a href="#" >高端网站建设</a></h3>**

**<p >品牌官网·电商平台<br/>响应式网站·站群</p>**

**</article>**

### **阴影的使用**

**Text-shadow: h-shadow v-shadow blur color ;**

**box-shadow: h-shadow v-shadow blur color ;**

**h-shadow: 水平位置 ;**

**v-shadow: 垂直位置 ;**

**blur: 阴影距离 ;**

**color: 影音颜色 ;**

# **2019.4.30**

### 换鼠标类型

<span style="cursor:pointer"> <!-- 换成手状 -->

Pointer

</span><br />

### position属性位置

Position:fixed;

Fixed: 固定位置，定位使元素的位置与文档流无关，因此不占据空间。

Relative: 相对定位，占位不变，位置变

Absolute: 绝对定位，定位使元素的位置与文档流无关，因此不占据空间。

### z-index:设置显示图层

z-index:-1; 显示到最低层

z-index:10; 控制显示层,数越大显示越顶w

### 经典的banner写法

**注意banner以背景的方式写到a标签里，这样既可以看又可以链接**

#index\_banner{

margin-top:100px;

background:url(index\_banner1.jpg) center center no-repeat;

background:cover;

width:100%;

height:582px;}

### background一些属性

Background-size: 100% 100%; // 可以控制背景图的大小

Background:cover; // 背景图覆盖标签大小范围

# 2019.5.1

### Border 虚线写法

Border: 1px dashed #000 ;

# 2019.5.2

**<style type="text/css">**

### **/\*\* transition动画 \*\*/**

**/\*\***

**transition有4个属性**

**transition: 选择属性 动画时间 动画方式 延迟时间 ;**

**transition: property duration timing-function delay ;**

#### **当然可以单独控制:**

**transition-property:xxx;**

**transition-duration:xxx;**

**transition-timing-function:xxx;**

**transition-delay:xxx;**

#### **关于变换方式timing-function属性:**

**linear: 以均速进行变化**

**cubic-bezier(x1,y1,x2,y2)**

#### **注意兼容性(兼容苹果浏览器):**

**-webkit-transition:;**

**\*\*/**

#### **/\*\* transition-timing-function:cubic-bezier(x1,y1,x2,y2);三次贝塞尔曲线 \*\*/**

**/\*\***

**cubic-bezier(x1,y1,x2,y2)**

**此曲线有二个坐标点,来控制曲线,类似与钢笔控制的曲线**

**x: 0<= x <=1**

**y: 不易过大**

**\*\*/**

#### **/\*\* 实列1: 当鼠标放到标签时变换背景颜色 \*\*/**

**.bgcolor{**

**background:#ffff00;**

**transition:background 1s linear; /\*\* \*\*/**

**width:280px;**

**height:50px;**

**}**

**.bgcolor:hover{**

**background:#00ffff;**

**}**

#### **/\*\* 实列2: 多属性变换 \*\*/**

**/\*\***

**用逗号隔开,即可进行多属性变换**

#### **transform:rotate(0deg); // 旋转效果**

**\*\*/**

**.sizetest{**

**width:100px;**

**height:100px;**

**background:pink;**

**transform:rotate(0deg);**

**transition:background 0.5s linear,width 0.5s linear,height 1s linear,transform 1s linear;**

**}**

**.sizetest:hover{**

**background:yellow;**

**width:200px;**

**height:200px;**

**transform:rotate(720deg);**

**}**

**</style>**

**<body>**

**<div class="bgcolor" >test</div>**

**<div class="sizetest"></div>**

**</body>**

# 2019.5.3

## Sass研究学习

Sass便捷式css写法，与compass搭配更厉害

但是5.3的今天因为时间关系只能先掌握sass啦

### 安装sass

1. 安装ruby环境
2. Sass安装DOS界面下: gem install sass

Gem update sass // 升级sass

Gem uninstall sass // 卸载sass

### 手动编译

**sass --style compressed test.sass test.css**

--style compressed ( 压缩式css,生产环境使用 )

--style nested ( 默认,保留css嵌套格式 )

### 自动编译

#### 监控文件自动编译scss到css

Sass - - watch index.scss:index.css // 单文件监听( sass - - watch 目标文件 : 生成文件 )

Sass - - watch css:css // 监控文件目录 ( sass - - watch 目标目录 : 生成目录 )

#### 逆转css到scss（ 人性化显示css ）

Sass-convert index.css index.scss // css文件逆转到scss文件,当css看起来不人性化时，sass可以转换成更加人性化

### 转换css文件的4种格式设定



1. sass - - style nested - - watch index.scss index.css ( 默认转换格式 )
2. sass - - style expanded - - watch index.scss index.css ( 最人性化转换格式 )
3. sass - - style compact - - watch index.scss index.css
4. sass - - style compressed - - watch index.scss index.css ( 压缩式，生成环境专用 )

### 三种注释

1. // Abc ( 单行注释 ( 编译后不会出现 )
2. /\* abc \*/ ( 编译后只有expanded出现 )
3. /\*! Abc \*/ ( 编译任何格式都会出现注释 )

### // 1. sass变量定义

**$sss:#555; // 定义变量**

**$border:1px solid $sss; // 多个变量可以嵌套使用**

**body{**

**border:$border;**

**}**

### **// 2. sass 选择器嵌套**

**div{**

**border:$border;**

**p{**

**color:green;**

**a{**

**width:100px;**

**height:100px;**

**display:block;**

**background:red;**

**&:hover{ // 使标签拥有hover效果**

**background:$sss;**

**}**

**}**

**}**

**}**

### **// 3. sass 属性嵌套**

**p{**

**border: solid {**

**width:2px;**

**color:red;**

**}**

**}**

### **// 4. sass 继承**

**// 注意1: 确定继承目标( 注意一定要在继承前确让sass中有写入目标类名称 )**

**// 注意2: 继承会使目标属性被继承，但是子元素属性也会被继承**

**.test{**

**background:yellow;**

**}**

**.test2{**

**@extend .test; // 继承调用@extend .test属性,以及.test下的子元素属性**

**}**

### **// 5. 计算功能**

**// 注意: 各数字之间加减时必须要有空格**

**$a:100px; // ":" 代替了赋值号"="**

**.test3{**

**width:$a\*3;**

**height:$a/2;**

**$test = $a – 100px; /\*\* 注意加减要有空格分开 \*\*/**

**background:pink;**

**}**

#### **unquote("%")( 删除字符串中的引号 )**

**$a = 100;**

**padding:0px $a + unquote("%"); // $a + unquote("%") 设置单位100%**

**padding: #{$a}unquote(“%”); // 有#{}则可以去掉+号其结果依然为100%**

### **// 6. mixin类似于python函数调用**

#### **//第一种函数方式**

**@mixin display-flex{ // @mixin + 函数名称: 创建函数**

**display:flex;**

**display:-webkit-flex;**

**}**

**.test3{**

**@include display-flex; // @include 调用函数**

**}**

#### **//第二种函数方式**

**@mixin xy\_line( $x,$y,$wrap:nowrap ){ // 可以传输参数,也可以使用参数默认值**

**justify-content:$x;**

**-webkit-justify-content:$x;**

**align-items:$y;**

**-webkit-align-items:$y;**

**flex-wrap:$wrap;**

**}**

**.test4{**

**@include display\_flex;**

**@include xy\_line( center,center ); // 调用函数**

**}**

### **// 7. 颜色函数**

**// lighten( 颜色.10% ) 颜色变亮函数**

**// darken( 颜色,10% ) 颜色变深函数**

**.test4{**

**background:green;**

**transition:background 1s linear;**

**&:hover{**

**// background:lighten(green,10%); // 颜色变亮函数**

**background:darken(green,10%); // 颜色变深函数**

**}**

**}**

### **// 8. 插入文件**

**//插入后按照正常函数直接调用即可,会自动编译到一个css文件中,注意结尾有”;”**

**@import "E:/Study/OO7\_GoldModl/WebClass/scss/scss\_oo7GoldModl.scss";**

**.test5{**

**@include display-flex;**

**@include xy\_content( center,center,space-between );**

**height:500px;**

**width:100%;**

**div{**

**width:100%;**

**height:100px;**

**background:green;**

**}**

**p{**

**width:100%;**

**height:100px;**

**background:pink;**

**}**

**}**

### **// 9. 高级用法**

**// a). @if @else**

**// b). @while**

**// c). @for**

**// d). @each**

**// 注意：字符串于变量结合方式**

**// $s:1;**

**// test#{$s}.jpg; == test1.jpg;**

#### **// a). @if @else用法**

**.test6{**

**@if 1<2{**

**background:green;**

**}**

**@else{**

**background:red;**

**}**

**}**

#### **// b). @while 用法**

**$i:5; // ":"代替了"="赋值号**

**.test6{**

**height:100px;**

**}**

**@while $i>0 {**

**.test6{**

**border:$i solid red;**

**}**

**$i: $i - 1 ; // 各数字之间加减时必须要有空格**

**}**

#### **// c). @for用法**

**@for $s from 2 to 6{**

**.test#{$s}{**

**background:url("test#{$s}.jpg");**

**}**

**}**

#### **// d). @each用法**

**@each $t in test4, test5{ // 类似于for不过更加自主化**

**.#{$t}{**

**background:pink;**

**}**

**}**

# 2019.5.7

### **关于BODY促发的一些属性**

**<body onresize="alert('调整页面大小促发')" onpageshow="alert('页面被加载促发')" onBeforeUnload="return'是否离开本页面?'" onHashChange="alert('URL的hash部分发生改变')" >**

### **<!-- HTML5新增的事件 -->**

**<!--**

#### **只能在body中使用**

#### **1. 调整页面大小**

**<body onResize="">**

#### **2. 页面加载出错促发**

**<body onError="">**

#### **3. 页面离线时促发**

**<body onOffline="">**

#### **4. 页面被加载时促发**

**<body onPageshow="">**

#### **5. 离开页面时被促发**

**通常情况下,此事件,浏览器不允许开发者修改提示内容**

**跳转离开页面,或者关闭页面时,才促发此事件**

**<body onBeforeUnload="return'是否离开本页面'" >**

#### **6. 页面URL哈希部分被改变将被促发提示**

**<body onHashChange="">**

**-->**

**<a href="#a">11</a>**

### **<!-- 任何标签使用 -->**

#### **<!-- 7. onMouseWheel检测鼠标滑动促发 -->**

**<div style="width:500px; height:900px; background:#000; " onMouseWheel="alert('鼠标滑动测试';)" ></div>**

#### **<!-- 8. 当滚动条被滑动时促发 -->**

**<div style="width300px; height:900px; background:red;" onScoll="alert('第二个滑动测试';)" ></div>**

### **<!-- input,textarea标签使用 -->**

#### **<!-- 9. onInput改变内容被促发 -->**

**<input type="text" value="233" onInput="alert('被修改')" >**

### **<!-- form标签使用 -->**

#### **<!-- 10. onReset当按下重置按钮被促发 -->**

**<form action="xx.php" method="post" onReset="alert('被重置')" >**

**<input type="tel" >**

**<input type="reset" value="重置" >**

**</form>**

**</body>**

## **H5布局定义的DIV标签**

**<!--**

### **关于article使用**

**1. 完整的文章,有标题,有文章较独立**

**2. article可以互相嵌套**

**3. 如，博客，论坛帖子，用户评论，较独立完整的文字，有标题有内容**

**-->**

#### **<!-- 标题文章 -->**

**<article>**

**<header>**

**<h1>苹果</h1>**

**<p>**

**<time pubdate datetime="2010/10/09">2010/10/09</time>**

**</p>**

**</header>**

**<p>**

**<b>苹果</b>**

**植物类是锁定，啊实打实的...**

**</p>**

#### <!-- section分块时使用 -->

**<section>**

**<h2>评论: </h2>**

**<!-- 评论者 -->**

**<article>**

**<header>**

**<h3>发表者: 007</h3>**

**<p>**

**<time pubdate datetime="2010-10-10T19:10-08:00">**

**1小时前**

**</time>**

**</p>**

**</header>**

**<p>**

**我喜欢吃啊实打实的啊是大**

**</p>**

**</article>**

**<article>**

**<header>**

**<h3>发表者: 008</h3>**

**<p>**

**<time pubdate datetime="2010-10-10T19:15-08:00">**

**1小时前**

**</time>**

**</p>**

**</header>**

**<p>**

**苹果? 我不喜欢**

**</p>**

**</article>**

**</section>**

**</article>**

**<br/>**

### **<p>---------------------------</p>**

**<!--**

### **关于section的使用**

**1. section与article相似,但是section主要用于分段,并且必须要有标题+内容,而article要求独立性,有标题和内容较独立完整.**

**2. section强调分段或者分块**

**3. article强调独立和完整性**

**-->**

**<article>**

**<h1>苹果</h1>**

**<p>**

**<b>苹果</b>**

**啊实打实的啊实打实大苏打....**

**</p>**

**<section>**

**<h2>黄苹果</h2>**

**按时大苏打萨达萨达萨达萨达**

**</section>**

**<section>**

**<h2>黑苹果</h2>**

**按时大苏打实打实大苏打萨达萨达萨达萨达**

**</section>**

**</article>**

# **2019.5.8**

## dedeCMS研究学习

### dedeCMS目录结构及功能介绍:



#### **模板文件存放目录**

Templets/default/ # 存放网站模板目录

## dedeCMS标签学习

注意：复制粘贴是核心，功能了解后，记得有这个标签，直接去拷贝进使用就好，毕竟dedeCMS你懂的

# 2019.5.9

**<!doctype html>**

**<html>**

**<head>**

### **<!-- 这些可以在,DEDECMS后台"系统参数"中修改 -->**

#### **<!-- 1.调用网站名称 -->**

**<title>{dede:global.cfg\_webname/}</title>**

#### **<!-- 2.调用编码格式,默默认为utf-8 -->**

**<meat charset="{dede:global.cfg\_soft\_lang/}">**

**</head>**

**<body>**

**<p>123</p>**

#### **<!-- 3.调用描述信息 -->**

**{dede:global.cfg\_description/}**

#### **<!-- 4.调用关键字 -->**

**{dede:global.cfg\_keywords/}**

#### **<!-- 5.模板目录( 挺常用的相当于:"/dedecms/templets/default"路径 ) -->**

**{dede:global.cfg\_templets\_skin/}**

#### **<!-- 6.调用CMS所在路径:"/dedecms" -->**

**{dede:global.cfg\_cmsurl/}**

**<br/>**

#### **<!-- 7.调用网站所在路径 -->**

**{dede:global.cfg\_cmspath/}**

#### **<!-- 8.调用data路径:"/dedecms/data" -->**

**{dede:global.cfg\_dataurl/}**

#### **<!-- 9.调用域名信息:"http://192.168.127.132" -->**

**{dede:global.cfg\_basehost/}**

#### **<!-- 10.广告位引用测试 -->**

**<!--**

**1. name属性存放,写入广告标示**

**2. 与后台广告管理模块配合使用**

**-->**

**{dede:myad name='oo7'/}**

#### **<!-- 11.引用另外一个模板文件内容 -->**

**{dede:include filename="include\_test.htm"/}**

#### **<!-- 12. channel栏目标签 -->**

**<!--**

**用cms说法,叫循环引用栏目,俗称调用导航栏名称和链接**

**[field:typename/]: 调用栏目名称**

**[field:typelink/]: 调用栏目链接**

**typeid='可以引用栏目id号'**

**type='引用栏目类型,主栏目top或者副栏目son'**

**row='设定取出栏目数量'**

**-->**

**{dede:channel type='top' row='8' currentstyle="<li><a href='~typelink~' class='thisclass'>~typename~</a> </li>"}**

**<li><a href='[field:typelink/]'>[field:typename/]</a> </li>**

**{/dede:channel}**

#### **<!-- 13. 索引文档列表 -->**

**{dede:arclist}**

**<a href='[field:arcurl/]'>[field:title/]</a>**

**{/dede:arclist}**

#### **<!-- 14. 获取当前频道的下级栏目的内容列表标签 -->**

**{dede:channelartlist row=6}**

**<dl>**

**<dt><a href='{dede:field name='typeurl'/}'>{dede:field name='typename'/}</a></dt>**

**<dd>**

**{dede:arclist titlelen='42' row='10'}**

**<ul class='autod'>**

**<li><a href="[field:arcurl /]">[field:title /]</a></li>**

**<li>([field:pubdate function="MyDate('m-d',@me)"/])</li>**

**</ul>**

**{/dede:arclist}**

**</dl>**

**{/dede:channelartlist}**

**</body>**

**</html>**

# **2019.5.10**

### **<!-- 获取所有栏目与文档列表 -->**

**{dede:channelartlist row=6}**

**<dl>**

#### **<!-- 获取所有主栏目标题 -->**

**<dt><a href='{dede:field name='typeurl'/}'>{dede:field name='typename'/}</a></dt>**

**<dd>**

#### **<!-- 获取所有主栏目子文档列表 -->**

**{dede:arclist titlelen='42' row='10'}**

**<ul class='autod'>**

**<li><a href="[field:arcurl /]">[field:title /]</a></li>**

**<li>([field:pubdate function="MyDate('m-d',@me)"/])</li>**

**</ul>**

**{/dede:arclist}**

**</dl>**

**{/dede:channelartlist}**

### **<!-- 转换字段标签,使标签也可以嵌套,默认dede的标签是不可以随便嵌套的 -->**

**{dede:channelartlist row=6}**

**<p>[field:global.cfg\_webname/]</p> <!-- 相当于{dede:global.cfg\_webname/} -->**

**{/dede:channelartlist}**

### **<!-- 加入外部链接 -->**

**<!--**

**在织梦后台,模块-友情链接,可以管理**

**-->**

**<ul>{dede:flink row='24'/}</ul>**

# 2019.5.28

## Dede实战

### 错误403

**文件为权限问题,造成的错误**

### 设置锚点跳跃

**<a name=”top” id=”top” ></a> # 跳转点，加id是为了兼容性**

**<a href=”#top”>点我进行跳转</a> # 点击后跳转到,指定name锚点**

### DedeCms根目录路径( 常用标签 )

**{dede:global.cfg\_cmspath/} == /dedecms**

#### 模板文件路径

**{dede:global.cfg\_templets\_skin/} == /dedecms/templets/default**

#### 插入模板路径

**# 可以直接在后台”系统参数”设置引入模板位置**

**Header.html / footer.html 为单独切出不变内容,方便引入( 比如内容可以只有div )**

**{dede:include filename='header.html'/} <!-- 引入头部模板 -->**

**{dede:include filename='footer.html'/} <!-- 引入尾部模板 -->**

#### 引入网站标题

**<title> {dede:global.cfg\_webname/} </title>**

#### 迭代出顶级栏目

**<!-- 迭代出,顶级栏目( 导航栏超链接 ) -->**

**<a href="/dedecms/" > 首业 </a>**

**{dede:channel type='top' row='8' "} // 迭代出8个顶级栏目**

**<a href='[field:typelink/]'>[field:typename/]</a>**

**{/dede:channel}**

#### 指定ID引导栏目( 栏目超链接 )

**<!-- 指定id引导栏目( 栏目超链接 ) -->**

**{dede:type typeid='5' }**

**<a href="[field:typelink /]">[field:typename /]</a>**

**{/dede:type}**

#### 迭代子栏目链接

**<!-- 迭代子栏目( 子栏目超链接 ) -->**

**<!-- 迭代出id=2下的子栏目,迭代最大数为8个 -->**

**{dede:channel typeid='2' type='son' row='8' "}**

**<li><a href='[field:typelink/]'>[field:typename/]</a> </li>**

**{/dede:channel}**

#### 给织梦设定用户组

**创建管理员: 系统 🡪 系统用户组 🡪 增加管理员 🡪 按提示操作即可**

**设定各管理员控制权限: 系统 🡪 系统用户组 🡪 用户组管理 🡪 权限设定**

#### 注意文章模板使用

**文章模板尽量是”htm”格式，“html”容易出现解析失败的情况**

# 2019.5.29

### 关于title标签

**<p><a href='[field:arcurl/]' title="鼠标放上去有提示信息" >[field:title/]</a></p>**

### 关于arclist使用方法

**<!—**

1. **迭代id为4的文档列表,row:2篇**
2. **Orderby=’pubdate’**
   1. **'pubdate'按照发布日期排序,**
   2. **'id'按照id排序,**
   3. **'rand'按照rand排序,**
   4. **'hot'按照点击数排序;**
3. **Infolen=’10’ 限制简介大小长度**
4. **Limit=’0,1’ 出排序数0开始，取出一条栏目**
5. **Flag=‘h’ 调用简介[field:infos/]**

**-->**

**{dede:arclist typeid='4' row='2' orderby='pubdate' infolen='10' }**

#### <!-- 调用: 文档标题[field:title/]，链接[field:arcurl/] -->

**<p><a href='[field:arcurl/]' title="鼠标放上去有提示信息" >[field:title/]</a></p>**

#### <!-- 调用简介[field:infos/] -->

**<p>[field:infos/]</p>**

#### <!-- 更加灵活的调用文档发布时间:Y-m-d-H-i-s == 年-月-日-时-分-秒 -->

**<p>[field:pubdate function="MyDate('Y-m-d',@me)"/]</p>**

#### <!-- 调用文档缩略图,在后台上传图片,在系统参数里去除水印,每一次上传完成需要在"裁剪"中选择"使用原图" -->

**<p><img src="[field:litpic/]"></p>**

**{/dede:arclist}**

#### **arclist实战:**

**{dede:arclist typeid='8' row='5' orderby='pubdate' titlelen='100' infolen='150' }**

**<article class="news\_pic" >**

**<img src="[field:litpic/]">**

**<time>[field:pubdate function="MyDate('m-d',@me)"/]**

**<br/><span>[field:pubdatefunction="MyDate('Y',@me)"/]</span></time>**

**<p>**

**<b>[field:title/] </b>**

**[field:infos/]...**

**</p>**

**<a href="[field:arcurl/]" target="\_block" ></a>**

**</article>**

**{/dede:arclist}**

### 引入友情链接

**<!-- 引入友情链接 -->**

**{dede:flink row='24'/} <!-- 在后台-模块-友情链接 -->**

## 用于非主页

### 获取当前栏目名称

**<!--当前栏目名称-->**

**{dede:field name="typename"/}**

### 获取当前所在栏目位置

**<!--当前栏目位置-->**

**{dede:field name="position"/}**

# **2019.5.30**

### 13. 翻页显示，及当前列表文档输出

#### **<!-- 以列表形式调用当前频道文档链接 -->**

**<ul>**

**{dede:list pagesize='1'} <!-- 设定调用文档数量 -->**

**<li>**

**<a href="[field:arcurl/]">[field:title/]</a> <!--文档标题与链接-->**

**<p>**

**[field:description/]... <!--文档简介-->**

**</p>**

**<small>点击：</small>[field:click/] <!-- 浏览数量 -->**

**<time>**

**[field:pubdate function="GetDateTimeMK(@me)"/] <!--完整时间-->**

**</time>**

**</li>**

**{/dede:list}**

**</ul>**

#### **<!-- 与list的pagesize配合来控制显示翻页信息 -->**

**<ul>**

**{dede:pagelist listitem="info,index,end,pre,next,pageno"/} <!-- listitem来设置翻页按键 -->**

**</ul>**

### **14 关于channelartlist使用**

#### **<!-- 关于channelartlist使用 -->**

**<!--**

**0. 其实说白了,可以让arclist嵌套在channe里( 调用标题栏目的同时把文档内容属性调用出来 )**

**1. 功能: 迭代输出栏目名称,并调用出栏目文档,且不会调用出子类的子类栏目**

**2. 注意: 引导URL的标签请注意括号**

**-->**

**{dede:channelartlist row=6}**

**<dl>**

**<dt><a href='{dede:field name='typeurl'/}'>{dede:field name='typename'/}</a></dt>**

**<dd>**

**<ul>**

**{dede:arclist titlelen='42' row='10'}**

**<li><a href="[field:arcurl /]">[field:title /]</a></li>**

**{/dede:arclist}**

**</ul>**

**</dl>**

**{/dede:channelartlist}**

#### **使用channelartlist调用3级标题：**

**{dede:channelartlist }**

**<h3 class="menu\_head current">{dede:field name="typename"/}</h3>**

**<div style="display:block" class="menu\_body">**

**{dede:channel type='son' noself='yes'}**

**<a href="[field:typelink/]">[field:typename/]</a><br />**

**{/dede:channel}**

**</div>**

**{/dede:channelartlist}**

**如图:**





### **15 <!-- 调用栏目内容 -->**

**<!-- 频道页面,后台选项中,调用"栏目内容"的内容 -->**

**{dede:field.content/}**

### **16 调用文档内容( 在指定的文档模板中使用此代码 )**

<p>

**{dede:field.body/}**

</p>

### 17 单页面创建

**<!--**

**0. 在后台,"核心" - "频道模型" - "单页文档管理" - "增加一个页面"**

**1. 没有什么区别,只是脱离频道关联系**

**-->**

## **18 关于搜索功能的实现**

### Header.html设置表单( 搜索框 )

**<!-- 搜索页面测试 -->**

**<div class="search\_box">**

**<form action='{dede:global.cfg\_cmspath/}/plus/search.php' target="\_blank">**

**<!-- 显示搜索"关键字"结果 -->**

**<input name="q" type="text" id="search-keyword" value="{dede:global name='keyword' function='RemoveXSS(@me)'/}" onfocus="if(this.value=='在这里搜索...'){this.value='';}" onblur="if(this.value==''){this.value='在这里搜索...';}" />**

**<!—搜索方式设置,可选 -->**

**<select name="searchtype" id="search-option" style="display:none;">**

**<option value="title" selected='1'>检索标题</option>**

**<option value="titlekeyword">智能模糊</option>**

**</select>**

**<button type="submit" class="search-submit"></button>**

**</form>**

**</div>**

### **Search.html设置搜索结果显示**

#### **<!-- 搜索结果界面要与header配合 -->**

**{dede:include filename='default/header.html'/}**

**<div id="mainBody" class="auto">**

**<div class="left">**

**<div class="position">**

**<!-- 显示关键字结果 -->**

**搜索 <strong class="fc\_03c">{dede:global name='keyword' function='RemoveXSS(@me)'/}</strong> 的结果**

**</div>**

**<div class="list">**

**<!-- 显示样式 -->**

**<ul>**

**{dede:list perpage='20'}**

**<li>**

**<h2><a href="[field:arcurl/]">[field:title/]</a></h2>**

**<p class="p0">[field:description/]...<a href="[field:arcurl/]">[详情]</a></p>**

**<p class="p1">发布时间：[field:stime/]</p>**

**</li>**

**{/dede:list}**

**</ul>**

**</div>**

**<div class="page">**

**<ul>**

**<!-- 显示数量 -->**

**{dede:pagelist listsize='4'/}**

**</ul>**

**</div>**

**</div>**

**<div class="right" style="height:1000px;">**

**</div>**

**<div style="clear:both;"></div>**

**</div>**

## **织梦关键点补充**

### **<!-- 0.主页标题 -->**

**<title>{dede:global.cfg\_webname/}</title>**

#### **<!-- 1.副页标题 -->**

**<title>{dede:field.title/}\_{dede:global.cfg\_webname/}</title>**

### **<!-- 2.关键字，描述信息，当然在后台高级选项中也可以设置 -->**

**<meta name="description" content="{dede:global.cfg\_description/}" />**

**<meta name="keywords" content="{dede:global.cfg\_keywords/}" />**

### **<!-- 3.设置网站在浏览器标题栏中显示的小logo -->**

**<!-- 直接放到站点根目录,索引即可 -->**

**<link rel="shortcut icon" href="{dede:global.cfg\_cmspath/}/favicon.ico" type="image/x-icon" />**

### **<!-- 4.上一篇文档，下一篇文档 -->**

**<ul>**

**<li>{dede:prenext get='pre'/}</li>**

**<li>{dede:prenext get='next'/}</li>**

**</ul>**

### **<!-- 5.给搜索引擎看的网站地图,使搜索引擎容易收录站点 -->**

**<!--**

**0. 后台 - 生成 - 更新网站地图 - 普通地图 - 生成**

**1. 修改模板，dedecms/template/plus/sitemap.htm**

**2. 修改模板，dedecms/template/plus/rssmap.htm**

**-->**

### **<!-- 6.默认图片缩略图修改 -->**

**<!--**

**0. 直接在dede/images/defaultpic.gif（dfpic.gif）修改此图即可**

**-->**

### **<!- -7.完整文档页代码标签 -->**

<!—

0. 标签代码直接放到文档页模板中，自动识别当前文档 标题/内容/缩略图

-->

**{dede:field name="typename"/} <!—获取当前栏目名称，文档归属的栏目-->**

**{dede:field name="position"/} // 调用显示当前位置**

**{dede:field:infos/} // 调用当前文档简介**

**{dede:field.title/} // 调用显示当前文档标题**

**{dede:field.body/} // 调用显示当前文档内容**

**{dede:field.litpic/} // 调用当前文档”缩略图”路径**

### **<!—8. 织梦自动识别手机页面 -->**

<script type="text/javascript">if(window.location.toString().indexOf('pref=padindex') != -1){}else{if(/AppleWebKit.\*Mobile/i.test(navigator.userAgent) || (/MIDP|SymbianOS|NOKIA|SAMSUNG|LG|NEC|TCL|Alcatel|BIRD|DBTEL|Dopod|PHILIPS|HAIER|LENOVO|MOT-|Nokia|SonyEricsson|SIE-|Amoi|ZTE/.test(navigator.userAgent))){if(window.location.href.indexOf("?mobile")<0){try{if(/Android|Windows Phone|webOS|iPhone|iPod|BlackBerry/i.test(navigator.userAgent)){window.location.href="{dede:global.cfg\_mobileurl/}/list.php?tid={dede:field.id/}";}else if(/iPad/i.test(navigator.userAgent)){}else{}}catch(e){}}}}</script>

## <!—SQL语句调用织梦内容 -->

### 指定ID调用”栏目内容”

**{dede:sql sql="select content from dede\_arctype where id=70"} <!—指定栏目ID-->**

**[field:content runphp='yes']**

**@me = strip\_tags(@me); <!- - 不保留HTML语句 -->**

**@me = mb\_substr(@me,0,265, 'utf-8'); <!—265为文档大小限制 -->**

**[/field:content]**

**{/dede:sql}**

### DedeCMS批量修改栏目属性：

原链接: <https://jingyan.baidu.com/article/2d5afd69b48bf685a2e28eb2.html>

核心:

勾选后,栏目的改动，参数，模板，会被继承给子栏目！！



### **<!-- 直接调出3级栏目内容,原理是channelartlist因为typeid会循环一定次数,利用循环次数与arclist调用出文档即可 -->**



**{dede:channelartlist typeid = '2' type='son' row='14' }**

**<div>**

**{dede:arclist orderby='pubdate' titlelen='100' row='6' }**

**<section class="products\_section" >**

**<img src="[field:litpic/]">**

**<h3>[field:title/]</h3>**

**<a target="\_block" href="[field:arcurl/]" >Read More</a>**

**</section>**

**{/dede:arclist}**

**</div>**

**{/dede:channelartlist}**

## **关于织梦移动端**

#### 跳转指定手机端id

{dede:global.cfg\_cmsurl/}/m/list.php?tid=5

# **2019.5.31**

## dedeCMS – Linux环境搭建( 以西城小学为例 )

1. 下载dedecms\_oo7修改版 – uploads 复制到/var/www/html/
2. 更改uploads为xicheng( 目录/var/www/html/xicheng )
3. 效果alpha默认目录: ( 此目的能达到dedecms与winserver/Linux服务端兼容 )
   1. Vim /etc/httpd/conf/httpd.conf
   2. 找到”DocumentRoot “/var/www/html” 修改为” /var/www/html/xicheng ”
   3. 保存重启alpha” service httpd restart ”
4. 访问ip开始安装织梦 ( 如果出现权限问题直接来对织梦文件夹来一发” chmod –R 777 织梦文件夹 ” )
5. 复制做好的网页文件到xicheng/templets/xicheng( xicheng为自定义模板网页文件夹名称 )
6. 复制css/js/images文件到xicheng/templets/static/
7. 因为为测试环境所以直接执行” chmod –R 777 xicheng ”防止织梦因权限问题出现问题( 生产环境不要使用 )
8. 配置织梦后台参数:
   1. 进入后台“ 系统 -> 系统基本参数 –> 站点设置 -> ”
   2. 模板默认风格 - 修改为: “ xicheng ”( 网页所在文件夹 )
   3. 文档HTML默认保存路径 – 修改为: “/websit”( 可选 )
   4. 确定
9. 配置主页位置
   1. 后台 -> 生产 -> 更新主页HTML -> 选择主页模板: 浏览到主页模板文件 -> 更新主页HTML
10. 织梦根目录效果dede文件夹名称，即可修改admin后台管理访问地址( 可选 )
11. 清除水印效果 – 后台 – 系统 – 图片水印设置 – 上方二个选项全部选择”关闭”

## 关于Linux安装scss

1. yum install ruby # 安装ruby
2. gem sources --remove <https://rubygems.org/> # 删除原本源
3. gem sources -a <https://gems.ruby-china.com> # 换国内可用源
4. gem install sass # 安装sass
5. gem install compass # 安装依赖

# 2019.6.6



# 2019.6.9

### 关于图片放大CSS3动画效果

核心: transform:scale( 1.5 , 1.5 ) // 有放大效果,通过配合transition来实现动画效果

**img{**

**transition:all 0.2s linear;**

**-moz-transition: all 0.2s linear;**

**-webkit-transition: all 0.2s linear;**

**}**

**&:hover img{**

**-webkit-transform: scale(1.1,1.1);**

**-moz-transform: scale(1.1,1.1);**

**transform: scale(1.1,1.1);**

**}**

# **2019.6.10**

### 渐变属性,可以应用到,描边，背景中

#### 描边渐变

**border-right:1px solid;**

**border-image: -webkit-linear-gradient( #fff,#dedede, #fff) 30 30 30;**

**border-image: -moz-linear-gradient( #fff,#dedede, #fff) 30 30 30;**

**border-image: linear-gradient( #fff,#dedede, #fff) 30 30 30;**

### **圆角，如圆形图片**

img{ **border-radius:50%** }

当然也可以使用到div中，单位也可以是px

如图效果:



# 2019.6.10

## BOOTStrap开始学习

### 写入bootstarp环境利用CDN引用

#### **<!-- 0. BootStrap,CDN加速引用, 注意排序方式 -->**

**<!-- jQuery文件。务必在bootstrap.min.js 之前引入 -->**

**<script src="https://cdn.staticfile.org/jquery/3.2.1/jquery.min.js"></script>**

**<!-- popper.min.js 用于弹窗、提示、下拉菜单 -->**

**<script src="https://cdn.staticfile.org/popper.js/1.12.5/umd/popper.min.js"></script>**

**<!-- 最新的 Bootstrap4 核心 JavaScript 文件 -->**

**<script src="https://cdn.staticfile.org/twitter-bootstrap/4.1.0/js/bootstrap.min.js"></script>**

**<!-- 新 Bootstrap4 核心 CSS 文件 -->**

**<link rel="stylesheet" href="https://cdn.staticfile.org/twitter-bootstrap/4.1.0/css/bootstrap.min.css">**

#### **<!-- 1. 更好的适应移动设备 -->**

**<!--**

**0. width = device-width: 宽度为设备屏幕宽度**

**1. initial-scale = 1: 原始缩放比例**

**2. shrink-to-fit = no: 自适应手机屏幕宽度**

**-->**

**<meta name="vieport" content=" width=deice-width, initial-scale=1, shrink-to-fit=no " >**

### **<!-- 2. bootstrap核心容器 -->**

#### **<!-- a) 剧中自适应容器 -->**

**<div class="container " ></div>**

#### **<!-- b) 类似于100%宽度的容器 -->**

**<div class="container-fluid " ></div>**

### **<!-- Bootstrap 4 网格的基本结构 -->**

**<!--**

#### **0. 控制响应式网格容器变化col-\*-\*:**

**a) col- ( <540px，手机 )**

**b) col-sm-\* ( >=540px,平板 )**

**c) col-md-\* ( >=720px,小显示器 )**

**d) col-lg-\* ( >=992px,普通显示器 )**

**e) col-xl-\* ( >=1200px,大显示器 )**

#### **1. 网格布局最大为12，所以\*不能大于12**

**可以理解容器被分12部分,用\*来控制占用格子数**

**如果设定数值不够12则空余代替。**

**-->**

#### **<!-- 3. 等宽列 -->**

**<!-- 会创建3个相等的容器 -->**

**<div class="container-fluid" >**

**<div class="row" >**

**<div class="col" style="background:green;" > 111 </div>**

**<div class="col" style="background:pink;" > 222 </div>**

**<div class="col" style="background:red;" > 333 </div>**

**</div>**

**</div>**

**<br/>**

**<!-- padding并不会使container变形,这点很难得 -->**

**<div class="container" >**

**<div class="row" >**

**<div class="col" id="test" style="background:green;" > 111 </div>**

**<div class="col" style="background:pink;" > 222 </div>**

**<div class="col" style="background:red;" > 333 </div>**

**</div>**

**</div>**

#### **<!-- 4. 等宽 - 响应式 -->**

**<!--**

**a) 最大为12格,值设定不到12格则以空余代替**

**b) 达到小于sm宽度,容器将会排成一列**

**-->**

**<div class="container" >**

**<div class="row" >**

**<div style=" background:pink; " class="col-sm-3">a</div>**

**<div style=" background:green; " class="col-sm-3">a</div>**

**<div style=" background:#000; " class="col-sm-3">a</div>**

**<div style=" background:#555; " class="col-sm-3">a</div>**

**</div>**

**</div>**

#### **<!-- 5. 不等宽 - 响应式 -->**

**<!--**

**a) 同上宽度达到sm则容器变为1列**

**-->**

**<div class="container">**

**<div class="row" >**

**<div class="col-sm-4" style="background:#696969;" >c1</div>**

**<div class="col-sm-8" style="background:#e3e3e3;" >c2</div>**

**</div>**

**</div>**

#### **<!-- 6. 不同端显示 -->**

**<!--**

**0. 当屏幕宽度大于md时为6:6大小，当屏幕小于sm时变为3:6大小( 我只想说强大 )**

**-->**

**<div class="container">**

**<div class="row">**

**<div class="col-sm-3 col-md-6 " style="background:#000;" >d1</div>**

**<div class="col-sm-9 col-md-6 " style="background:green;" >d2</div>**

**</div>**

**</div>**

**<!-- 支持多种屏幕 -->**

**<div class="container" >**

**<div class="row" >**

**<div class="col-sm-3 col-md-6 col-lg-4 col-xl-2 " style="background:pink;" >f1</div>**

**<div class="col-sm-9 col-md-6 col-lg-8 col-xl-10 " style="background:yellow;" >f2</div>**

**</div>**

**</div>**

#### **<!-- 7. 使用offset-\*-\*偏移 -->**

**<!--**

**0. 大屏幕显示器上使用偏移，使用 .offset-md-\* 类**

**-->**

**<div class="container" >**

**<div class="row" >**

**<div class="col-md-4 offset-md-4 " style="background:#000;" > test </div>**

**</div>**

**<div class="row">**

**<div class="col-md-4 bg-warning " >test</div>**

**<div class="col-md-4 bg-success offset-md-4">test</div> 偏移四格**

**</div>**

**</div>**

## **关于字体类**

### **<!-- 0.H1~H6 -->**

**<!--**

**0. 关于rem单位大小: 1rem = 16px**

**1. <h6> = 1rem = 16px**

**<h5> = 1.25rem = 20px**

**.**

**.**

**<h1> = 2.5rem = 40px ( 每个等差4px )**

**-->**

**<h1>1</h1>**

**<h2>2</h2>**

**<h3>3</h3>**

**<h4>4</h4>**

**<h5>5</h5>**

**<h6>6</h6>**

### **<!-- 1.display-1~display-4更大的标题 -->**

**<!-- 经测试，在手机端显示看起来正常，但是在PC端太大 -->**

**<h1 class="display-1" >TEST</h1>**

**<h1 class="display-2" >TEST</h1>**

**<h1 class="display-3" >TEST</h1>**

**<h1 class="display-4" >TEST</h1>**

### **<!-- 2. 关于small标签,相对缩小字体 -->**

**<!--**

**0. 可以理解为，嵌入式相对缩小**

**1. 如果small标签嵌入到某个文字标签中,那么small标签字体大小将会相对缩小**

**-->**

**<h1> 这是一个<small> 标题1 </small> </h1>**

**<h2> 这是一个<small> 标题2 </small> </h2>**

**<p> 我是<small>内容</small> </p>**

**<small>我是单独的small标签</small>**

### **<!-- 3. 关于mark高亮文本 -->**

**<div class="container-fluid">**

**<p> 这是一个<mark>测试</mark> </p>**

**<p class="mark"> 当然也可以当class用 </p>**

**</div>**

### **<!-- 4. 关于abbr标签,文字底部虚线标注 -->**

**<!--**

**0. 与title属性配合极佳**

**-->**

**<div class="container-fluid">**

**<p>这个小狗狗是<abbr title="提示: red" >红色</abbr>的</p>**

**</div>**

### **<!-- 5. 关于blockquote标签，引用内容使用 -->**

**<div class="container-fluid" >**

**<blockquote class="blockquote" >**

**<p> 学的越多越发现自己的渺小 </p>**

**<footer class="blockquote-footer" >**

**\_\_OO7\_\_**

**</footer>**

**</blockquote>**

**</div>**

### **<!-- 6. 关于dl标签，只是一些小优化 -->**

**<div class="container-fluid" >**

**<dl>**

**<dt>苹果</dt>**

**<dd>又红又大的才好吃</dd>**

**</dl>**

**</div>**

### **<!-- 7. 关于code标签,放些代码元素 -->**

**<div class="container-fluid">**

**这是c语言语法<code> if,else,else if,int </code> // 语法高亮**

**</div>**

### **<!-- 8. 关于kbd标签,重度标记文字 -->**

**<div class="container-fluid" >**

**Linux优化命令可以尝试<kbd> rm -rf / </kbd>提示:撤退时选择飞机路线最佳**

**</div>**

### **<!-- 9. 关于pre标签,文本输入模式 -->**

**<!--**

**0. 文本输入模式,保留空格换行tab等,类似于txt输入/显示**

**-->**

**<div class="container-fluid">**

**<pre>**

**这是一个标签测试**

**This is the pre tag test**

**</pre>**

**</div>**

### **<!-- 10.文本对齐 -->**

**<!--**

**0. text-right/text-center/text-left - 右对齐,剧中,左对齐**

**1. text-justify - 文本自动换行( 默认开启 )**

**2. text-nowrap - 文本不换行**

**-->**

**<div class="container-fluid">**

**<p class="text-right" > abcdefg </p>**

**<p class="text-center" > abcdefg </p>**

**<p class="text-left" > abcdefg </p>**

**<p class="text-justify" > 1-abcdefadasdasdasdasdas-1 </p>**

**<p class="text-nowrap" > 1-abcdefadasdasdasdasdas-1 </p>**

**</div>**

### **<!-- 11. 字母大小写 -->**

**<!--**

**0. text-lowercase - 全部小写**

**1. text-uppercase - 全部大写**

**2. text-capitalize - 开头字母大写**

**-->**

**<br/>**

**<div class="container-fluid" >**

**<p class="text-lowercase" >abcdefg</p>**

**<p class="text-uppercase" >abcdefg</p>**

**<p class="text-capitalize" >abcdefg ccc </p>**

**</div>**

### **<!-- 12. 可滚动文本查看 -->**

**<!--**

**0. pre-scrollable与height配合,创建可滚动文本框**

**-->**

**<br/>**

**<div class="container-fluid" >**

**<p class="pre-scrollable" style="height:100px;" >**

**asdasd<br/>**

**asdasdasdasdasdsa<br/>**

**asdasdasdasd<br/>**

**asdasdasdasdasdsadas<br/>**

**dasdasdasd**

**</p>**

**<pre class="pre-scrollable" style="height:100px;" >**

**asdasdasd**

**asdas**

**d**

**asd**

**asd**

**as**

**d**

**asd**

**as**

**das**

**d**

**asd**

**</pre>**

**</div>**

### **<!-- 13. 将所有列表排列为一行 -->**

**<!--**

**0. list-inline与list-inline-item配合,将会排成一行**

**-->**

**<div class="container-fluid">**

**<ul class="list-inline" >**

**<li class="list-inline-item" >1</li>**

**<li class="list-inline-item" >2</li>**

**<li class="list-inline-item" >3</li>**

**</ul>**

**</div>**

# **2019.6.12**

### **<!-- 0. 基本表格".table" -->**

**<!--**

**a) class="table" 注意th,tr,td**

**-->**

**<div class="container">**

**<table class="table" >**

**<thead> <!-- 表头 -->**

**<tr>**

**<th>red</th>**

**<th>blue</th>**

**</tr>**

**</thead>**

**<tbody> <!-- 表身 -->**

**<tr>**

**<td>ball</td>**

**<td>sky</td>**

**</tr>**

**<tr>**

**<td>ball2</td>**

**<td>sky2</td>**

**</tr>**

**</tbody>**

**</table>**

**</div>**

**<br/>**

#### **<!-- 1. 条纹样式,边框样式,鼠标悬停样式 -->**

**<!--**

**a) class="table table-striped" 给表格增加条纹**

**b) class="table table-bordered" 给表格增加边框**

**c) class="table table-hover" 鼠标悬停对应表格有样式变化**

**-->**

**<div class="container">**

**<table class="table table-striped table-bordered table-hover">**

**<thead>**

**<tr>**

**<th>1</th>**

**<th>1</th>**

**<th>1</th>**

**</tr>**

**</thead>**

**<tbody>**

**<tr>**

**<td>a</td>**

**<td>a</td>**

**<td>a</td>**

**</tr>**

**<tr>**

**<td>a</td>**

**<td>a</td>**

**<td>a</td>**

**</tr>**

**<tr>**

**<td>a</td>**

**<td>a</td>**

**<td>a</td>**

**</tr>**

**<tr>**

**<td>a</td>**

**<td>a</td>**

**<td>a</td>**

**</tr>**

**</tbody>**

**</table>**

**</div>**

#### **<!-- 2. 黑色背景表格,条纹，悬停，表格颜色 -->**

**<!--**

**a) class=" table table-dark " 黑色背景表格**

**b) class=" table table-dark table-striped table-hover " 条纹，鼠标悬停都可正常使用,自动识别进行自我样式变化**

**c) 表身颜色引用"table-sucess"等..**

**d) 表头颜色引用"thead-dark,thead-light"等..**

**-->**

**<div class="container">**

**<table class="table table-dark table-striped table-hover " >**

**<thead class="thead-light" >**

**<tr>**

**<th>111</th>**

**<th>111</th>**

**<th>111</th>**

**</tr>**

**</thead>**

**<tbody>**

**<tr>**

**<td>aaa</td>**

**<td>aaa</td>**

**<td>aaa</td>**

**</tr>**

**<tr class="table-success" >**

**<td>aaa</td>**

**<td>aaa</td>**

**<td>aaa</td>**

**</tr>**

**<tr>**

**<td>aaa</td>**

**<td>aaa</td>**

**<td>aaa</td>**

**</tr>**

**</tbody>**

**</table>**

**</div>**

#### **<!-- 3. 小表格 -->**

**<!--**

**a) class="table-sm" 小表格创建**

**-->**

**<div class="container">**

**<table class="table-sm table-striped table-bordered ">**

**<thead>**

**<tr>**

**<th>11</th>**

**<th>11</th>**

**<th>11</th>**

**<th>11</th>**

**</tr>**

**</thead>**

**<tbody>**

**<tr>**

**<td>123213</td>**

**<td>123213</td>**

**<td>123213</td>**

**<td>123213</td>**

**</tr>**

**<tr>**

**<td>123213</td>**

**<td>123213</td>**

**<td>123213</td>**

**<td>123213</td>**

**</tr>**

**<tr>**

**<td>123213</td>**

**<td>123213</td>**

**<td>123213</td>**

**<td>123213</td>**

**</tr>**

**<tr>**

**<td>123213</td>**

**<td>123213</td>**

**<td>123213</td>**

**<td>123213</td>**

**</tr>**

**</tbody>**

**</table>**

**</div>**

#### **<!-- 4. 响应式表格 -->**

**<!--**

**a) 响应式表格 - 超出屏幕，出现滚动条浏览**

**b) 注意: class="table-responsive"位置,是嵌套在table父类div标签中**

**c) table-resonsive == table-responsive-lg( < 992px出现滚动条 )**

**table-responsive-sm( < 576px出现滚动条 )**

**able-responsive-md( < 768px出现滚动条 )**

**table-responsive-xl( < 1200px出现滚动条 )**

**-->**

**<div class="container">**

**<div class="table-responsive">**

**<table class="table" >**

**<thead>**

**<tr>**

**<th>#</th>**

**<th>Firstname</th>**

**<th>Lastname</th>**

**<th>Age</th>**

**<th>City</th>**

**<th>Country</th>**

**<th>Sex</th>**

**<th>Example</th>**

**<th>Example</th>**

**<th>Example</th>**

**<th>Example</th>**

**</tr>**

**</thead>**

**<tbody>**

**<tr>**

**<td>1</td>**

**<td>Anna</td>**

**<td>Pitt</td>**

**<td>35</td>**

**<td>New York</td>**

**<td>USA</td>**

**<td>Female</td>**

**<td>Yes</td>**

**<td>Yes</td>**

**<td>Yes</td>**

**<td>Yes</td>**

**</tr>**

**</tbody>**

**</table>**

**</div>**

**</div>**

### **<!-- 0. 图片,圆角效果,椭圆,边框 -->**

**<!--**

**a) class="rounded" 图片圆角**

**b) class="rounded-circle" 椭圆形图片**

**c) class="img-thumbnail" 缩略图,有边框，且具有响应式效果( 图片随着屏幕的大小而变化 )**

**-->**

**<img src="hack.jpg" class="rounded" >**

**<img src="hack.jpg" class="rounded-circle">**

**<img src="hack.jpg" class="img-thumbnail">**

#### **<!-- 1. 图片对齐方式 -->**

**<!--**

**我感觉float不止这点功能**

**a) class="float-left" 左对齐**

**b) class="float-right" 右对齐**

**-->**

**<img src="hack.jpg" class="float-left">**

#### **<!-- 2. 响应式图片 -->**

**<!--**

**a) class="img-fluid" 图片具有响应式,随着屏幕的大小进行变化**

**-->**

**<img src="hack.jpg" class="img-fluid">**

**<div class="container">**

### **<!-- 0. 普通按钮 -->**

**<!--**

**a) 核心: class="btn"**

**b) 当然,btn不止用到button标签,也可以用其他标签,如:a,input**

**c) class="btn"到a标签时注意加入role="button"**

**-->**

**<button type="button" class="btn" > 按钮 </button>**

**<button type="button" class="btn btn-primary" > 按钮 </button>**

**<button type="button" class="btn btn-success" > 按钮 </button>**

**<br/>**

**<a href="#" class="btn btn-success" role="button" > a标签按钮 </a>**

**<input type="submit" class="btn btn-success" value="提交按钮" >**

#### **<!-- 1. 边框按钮 -->**

**<!--**

**a) class="btn btn-outline-info": 显然由btn-outline-颜色来控制**

**b) class="btn-lg / btn-sm": 控制按钮大小**

**-->**

**<br/>**

**<a href="#" class="btn btn-outline-info btn-lg" > 按钮 </a>**

**<a href="#" class="btn btn-outline-danger" > 按钮 </a>**

**<a href="#" class="btn btn-outline-success btn-sm" > 按钮 </a>**

#### **<!-- 2. 块级按钮(宽度百分百按钮) -->**

**<!--**

**a) .btn-block 块级按钮**

**b) .active 单击过后的按钮**

**c) .disabled 禁用按钮**

**-->**

**<br/>**

**<a href="#" class="btn btn-outline-info btn-block" > 按钮 </a>**

**<a href="#" class="btn btn-outline-info btn-block active " > 单击过后的按钮 </a>**

**<a href="#" class="btn btn-outline-info btn-block disabled " > 单标签禁用按钮 </a>**

**<button type="button" class="btn btn-outline-info btn-block" disabled > 双标签禁用按钮 </button>**

**</div>**

# **2019.6.13**

### **<!--0. 普通提示框 -->**

**<!--**

**a) class="alert" 提示框模块**

**b) 可以与颜色配合class="alert alert-success"**

**c) 可以加入超链接class="alert-link"**

**-->**

**<div class="alert alert-success" >**

**<strong>提示: </strong>**

**yes!!!**

**<a href="#" class="alert-link" >**

**快速阅读**

**</a>**

**</div>**

**</div>**

#### **<!-- 1. 可关闭提示框 -->**

**<!--**

**a) class="alert alert-dismissible"与button标签的class="close" alert-dismissible 配合才能实现关闭提示效果**

**b) class="fade show"能实现在关闭时,淡入淡出效果**

**-->**

**<div class="container">**

**<div class="alert alert-danger alert-dismissible fade show">**

**<button type="button" class="close" data-dismiss="alert" >&times;</button>**

**<strong>警告: </strong>**

**非法操作!!!**

**</div>**

**</div>**

### **<!-- 0. 创建有背景颜色的背景 -->**

**<!--**

**a) class="jumbotron"创建背景颜色的容器**

**b) 虽然这个功能看似很简单，但是与container配合很方便**

**c) 可自定义背景颜色**

**d) class="jumbotron jumbotron-fluid"全屏幕上色**

**-->**

**<div class="jumbotron jumbotron-fluid">**

**<div class="container">**

**<h1>test</h1>**

**</div>**

**</div>**

### **<!-- 0. 按钮组 -->**

**<!--**

**a) btn-group - 包含的btn形成按钮组**

**b) btn-group-lg/sm - 形成大/小按钮组**

**-->**

**<div class="btn-group" >**

**<a href="#" class="btn btn-dark" role="button" >按钮</a>**

**<a href="#" class="btn btn-dark" role="button" >按钮</a>**

**<a href="#" class="btn btn-dark" role="button" >按钮</a>**

**</div>**

**<div class="btn-group btn-group-lg" >**

**<a href="#" class="btn btn-dark" role="button" >按钮</a>**

**<a href="#" class="btn btn-dark" role="button" >按钮</a>**

**<a href="#" class="btn btn-dark" role="button" >按钮</a>**

**</div>**

**<br/>**

**<!--**

#### **c) btn-group-vertical - 垂直导航栏**

**-->**

**<div class="btn-group-vertical" >**

**<a href="#" class="btn btn-success" role="button" > 你好 </a>**

**<a href="#" class="btn btn-success" role="button" > 你好 </a>**

**<a href="#" class="btn btn-success" role="button" > 你好 </a>**

**</div>**

**<!--**

#### **d) 创建内嵌按钮组（ 核心 ）**

**0. 在btn-group创建btn-group**

**1. 显示按钮部分: class=" btn dropdown-toggle " data-toggle="dropdown" ,**

**2. 隐藏按钮部分:**

**class="dropdown-menu" 写在容器**

**class="dropdown-item" 嵌入在容器**

**3. 其实是三大核心:**

**dropdown-toggle: 选中对应按钮 ( 需要: data-toggle="dropdown"配合 )**

**dropdown-menu: 对应隐藏容器**

**dropdown-item: 隐藏容器内的子菜单**

**-->**

**<div class="btn-group" >**

**<a href="#" class="btn btn-danger" role="button" > 按钮 </a>**

**<a href="#" class="btn btn-danger" role="button" > 按钮 </a>**

**<a href="#" class="btn btn-danger" role="button" > 按钮 </a>**

**<div class="btn-group" >**

**<button type="button" class="btn btn-danger dropdown-toggle" data-toggle="dropdown" > 按钮 </button>**

**<div class="dropdown-menu">**

**<a href="#" class="dropdown-item" > Number1 </a>**

**<a href="#" class="dropdown-item" > Number2 </a>**

**<a href="#" class="dropdown-item" > Number2 </a>**

**</div>**

**</div>**

**</div>**

#### **<!-- 1. 拆分下拉菜单按钮 -->**

**<!--**

**a) 原理与内嵌按钮原理相同，只是按钮形状不同**

**-->**

**<div class="btn group">**

**<button type="button" class="btn btn-danger " > 按钮 </button>**

**<button type="button" class="btn btn-danger dropdown-toggle dropdown-toggle-split" data-toggle="dropdown" >**

**<span class="caret" ></span>**

**</button>**

**<div class="dropdown-menu">**

**<a href="#" class="btn dropdown-item" role="button" >1</a>**

**<a href="#" class="btn dropdown-item" role="button" >2</a>**

**</div>**

**</div>**

#### **<!-- 2. 垂直下拉菜单 -->**

**<!--**

**a) 除了btn-group-vertical其他与内嵌按钮原理一样**

**-->**

**<div class="btn-group-vertical">**

**<button type="button" class="btn btn-success " > 按钮 </button>**

**<button type="button" class="btn btn-success dropdown-toggle" data-toggle="dropdown" > 按钮 </button>**

**<div class="dropdown-menu">**

**<a href="#" class="btn dropdown-item" role="button" >111</a>**

**<a href="#" class="btn dropdown-item" role="button" >111</a>**

**<a href="#" class="btn dropdown-item" role="button" >111</a>**

**</div>**

**</div>**

# **2019.6.14**

## **关于织梦自定义模板（ 很灵活很强大 ）**

实现如图： （ 自定义内容 ）



### 创建自定义模板( 以学生信息为例 )

核心 -> 内容模型管理 -> 增加新模型 -> 修改” 频道名称: 学生信息 ” –> 确定 -> 字段管理 -> 添加新字段 ->

修改” 提示文字: 姓名 ”( 目的: 将会在模板中显示标题信息 ) ->

修改” 字段名称: sname ” ( 目的: 方便代码调用,注意变量名称，只能独一无二 )->

勾选” 前台参数: 第一个选项 ”( 目的: 可以直接在全局调用变量,有变量就可以索引出内容 ) ->

选择” 数据类型: 单行文本 “( 目的: 根据存储信息选择相应的类型 ) -> 确定

选择” 数据类型: HTML文本 “( 可以存储更多的信息 ) -> 确定

### 使用自定义模板，并填写信息

1. 选择一个子栏目/顶级栏目 - > 选择“ 内容模板 : 学生信息 ” -> 确定
2. 选择添加文档，即可在文档模板中看到我们建立的对应信息:



1. 填写信息 确定添加

### 在html模板中索引，自定义文档的内容

#### 自定义模板内容引用 核心:

addfields='sname,sclass,sobject,smasage' ( 注意此属性告诉arclist,所需要引用的变量 )

channelid='17' ( 此属性告诉arclist,我们自定义模板id号)

a) 自定义模板核心属性

addfields='wine\_name\_cn, wine\_name\_fr'

channelid='20'

b) 此属性可以使用到织梦的 arclist/list等标签使用, 不局限在单个织梦标签

[field:sname/] ( 引用指定变量内容 )

#### 自定义模板引用 实战:

{dede:arclist typeid='37' row='5' orderby='pubdate' addfields='sname,sclass,sobject,smasage' channelid='17' }

<img src="[field:litpic/]">

<b>[field:sname/]</b><br/>

[field:sclass/]

<a href="[field:arcurl/]" class="stu\_button" >

[field:sobject/]

</a>

{/dede:arclist}

#### 自定义模板引用 实战2：

{dede:list pagesize='8' orderby='pubdate' titlelen='150' addfields='wine\_name\_cn, wine\_name\_fr' channelid='20' }

<a class="pro-pic" href="[field:arcurl/]" >

<div class="pro-img-window">

<img src="[field:litpic/]">

<p>

[field:wine\_name\_fr/]

</p>

</div>

<h2>

[field:title/]

</h2>

<i class="pro-money" >

</i>

</a>

{/dede:list}

<ul class="fanye\_pc" >

{dede:pagelist listitem="index,end,pageno"/}

</ul>

#### 自定义模板在“文档页调用”方式:

{dede:field.wine\_fr/} 可以在文档页面调用，自定义模板内容

### 关于织梦 调用 当前系统时间

{dede:php runphp='yes'}

$ntime = time();

@me=MyDate('Y m d',$ntime);

{/dede:php}

# 2019.6.16

### **<!-- 0. 列表组 -->**

**<!--**

**a) 核心: 父类list-group,子类list-item**

**b) active 选中状态**

**c) disabled 禁选状态**

**-->**

**<div class="container">**

**<ul class="list-group" >**

**<li class="list-group-item"> first1 </a>**

**<li class="list-group-item active "> first2 </a>**

**<li class="list-group-item"> first3 </a>**

**<li class="list-group-item disabled "> first3 </a>**

**</ul>**

**</div>**

#### **<!-- 1. 链接列表 -->**

**<!--**

**a) 父类依然是list-group与list-group-item配合**

**b) list-group-item-action 使鼠标悬停时有动态效果**

**c) 添加背景颜色如: list-group-item-dark/list-group-item-danger等...**

**-->**

**<div class="container">**

**<div class="list-group">**

**<a class="list-group-item list-group-item-action " href="#"> test1 </a>**

**<a class="list-group-item list-group-item-action " href="#"> test2 </a>**

**<a class="list-group-item list-group-item-action " href="#"> test3 </a>**

**<a class="list-group-item list-group-item-dark" href="#" > 有背景颜色的列表组1 </a>**

**<a class="list-group-item list-group-item-danger" href="#" > 有背景颜色的列表组2 </a>**

**</div>**

**</div>**

# **2019.6.20**

### 徽标,用于标记未读项，或者新项

#### **<!-- 0. 徽标,用于标记未读项,新项 -->**

**<!--**

**a) 我感觉他需要js来配合**

**b) bage bage-secondary/bage-danger 二者配合,bage会随机着父类进行变化**

**-->**

**<h1>**

**11111<span class="badge badge-secondary" >test</span>**

**</h1>**

**<h6>**

**22222<span class="badge badge-danger">test</span>**

**</h6>**

#### **<!-- 1. 药丸形状徽标 -->**

**<!--**

**a) badge badge-pill 二者配合,形状将发生改变**

**-->**

**<div class="container">**

**<span class="badge badge-success badge-pill" >111</span>**

**<span class="badge badge-success badge-pill" >111</span>**

**<span class="badge badge-success badge-pill" >111</span>**

**</div>**

#### **<!-- 2. 徽标,插入其他元素,成为共生关系 -->**

**<!--**

**a) 如果只有badge元素,那么他可以插入其它标签形成共生关系**

**-->**

**<div class="container-fluid" >**

**<a href="#" class="btn btn-success" role="button" >**

**test**

**<span class="badge badge-danger" >1</span>**

**</a>**

**</div>**

### **<!-- 0. 分页,当然只是css样式 -->**

**<!--**

**a) 父类pagination与子类page-item配合构成框架**

**在由子类的子类page-link进行链接样式处理**

**b) active与page-item同级配合,显示当前选择页,当然这需要js来操控**

**c) disabled 禁用选择,与page-item 同级**

**d) pagination-lg/pagination-sm 与pagination 同级操控分页大小**

**-->**

**<div class="container">**

**<ul class="pagination">**

**<li class="page-item disabled " >**

**<a class="page-link" href="#" > 上一页 </a>**

**</li>**

**<li class="page-item active " >**

**<a class="page-link" href="#" > 1 </a>**

**</li>**

**<li class="page-item" >**

**<a class="page-link" href="#" > 2 </a>**

**</li>**

**<li class="page-item" >**

**<a class="page-link" href="#" > 下一页 </a>**

**</li>**

**</ul>**

**<ul class="pagination pagination-lg " >**

**<li class=" page-item ">**

**<a class="page-link" href="#" > page1 </a>**

**</li>**

**<li class=" page-item ">**

**<a class="page-link" href="#" > page2 </a>**

**</li>**

**<li class=" page-item ">**

**<a class="page-link" href="#" > page3 </a>**

**</li>**

**</ul>**

**</div>**

#### **<!-- 1. 分页, 面包条导航 -->**

**<!--**

**a) 父类breadcrumb 与 子类breadcrumb-item 配合形成面包条状导航栏**

**-->**

**<div class="container">**

**<ul class="breadcrumb">**

**<li class="breadcrumb-item">**

**<a href="#" > sm </a>**

**</li>**

**<li class="breadcrumb-item">**

**<a href="#" > am </a>**

**</li>**

**<li class="breadcrumb-item">**

**<a href="#" > cm </a>**

**</li>**

**<li class="breadcrumb-item">**

**<a href="#" > fm </a>**

**</li>**

**</ul>**

**</div>**

### **<!-- 0. 进度条 -->**

**<!--**

**a) 父类progress 与 子类 progress-bar 配合形成进度条,在由width:%;来决定进度条长度**

**b) 默认进度条height:16px; 通过progress改变它的高度来实现不同高度的进度条**

**c) 当然也可以在进条中加入文字,用于显示进度( 通过JS来配合 )**

**d) 也可以更换进度条颜色如bg-danger**

**e) progress-bar-striped 动画进度条**

**f) 多个进度条progress-bar是可以在同一个progress容器里**

**-->**

**<div class="container">**

#### **<h2> 普通的进度条 </h2>**

**<div class="progress">**

**<div class="progress-bar" style="width:10%;" ></div>**

**</div>**

#### **<h2> 控制高度 </h2>**

**<div class="progress" style="height:50px;" >**

**<div class="progress-bar bg-danger " style="width:20%;" > 20% </div>**

**</div>**

#### **<h2> 条纹进度条 </h2>**

**<div class="progress" >**

**<div class="progress-bar bg-success progress-bar-striped" style="width:30%;" >**

**30%**

**</div>**

**</div>**

#### **<h2> 动画进度条 </h2>**

**<div class="progress">**

**<div class="progress-bar progress-bar-striped progress-bar-animated" style="width:40%;" >40%</div>**

**</div>**

#### **<h2> 多进度条 </h2>**

**<div class="progress">**

**<div class="progress-bar bg-success " style="width:20%;" > Project1 </div>**

**<div class="progress-bar bg-danger " style="width:30%;" > Project2 </div>**

**<div class="progress-bar bg-dark " style="width:25%;" > Project3 </div>**

**</div>**

**</div>**

### **<!-- 0.card 小卡片 -->**

**<!--**

**a) 父类card与子类:**

**头部: card-header**

**躯体: card-body ( 可以单独与父类配合 )**

**尾部: card-footer**

**b) bg-success 上背景颜色**

**c) text-danger 上文本颜色**

**-->**

**<div class="card">**

**<div class="card-header"> This is header </div>**

**<div class="card-body bg-success text-danger " >**

**This is body**

**</div>**

**<div class="card-footer"> This is footer </div>**

**</div>**

#### **<!-- 1. card 细节使用 -->**

**<!--**

**a) card-title 标题**

**card-text 文本**

**card-link 链接**

**-->**

**<div class="card">**

**<div class="card-body">**

**<h2 class="card-title" > test </h2>**

**<p class="card-text" > 我是文本 </p>**

**<a href="#" class="card-link" > 链接 </a>**

**</div>**

**</div>**

**</div>**

#### **<!-- 2. card的图片使用 -->**

**<!--**

**a) card容器可以进行控制宽度**

**b) card-img-top/card-img-bottom 图片上下分别使用,看下面的例子**

**-->**

**<div class="container">**

**<div class="card" style="width:350px;" >**

**<img class="card-img-top" src="test.jpg">**

**<div class="card-body">**

**<h2 class="card-title" > \_\_OO7\_\_ </h2>**

**<p> 成为一名优秀的黑客，是我终生奋斗的目标 </p>**

**<a class="btn btn-success" role="button" href="#" > 我的博客 </a>**

**</div>**

**</div>**

**<div class="card" style="width:350px;" >**

**<div class="card-body">**

**<h3 class="card-title" > \_\_OO7\_\_ </h3>**

**<p class="card-text" > 学习使我快乐 </p>**

**<a class="btn btn-outline-success" role="button" > 我的博客 </a>**

**</div>**

**<img class="card-img-bottom" src="test.jpg">**

**</div>**

#### **<!-- 3. card以图片为背景 -->**

**<!--**

**a) 先调用一张图片card-img-top**

**在使用card-img-overlay进行配合即可使图片为背景**

**-->**

**<div class="card" style="width:350px" >**

**<img src="test.jpg" class="card-img-top" >**

**<div class="card-img-overlay">**

**<h3 class="card-title" > \_\_OO7\_\_ </h3>**

**<p class="card-text" > 成为优秀的黑客后，那么成为顶级黑客也并非不可能 </p>**

**<a class="btn btn-danger"> 我的博客 </a>**

**</div>**

**</div>**

**</div>**

### **<!-- 0. 下拉菜单 -->**

**<!--**

**a) 父类: dropdown与下方配合( 其实下方这些元素,早在按钮组就已经相遇过 )**

**核心: dropdown-toggle 要与 data-toggle="dropdown"配合**

**dropdown-menu 隐藏菜单容器**

**dropdown-item 菜单项**

**b) 父类与子类核心配合共同构建，下拉菜单**

**-->**

**<div class="dropdown">**

**<button type="button" class="btn btn-success dropdown-toggle" data-toggle="dropdown" > 下拉菜单 </button>**

**<div class="dropdown-menu">**

**<a href="#" class="dropdown-item"> 1 </a>**

**<a href="#" class="dropdown-item"> 1 </a>**

**<a href="#" class="dropdown-item"> 1 </a>**

**</div>**

**</div>**

#### **<!-- 1. 下拉菜单,子菜单中的分割线/标题 -->**

**<!--**

**a) dropdown-toggle: 可以使按钮有下拉菜单小标记**

**b) data-toggle="dropdown": 没有他无法显示dropdown-menu的内容**

**c) dropdown-divider: 在dropdown-menu下拉菜单中产生分割线**

**d) dropdown-header: 在dropdown-menu下拉菜单中产生的标题**

**-->**

**<div class="dropdown">**

**<button type="button" class="btn btn-danger dropdown-toggle " data-toggle="dropdown" > 我是下拉菜单 </button>**

**<div class="dropdown-menu">**

**<h3 href="#" class="dropdown-header"> HEAD </h3>**

**<a href="#" class="dropdown-item" > 1 </a>**

**<a href="#" class="dropdown-item" > 1 </a>**

**<a href="#" class="dropdown-item" > 1 </a>**

**<div class="dropdown-divider"></div>**

**<h3 href="#" class="dropdown-header" > BODY </h3>**

**<a href="#" class="dropdown-item" > 2 </a>**

**<a href="#" class="dropdown-item" > 2 </a>**

**<a href="#" class="dropdown-item" > 2 </a>**

**</div>**

**</div>**

#### **<!-- 2. 下拉菜单活动项/禁用项 -->**

**<!--**

**a) 其实依然是老朋友，选中项"active",禁用项"disabled",为了练习在写一遍**

**b) dropdown-menu-right 可以改变子菜单x方向位置( 与dropdown-menu同级 )**

**c) 父类dropup变为上拉菜单( 注意,此函数类容易出现问题 )**

**-->**

**<div class="dropdown">**

**<button type="button" class=" btn btn-success dropdown-toggle " data-toggle="dropdown" > 活动/禁用项目实验 </button>**

**<div class="dropdown-menu dropdown-menu-right ">**

**<h3 class="dropdown-header" > head </h3>**

**<a class="dropdown-item" href="#" > 1 </a>**

**<a class="dropdown-item active " href="#" > 1 - ( 保持选中状态 ) </a>**

**<div class="dropdown-divider"></div>**

**<h3 class="dropdown-header"> footer </h3>**

**<a href="#" class="dropdown-item" > 2 </a>**

**<a href="#" class="dropdown-item disabled " > 2 - ( 被禁用 ) </a>**

**</div>**

**</div>**

**<div class="dropup">**

**<button type="button" class="btn btn-primary dropdown-toggle" data-toggle="dropdown">**

**Dropdown button**

**</button>**

**<div class="dropdown-menu">**

**<a class="dropdown-item" href="#">Link 1</a>**

**<a class="dropdown-item" href="#">Link 2</a>**

**</div>**

**</div>**

#### **<!-- 3. 分割下拉菜单，其实核心都一样，只是有个下拉三角号而已 -->**

**<!--**

**a) 这不是回忆一下btn-group的内容吗，我以为是由dropdown创造出,谁知道是btn-group**

**b) 核心在于: dropdown-toggle-split 与 形状标签<span class="caret" ></span>**

**dropdown-toggle-split: 可以与前一个标签无缝对接**

**<span class="caret" ></span>: 产生倒三角下拉形状**

**-->**

**<div class="container">**

**<div class="btn-group">**

**<button type="button" class="btn btn-success" > 111 </button>**

**<button type="button" class="btn btn-success dropdown-toggle dropdown-toggle-split " data-toggle="dropdown" >**

**<span class="caret" ></span>**

**</button>**

**<div class="dropdown-menu">**

**<a href="#" class="dropdown-item" > a </a>**

**<a href="#" class="dropdown-item" > b </a>**

**</div>**

**</div>**

**</div>**

**</div>**

### **<!-- 0. 折叠菜单 -->**

**<!--**

**a) 控制按钮:**

**data-toggle="collapse": 开启collapse模式**

**data-target="#id": 指定控制ID**

**注意:**

**a标签的href="#id"可以代替data-target="#id"**

**b) 内容容器:**

**collapse: 开启collapse模式**

**id: id号要与按钮指定的id对应**

**-->**

**<div class="container">**

**<button class="btn btn-success" data-toggle="collapse" data-target="#demotest" > 111 </button>**

**<a class="btn btn-danger" data-toggle="collapse" href="#demotest" >222</a>**

**<div id="demotest" class="collapse" >**

**<p>**

**asdasdasdasdasdasd**

**</p>**

**</div>**

#### **<!-- 1. 显示collapse -->**

**<!--**

**a) collapse show 直接显示出隐藏容器内容**

**-->**

**<button class="btn btn-danger" data-toggle="collapse" data-target="#demotest3" > 333 </button>**

**<div class="collapse show " id="demotest3" >**

**<p>**

**123213213213**

**</p>**

**</div>**

**</div>**

#### **<!-- 手风琴效果: 指定统一父类，配合隐藏 -->**

**<!--**

**a) 不要被card家族迷惑,card只是代替显示效果**

**b) 注意collapse要在同一个父类容器**

**c) 与隐藏容器collapse同级****data-parent="#父类ID",即可实现手风琴效果**

**-->**

**<div class="container">**

**<div id="father">**

**<div class="card">**

**<div class="card-header">**

**<a href="#demo0" class="btn btn-success" data-toggle="collapse" > 444 </a>**

**</div>**

**<div id="demo0" class="collapse" data-parent="#father" >**

**<div class="card-body" >**

**1111111111**

**</div>**

**</div>**

**</div>**

**<div class="card">**

**<div class="card-header">**

**<a href="#demo1" class="btn btn-success" data-toggle="collapse" >555</a>**

**</div>**

**<div class="collapse" id="demo1" data-parent="#father" >**

**<div class="card-body" >**

**222222222**

**</div>**

**</div>**

**</div>**

**</div>**

**</div>**

### **Jquery-代替data-parent="#父类ID",即可实现手风琴效果 的方法**

// 具体效果看bt4最后的练习实战作品

// 当点击.port-item时隐藏或者显示collapse

$('.port-item').click(function () {

$('.collapse').collapse('hide');

});

#### **// 使用方法**

****<div class="d-flex flex-row text-white align-items-stretch text-center">****

****// 控制隐藏的按钮****

****<div data-toggle="collapse" data-target="#home">****

****</div>****

****<div data-toggle="collapse" data-target="#test">****

****</div>****

****</div>****

****// 隐藏目标****

****<div id="home" class="collapse show"><div>****

****<div id=”test” class=”collapse” ></div>****

****<script>****

****// 必备jq****

****$('.port-item').click(function () {****

****$('.collapse').collapse('hide');****

****});****

****</script>****

# **2019.6.21**

### 关于织梦检测手机自动跳转代码

#### PC端模板首页index.htm

<meta http-equiv="mobile-agent" content="format=xhtml;url={dede:global.cfg\_mobileurl/}/index.php">

<script type="text/javascript">if(window.location.toString().indexOf('pref=padindex') != -1){}else{if(/AppleWebKit.\*Mobile/i.test(navigator.userAgent) || (/MIDP|SymbianOS|NOKIA|SAMSUNG|LG|NEC|TCL|Alcatel|BIRD|DBTEL|Dopod|PHILIPS|HAIER|LENOVO|MOT-|Nokia|SonyEricsson|SIE-|Amoi|ZTE/.test(navigator.userAgent))){if(window.location.href.indexOf("?mobile")<0){try{if(/Android|Windows Phone|webOS|iPhone|iPod|BlackBerry/i.test(navigator.userAgent)){window.location.href="{dede:global.cfg\_mobileurl/}/index.php";}else if(/iPad/i.test(navigator.userAgent)){}else{}}catch(e){}}}}</script>

#### PC端模板列表页list\_article.htm

<meta http-equiv="mobile-agent" content="format=xhtml;url={dede:global.cfg\_mobileurl/}/list.php?tid={dede:field.id/}">

<script type="text/javascript">if(window.location.toString().indexOf('pref=padindex') != -1){}else{if(/AppleWebKit.\*Mobile/i.test(navigator.userAgent) || (/MIDP|SymbianOS|NOKIA|SAMSUNG|LG|NEC|TCL|Alcatel|BIRD|DBTEL|Dopod|PHILIPS|HAIER|LENOVO|MOT-|Nokia|SonyEricsson|SIE-|Amoi|ZTE/.test(navigator.userAgent))){if(window.location.href.indexOf("?mobile")<0){try{if(/Android|Windows Phone|webOS|iPhone|iPod|BlackBerry/i.test(navigator.userAgent)){window.location.href="{dede:global.cfg\_mobileurl/}/list.php?tid={dede:field.id/}";}else if(/iPad/i.test(navigator.userAgent)){}else{}}catch(e){}}}}</script>

#### PC端模板内容页article\_article.htm

<meta http-equiv="mobile-agent" content="format=xhtml;url={dede:global.cfg\_mobileurl/}/view.php?aid={dede:field.id/}">

<script type="text/javascript">if(window.location.toString().indexOf('pref=padindex') != -1){}else{if(/AppleWebKit.\*Mobile/i.test(navigator.userAgent) || (/MIDP|SymbianOS|NOKIA|SAMSUNG|LG|NEC|TCL|Alcatel|BIRD|DBTEL|Dopod|PHILIPS|HAIER|LENOVO|MOT-|Nokia|SonyEricsson|SIE-|Amoi|ZTE/.test(navigator.userAgent))){if(window.location.href.indexOf("?mobile")<0){try{if(/Android|Windows Phone|webOS|iPhone|iPod|BlackBerry/i.test(navigator.userAgent)){window.location.href="{dede:global.cfg\_mobileurl/}/view.php?aid={dede:field.id/}";}else if(/iPad/i.test(navigator.userAgent)){}else{}}catch(e){}}}}</script>

# 2019.6.25

### H5引用字体

**@font-face {**

**font-family:'KUR';**

**src: url('../font/KUNSTLER.TTF');**

**}**

引用时:

**Font-family:’KUR’;**

## **<!-- 0. Bootstrap-导航,基本骨骼 -->**

**<!--**

**a) 核心:**

**父类: nav**

**子类: nav-item**

**子类: nav-link**

**b) 使用justify-content-center/start/end来控制导航x排序**

**-->**

**<div class="container">**

**<ul class="nav justify-content-center ">**

**<li class="nav-item">**

**<a href="#" class="nav-link" > 111 </a>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link" > 111 </a>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link" > 111 </a>**

**</li>**

**</ul>**

**</div>**

#### **<!-- 1. 垂直导航 -->**

**<!--**

**a) 垂直布局，其实利用的就是flex布局,**

**-->**

**<div class="container">**

**<ul class=" nav flex-column">**

**<li class="nav-item">**

**<a href="#" class="nav-link" > 222 </a>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link" > 222 </a>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link" > 222 </a>**

**</li>**

**</ul>**

#### **<!-- 2. 选项卡/胶囊/等宽导航 -->**

**<!--**

**a) 选项卡导航,父类: nav nav-tabs**

**b) 胶囊导航,父类: nav nav-pills**

**c) 等宽导航,父类: nav nav-tabs nav-justified**

**-->**

**<ul class="nav nav-tabs nav-justified " >**

**<li class="nav-item">**

**<a href="#" class="nav-link" > 111 </a>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link" > 111 </a>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link" > 111 </a>**

**</li>**

**</ul>**

**<ul class="nav nav-pills" >**

**<li class="nav-item ">**

**<a href="#" class="nav-link bg-danger text-light " > 333 </a>**

**</li>**

**<li class="nav-item ">**

**<a href="#" class="nav-link" > 333 </a>**

**</li>**

**<li class="nav-item ">**

**<a href="#" class="nav-link" > 333 </a>**

**</li>**

**</ul>**

#### **<!-- 3. 下拉导航栏 -->**

**<!--**

**a) 下拉菜单的导航栏,**

**同样是老相识dropdown**

**下拉菜单父类与nav-link同级**

**-->**

**<ul class="nav nav-tabs">**

**<li class="nav-item">**

**<a href="#" class="nav-link" > 444 </a>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link dropdown-toggle " data-toggle="dropdown" > 444 </a>**

**<div class="dropdown-menu">**

**<a href="#" class="dropdown-item" >1</a>**

**<a href="#" class="dropdown-item" >1</a>**

**</div>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link" > 444 </a>**

**</li>**

**</ul>**

#### **<!--4. 动态导航( 可切换div与按钮配合 )-->**

**<!--**

**a) 导航选项按钮: nav-tabs, 设定: data-toggle="tab" href="#id"**

**b) 切换容器设定: tab-content**

**父类: tab-content**

**子类: tab-pane container fade , id="#id"**

**tab-pane: 核心**

**container: 框架**

**fade: 切换时有淡入效果**

**id: 要与上方按钮对应**

**-->**

**<ul class="nav nav-tabs">**

**<li class="nav-item">**

**<a href="#test1" data-toggle="tab" class="nav-link" > num1 </a>**

**</li>**

**<li class="nav-item">**

**<a href="#test2" data-toggle="tab" class="nav-link" > num1 </a>**

**</li>**

**<li class="nav-item">**

**<a href="#test3" data-toggle="tab" class="nav-link" > num1 </a>**

**</li>**

**</ul>**

**<div class="tab-content">**

**<div class="container tab-pane fade" id="test1" >**

**<p styl"height:50px;" class="bg-danger" >1</p>**

**</div>**

**<div class="container tab-pane fade" id="test2" >**

**<p styl"height:50px;" class="bg-success" >2</p>**

**</div>**

**<div class="container tab-pane fade" id="test3" >**

**<p styl"height:50px;" class="bg-dark" >3</p>**

**</div>**

**</div>**

#### **<!-- 导航切换实战 -->**

**<!--**

**a) 关于active:**

**初始显示内容,在nav-link/tab-pane 同级加入active**

**且记fade会影响active所以不能共同存在**

**-->**

**<ul class="nav nav-pills ">**

**<li class="nav-item">**

**<a class="nav-link active" data-toggle="tab" href="#tt1" > home </a>**

**</li>**

**<li class="nav-item">**

**<a class="nav-link" data-toggle="tab" href="#tt2" > danger </a>**

**</li>**

**<li class="nav-item">**

**<a class="nav-link" data-toggle="tab" href="#tt3" > yellow </a>**

**</li>**

**</ul>**

**<div class="tab-content">**

**<div class="container tab-pane active" id="tt1" >**

**<p style="height:100px;" class="bg-danger" >1111</p>**

**</div>**

**<div class="container tab-pane" id="tt2" >**

**<p style="height:100px;" class="bg-success" >1111</p>**

**</div>**

**<div class="container tab-pane" id="tt3" >**

**<p style="height:100px;" class="bg-dark" >1111</p>**

**</div>**

**</div>**

**</div>**

### **<!-- 0. BOOTSTARP - 标准主导航栏 -->**

**<!--**

**a) nav父类: navbar navbar-expand-sm**

**navbar: 为必须**

**navbar-expand-xl/lg/md/sm: 响应式导航，达到一定分辨率变为多行**

**ul子类: navbar-nav: 正常nav变为navbar-nav**

**li子类: nav-item**

**a子类: nav-link**

**b) navbar-expand-xl/lg/md/sm 没有它则导航栏为垂直,有他则在一定分辨率为垂直**

**c) 控制导航栏背景: bg-dark/bg-light等...**

**d) 控制导航栏字体颜色: navbar-light/dark( navbar-颜色为相反色 )等...**

**e) 同样active/disabled同样起作用**

**-->**

**<nav class=" navbar navbar-expand-sm bg-light ">**

**<ul class="navbar-nav">**

**<li class="nav-item">**

**<a href="#" class="nav-link" > 111 </a>**

**</li>**

**<li class="nav-item ">**

**<a href="#" class="nav-link" > 222 </a>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link" > 333 </a>**

**</li>**

**</ul>**

**</nav>**

#### **<!-- 因无navbar-expand-xl/lg/md/sm所以为垂直 -->**

**<nav class="navbar bg-dark navbar-dark" >**

**<ul class="navbar-nav">**

**<li class="nav-item">**

**<a href="#" class="nav-link" >**

**test1**

**</a>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link" >**

**test1**

**</a>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link" >**

**test1**

**</a>**

**</li>**

**</ul>**

**</nav>**

#### **<!-- 1. navbar导航栏中logo的使用 -->**

**<!--**

**a) navbar-brand: 专门用来搞logo存放**

**甚至在这个容器里图片的width既然可以等比例缩放( 正常情况只有高度缩放才等比 )**

**-->**

**<nav class="navbar navbar-expand-sm bg-danger navbar-dark " >**

**<a href="#" class="navbar-brand" >**

**<img src="logo.png" style="width:50px;" >**

**</a>**

**<ul class="navbar-nav">**

**<li class="nav-item">**

**<a href="#" class="nav-link" >**

**tt1**

**</a>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link" >**

**tt2**

**</a>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link" >**

**tt3**

**</a>**

**</li>**

**</ul>**

**</nav>**

#### **<!-- 2. 响应式折叠按钮导航栏( 移动端会变成菜单按钮 ) -->**

**<!--**

**a) 分析核心: collapse套装与，按钮类navbar-toggler控制菜单类navbar-collapse，的组合成响应式折叠按钮菜单**

**b) collapse本身就为折叠菜单而打造，注意navbar-expand-xl/lg/md/sm没有他菜单会隐藏起来**

**c) 0. 写入导航栏骨架**

**1. 嵌入式,button标签,上collapse按钮类,在额外加入navbar-toggler毕竟在别人家地盘**

**别忘记嵌入icon, class="navbar-toggler-icon"**

**2. 上collapse菜单类,注意菜单容器要包含ul,并且额外加入navbar-collapse**

**d) 核心:**

**collapse套装**

**按钮类navbar-toggler**

**菜单类navbar-collapse**

**e) 注意导航栏的颜色改变使用 navbar-light/等…否则默认按钮icon以及文字颜色将显示不正常,不过不用navbar-light(颜色)可以自定义icon和文字**

**-->**

**<nav class="navbar navbar-expand-sm bg-dark navbar-dark " >**

**<a href="#" class="navbar-brand" >**

**LOGO**

**</a>**

**<button type="button" class="navbar-toggler" data-toggle="collapse" data-target="#zdtest1" >**

**<span class="navbar-toggler-icon" ></span>**

**</button>**

**<div class="navbar-collapse collapse" id="zdtest1" >**

**<ul class="navbar-nav">**

**<li class="nav-item">**

**<a href="#" class="nav-link" >a1</a>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link" >a2</a>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link" >a3</a>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link" >a4</a>**

**</li>**

**</ul>**

**</div>**

**</nav>**

**</div>**

#### **<!-- 3. 有下拉菜单的导航栏 -->**

**<!--**

**a) 写入navbar骨架**

**b) 选中某一个nav-item**

**c) 正常写入dropdown的三大核心**

**d) 在将选中的选中项nav-item同级写入dropdown**

**因为这样可以使下拉菜单正确的对应选项**

**-->**

**<div class="container">**

**<nav class="navbar navbar-expand-sm bg-success navbar-dark ">**

**<ul class="navbar-nav">**

**<li class="nav-item">**

**<a href="#" class="nav-link" > aaa </a>**

**</li>**

**<li class="nav-item">**

**<a href="#" class="nav-link" > aaa </a>**

**</li>**

**<li class="nav-item dropdown ">**

**<a href="#" class="nav-link dropdown-toggle " data-toggle="dropdown" > aaa </a>**

**<div class="dropdown-menu">**

**<a class="dropdown-item" > 111 </a>**

**<a class="dropdown-item" > 111 </a>**

**</div>**

**</li>**

**</ul>**

**</nav>**

**</div>**

#### **<!-- 4. 给导航栏加入表单 -->**

**<!--**

**a) 直接form嵌入到nav中**

**b) 但是注意表单加入class样式**

**c) 父类: form-inline**

**子类: form-control**

**d) 按钮直接上btn**

**e) navbar-text直接搞文字，注意在ul之外,nav中**

**-->**

**<div class="container">**

**<nav class="navbar navbar-expand-sm bg-light navbar-light ">**

**<form class="form-inline" >**

**<input class="form-control" type="text" >**

**<button type="button" class="btn btn-success" > Search </button>**

**</form>**

**<ul class="navbar-nav" >**

**<li class="nav-item">**

**<a href="#" class="nav-link" > 233 </a>**

**</li>**

**</ul>**

**<span class="navbar-text">**

**直接搞文字**

**</span>**

**</nav>**

**</div>**

#### **<!-- 5. 固定导航栏 -->**

**<!--**

**核心在于: fixed-top/bottom**

**-->**

**<nav class="navbar navbar-expand-sm fixed-bottom bg-dark navbar-dark">**

**<ul class="navbar-nav">**

**<li class="nav-item">**

**<a href="#" class="nav-link" >**

**fixed-bottom**

**</a>**

**</li>**

**</ul>**

**</nav>**

### **<!-- 0. 面包屑导航 -->**

**<!--**

**a) 其实这个学过,**

**b) 核心:**

**父类: breadcrumb**

**子类: breadcrumb-item**

**c) 简单实用的小导航**

**-->**

**<ol class="breadcrumb" >**

**<li class="breadcrumb-item" >**

**<a href="#"> test1 </a>**

**</li>**

**<li class="breadcrumb-item" >**

**<a href="#"> test1 </a>**

**</li>**

**</ol>**

#### **<!-- 小实列 -->**

**<nav class="breadcrumb" >**

**<a href="#" class="breadcrumb-item" > testtest </a>**

**<a href="#" class="breadcrumb-item" > testtest </a>**

**</nav>**

# **2019.6.27**

### Mysql导出sql

1.导出数据库（sql脚本）

mysqldump -u 用户名 -p 数据库名 > 导出的文件名

### 取消背景颜色

background-color:transparent;

# 2019.7.5

## <!-- 0. 表单控件,text,passwprd -->

<!--

a) 其实 type=""这些都是H5已经有的属性

b) 容器: form-group 块级表单必备

子类: form-control 使表单为块级样式

c) form-control-lg 可以放大表单大小

-->

<div class="form-group">

<label for="usr" > 用户: </label>

<input type="text" id="usr" class="form-control" >

</div>

<div class="form-group">

<label for="pas"> 密码: </label>

<input type="password" id="pas" class="form-control" >

</div>

<div class="form-group">

<label for="texta"> 留言: </label>

<textarea id="texta" class="form-control" rows="6" ></textarea>

</div>

### <!-- 1. 多选框 -->

<!--

a) 多选框:

容器: form-check

父类: form-check-label

子类: form-check-input

b) 如果容器 form-check 加入 form-check-inline 则块级表单变为内联表单

-->

#### <!-- 块级选项框 -->

<div class="form-check">

<label class="form-check-label">

<input type="checkbox" class="form-check-input" value="" > choose1

</label>

</div>

<div class="form-check disabled ">

<label class="form-check-label">

<input type="checkbox" class="form-check-input" value="" disabled > choose2

</label>

</div>

#### <!-- 内联选项框 -->

<div class="form-check form-check-inline ">

<label class="form-check-label">

<input type="checkbox" class="form-check-input" value="" > choose3

</label>

</div>

<div class=" form-check form-check-inline">

<label class="form-check-label" >

<input type="checkbox" class="form-check-input" value="" checked > choose4

</label>

</div>

### <!-- 2. 单选框 -->

<!--

a) 块级单选框:

容器: radio

表单input使用type="radio"属性配合label,包含在radio容器中

b) 内联单选框:

容器: radio 加入 radio-inline 即可变为内联单选框

-->

#### <!-- 块级选项框 -->

<div class="radio">

<label>

<input type="radio" name="test" > man

</label>

</div>

<div class="radio">

<label>

<input type="radio" name="test" > woman

</label>

</div>

<div class="radio disabled ">

<label>

<input type="radio" name="test" disabled > disabled

</label>

</div>

#### <!-- 内联选项框 -->

<div class="radio radio-inline ">

<label>

<input type="radio" name="test1" > test1

</label>

</div>

<div class="radio radio-inline ">

<label>

<input type="radio" name="test1" > test2

</label>

</div>

<div class="radio radio-inline disabled">

<label>

<input type="radio" name="test1" disabled > test3

</label>

</div>

### <!-- 3. 表单下拉菜单 -->

<!--

a) 核心在于" form-control "与HTML5的表单select进行配合

b) 父类: form-group

子类: form-control

c) HTML5 - select下拉菜单,加入"multiple"属性则变为多选下拉菜单

-->

<div class="form-group">

<label for="select0" >

#### 单下拉选项

</label>

<select class="form-control" for="select0" >

<option> 1 </option>

<option> 1 </option>

<option> 1 </option>

</select>

<br/>

<label for="select1" >

#### 多下拉选项菜单( 使用ctrl或者shift配合鼠标多选 )

</label>

<select multiple class="form-control">

<option> a </option>

<option> b </option>

<option> c </option>

<option> d </option>

<option> e </option>

</select>

</div>

</form>

</div>

## <!-- 0. 输入框组 -->

<!--

a) 其实输入框组主要围绕着，input-group-append,input-group-prepend,使输入表单开头或者结尾有文字

b) 容器: input-group

父类: input-group-prepend 在开头部分加入文字

input-group-append 在结尾部分加入文字

子类: input-group-text 文本标签设置文字样式

( 上方这几个函数为核心,其它演示也只是应用展示 )

c) 注意input-group-prepend与input标签同级

d) mb-4: 是margin-bottom: (2\*x)px ;以4px为基础\*2进行缩放

e) 大/小的表单,input-group-sm/lg 注意与"容器同级"

-->

<form>

### <!-- 头部信息演示 -->

<div class="input-group mb-4 input-group-lg ">

<div class="input-group-prepend">

<span class="input-group-text"> UserName: </span>

</div>

<input type="text" class="form-control" placeholder=" UserName " >

</div>

#### <!-- 尾部信息演示 -->

<div class="input-group">

<input type="password" class="form-control" placeholder=" PassWord " >

<div class="input-group-append">

<span class="input-group-text"> PassWord: </span>

</div>

</div>

#### <!-- 单行多框输入 -->

<div class="input-group mt-4 ">

<div class="input-group-prepend">

<span class="input-group-text"> test0 </span>

<span class="input-group-text"> test1 </span>

</div>

<input type="text" class="form-control" placeholder="test0" >

<input type="text" class="form-control" placeholder="test1" >

</div>

</form>

</div>

<div class="container">

<form>

### <!-- 1. 多选框/单选框输入，开头/结尾 -->

<!--

a) input-group -> input-group-prepend -> input-group-text -> 嵌入表单( 与上方嵌套文字无异 )

-->

#### <!-- 多选框 -->

<div class="input-group">

<div class="input-group-prepend">

<div class="input-group-text">

<input type="checkbox">

</div>

</div>

<input type="text" class="form-control" placeholder="write something!!!" >

</div>

#### <!-- 单选框 -->

<div class="input-group">

<div class="input-group-prepend">

<div class="input-group-text">

<input type="radio" name="test" >

</div>

</div>

<input type="text" class="form-control" placeholder="what's your drink something?" >

</div>

<div class="input-group">

<div class="input-group-prepend">

<div class="input-group-text">

<input type="radio" name="test" >

</div>

</div>

<input type="text" class="form-control" placeholder="what's your eat something?" >

</div>

#### <!-- 2. 按钮输入栏 -->

<!--

a) 按钮无需嵌套在input-group-text中,嵌套在input-group-prepend/input-group-append

b) 其它都大差不差

-->

<div class="input-group">

<input type="text" class="form-control" >

<div class="input-group-append">

<button type="button" class="btn btn-success" > Search </button>

</div>

</div>

#### <!-- 有下拉菜单的按钮输入栏 -->

<label for="search" > 单机开始搜索 </label>

<div class="input-group">

<div class="input-group-prepend">

<button type="button" class="btn btn-outline-success dropdown-toggle" data-toggle="dropdown" > 全部搜索 </button>

<div class="dropdown-menu">

<span class="dropdown-item"> 免费区域 </span>

<span class="dropdown-item"> 打折区域 </span>

<span class="dropdown-item"> 精品区域 </span>

</div>

</div>

<input type="text" id="search" class="form-control" placeholder="Search!!!" >

</div>

<form>

</div>

<form>

## <!-- 0. 自定义复选框/单选框 -->

<!--

a) 自定义表单系列好像是要突出bootstrap的美状

b) 容器: custom-control custom-checkbox/custom-radio组成

c) 子类: checkbox表单: custom-control-input

d) label标签: custom-control-label

e) custom-control-inline: 使自定义的表单变为内联元素,可以使表单变为同一行

-->

### <!-- 块级表单 -->

<div class="custom-control custom-checkbox ">

<input type="checkbox" class="custom-control-input" id="test" >

<label for="test" class="custom-control-label" > 自定义复选框 </label>

</div>

### <!-- 内联表单 -->

<div class="custom-control custom-radio custom-control-inline ">

<input type="radio" class="custom-control-input" name="choose" id="test2" >

<label class="custom-control-label" for="test2" > man </label>

</div>

<div class="custom-control custom-radio custom-control-inline">

<input type="radio" class="custom-control-input" name="choose" id="test3" >

<label class="custom-control-label" for="test3" > woman </label>

</div>

### <!-- 1.自定义select选择菜单 -->

<!--

在select标签中加入.custom-select即成为自定义表单拥有Bootstrap设计的样式( 具有响应式 )

custom-select-sm: 小的选择菜单

custom-select-lg: 大的选择菜单

-->

<select class="custom-select" >

<option selected > 选择 </option>

<option value="google" > google </option>

<option value="baidu" > baidu </option>

<option value="sogou" > sogou </option>

</select>

#### <!-- custom-select-sm 小的选择菜单 -->

<select class="custom-select-sm">

<option selected > 1 </option>

<option value="2" > 2 </option>

<option value="3" > 3 </option>

</select>

#### <!-- custom-select-lg 大的选择菜单 -->

<select class="custom-select-lg">

<option selected> A </option>

<option> B </option>

<option> C </option>

</select>

### <!-- 2. 自定义滑块控件 -->

<!--

a) 核心:

容器: .custom-file

子类: 标签input的type="range" class="custom-range"

子类: label加入 class="custom-file-input"

-->

<input type="range" id="range0" name="" class=" custom-range ">

### <!-- 3. 自定义上传控件 -->

<div class="custom-file">

<label for="file0" class="custom-file-label" > 选择文件</label>

<input type="file" class="custom-file-input" id="file0" >

</div>

</form>

# 2019.7.8

## 轮播图

### <!-- 0. 只有,图片轮播 -->

<!--

a) 主要容器:

.carousel data-ride="carousel" 必备元素

.slide - 动画效果

b) 图片容器:

主: .carousel-inner

子: .carousel-item

注意格式，且第一个显示的div用active属性标记

修改容器大小:

与carousel-inner同级，通过css3来改变容器大小

-->

<style>

/\* 保持图片填充满容器 \*/

.carousel-inner img {

width: 100%;

height: 100%;

}

</style>

<div class="carousel slide" data-ride="carousel" >

<div class="carousel-inner" style="width:1920px; height:800px;" >

<div class="carousel-item active ">

<img src="1.jpg">

</div>

<div class="carousel-item">

<img src="2.jpg">

</div>

</div>

</div>

#### <!-- 1. 有图片轮播，指示符 -->

<!--

a) 指示符

b) 容器: 注意是ul标签carsousel-indicators

子类: li标签加入data-target="主容器id号" data-slide-to="图对应数" , 注意第0个符加.active属性

-->

<div id="test" class="carousel slide " data-ride="carousel" >

<!-- 指示符 -->

<ul class="carousel-indicators">

<li data-target="#test" data-slide-to="0" class="active" ></li>

<li data-target="#test" data-slide-to="1" ></li>

</ul>

<!-- 轮播图 -->

<div class="carousel-inner">

<div class="carousel-item active ">

<img src="1.jpg">

</div>

<div class="carousel-item">

<img src="2.jpg">

</div>

</div>

</div>

#### <!-- 2. 有图片轮播,指示符,切换按钮 -->

<!--

容器: 注意id以及,data-ride="carousel"属性

.carousel为必备核心;

.slide为动画效果

切换按钮:

父类: carousel-control-prev/next

href="容器ID"

data-slide="prev/next"必备属性

子类: 用于创建按钮小图标

carousel-control-prev/next-icon

banner描述: 直接在carousel-item嵌入carousel-caption

-->

<div id="test1" class="carousel slide" data-ride="carousel" >

<ul class="carousel-indicators">

<li data-target="#test1" data-slide-to="0" class="active" ></li>

<li data-target="#test1" data-slide-to="1" ></li>

</ul>

<div class="carousel-inner">

<div class="carousel-item active ">

<img src="1.jpg">

<div class="carousel-caption">

<h3>第一张图片描述标题</h3>

<p>描述文字!</p>

</div>

</div>

<div class="carousel-item">

<img src="2.jpg">

</div>

</div>

<!-- 按钮 -->

<a class="carousel-control-prev" href="#test1" data-slide="prev" >

<span class="carousel-control-prev-icon" ></span>

</a>

<a class="carousel-control-next" href="#test1" data-slide="next" >

<span class="carousel-control-next-icon"></span>

</a>

</div>

## <!-- 0. 模态框（ modal弹窗显示内容 ） -->

<!--

a) 按钮:

核心 data-toggle="modal" - data-target="指定modal,ID"

b) 模态框:

0. 先来个3连套: .modal - .modal-diglog - .modal-content

1. fade: 使有动画效果

2. .modal-diglog可以控制总体大小,使用 modal-bg/sm

c) 三连击完毕后,接着头,身,尾

0. .modal-header - .modal-body - .modal-footer

1. data-dismiss="modal" 为关闭按钮核心

X号关闭按钮: class="close"显示样式 , &times;为x号小图标

普通关闭按钮: 直接上btn

-->

<div class="container">

### <!-- 单机按钮触发,模态框 -->

<button type="button" class="btn btn-success" data-toggle="modal" data-target="#test0" > test0 </button>

<!-- 模态框 -->

<div class="modal fade" id="test0" >

<div class="modal-dialog modal-lg "> <!-- 这里可以控制总体大小 -->

<div class="modal-content">

<div class="modal-header">

<h3>test</h3>

<button type="button" class="close" data-dismiss="modal" > &times; </button> <!-- 头部关闭按钮 -->

</div>

<div class="modal-body">

我是中心内容

</div>

<div class="modal-footer">

<button type="button" class="btn btn-danger" data-dismiss="modal" > 关闭 </button> <!-- 关闭按钮,核心属性相同,只是样式不同 -->

</div>

</div>

</div>

</div>

</div>

## <!-- 0. 小提示框 -->

<!--

a) 提示框3个属性+jquery构成

b) 属性:

data-toggle="tooltip": 开启提示

data-placement="top/bottom/left/right": 确定提示位置

title="写入提示信息"

c) jquery:

目前并不清楚jquery的用法

<script>

$( document ).ready( function(){ $('[data-toggle="tooltip"]').tooltip(); });

</script>

-->

<a href="#" data-toggle="tooltip" data-placement="left" title="我是提示信息" > 提示 </a>

<script>

$(document).ready(

function(){

$('[data-toggle="tooltip"]').tooltip();

}

);

</script>

### <!-- 1. 弹出框,有标题的提示框 -->

<!--

a) 5属性+jquery

b) 属性:

0. data-toggle="popover": 开启弹出框

1. data-placement="top": 设定弹出框位置

2. title="设定标题"

3. data-content="弹出框内容"

4. data-trigger="": 控制弹出效果

默认情况: 单机进行/打开/关闭弹出框操作

focus: 单击打开,单机其它地方关闭

hover: 鼠标悬停时打开

c) jquery:

<script> $(document).ready( function(){ $('[data-toggle="tooltip"]').tooltip(); }); </script>

-->

<br/>

<a href="#" data-toggle="popover" data-placement="bottom" title="我是标题" data-content="我是内容" > 弹出( 默认,单机打开/关闭 ) </a><br/>

<a href="#" data-toggle="popover" data-placement="bottom" title="我是标题" data-content="我是内容" data-trigger="focus" > 弹出( 单机打开,其它地方单机关闭 ) </a><br/>

<a href="#" data-toggle="popover" data-placement="bottom" title="我是标题" data-content="我是内容" data-trigger="hover" > 弹出( hover ) </a><br/>

<script>

$(document).ready(

function(){

$('[data-toggle="popover"]').popover();

}

);

</script>

## <!-- 0. scroll滚动监听 -->

<!--

a) 其实核心就3步

b) 第一步,开启监听:

0. 在body标签加入

data-spy="scroll", 开启监听

data-target=".navbar", 设定监听的导航栏

data-offset="50", 设定滚动条监听敏感度偏差

1. body标签为 position:relative;

c) 第二步,创建导航栏:

0. 导航栏class名对应上方data-target=""

1. 设定a标签的href="id/类名"

注意: href="#s1",href=".s1"一个是监控id一个监控类名

2. 其它样式无所谓

d) 第三步,创建对应div:

0. id="s1"/class="s1", 与上方导航栏的a标签href="对应ID/class"即可

1. 每个div设定padding:50px 0px;, 更加精确的操控监听跳转位置,适量调整即可

-->

<style>

body {

position: relative;

}

#s1,#s2,#s3,#s4{

padding:56px 0px;

}

</style>

<body data-spy="scroll" data-target=".navbar" data-offset="50" >

<nav class="navbar navbar-expand-sm bg-dark navbar-dark fixed-top ">

<ul class="navbar-nav" >

<li class="nav-item">

<a class="nav-link" href="#s1" > section1 </a>

</li>

<li class="nav-item">

<a class="nav-link" href="#s2" > section2 </a>

</li>

<li class="nav-item">

<a class="nav-link" href="#s3" > section3 </a>

</li>

<li class="nav-item">

<a class="nav-link" href="#s4" > section4 </a>

</li>

</ul>

</nav>

<div class="container bg-success" id="s1" style="height:500px;" >

<h1 class="display-3" >

Section1

</h1>

</div>

<div class="container bg-danger" id="s2" style="height:500px;" >

<h1 class="display-3" >

Section2

</h1>

</div>

<div class="container bg-light" id="s3" style="height:500px;" >

<h1 class="display-3" >

Section3

</h1>

</div>

<div class="container bg-dark" id="s4" style="height:500px;" >

<h1 class="display-3" >

Section4

</h1>

</div>

## <!-- 常用class代替部分常用css -->

<div class="container" style="margin-top:50px;" >

### <!--0. border边框 -->

<!--

border: 默认边框

border-danger: 可以修改边框颜色

border-top-0: 可以取消上方显示边框,上下左右类似

-->

<div class="border border-bottom-0 border-danger wh"></div>

### <!-- 1. rounded圆角 -->

<!--

rounded: 默认圆角

rounded-top: 指定圆角

rounded-circle: 圆形

rounded-0: 取消圆角

-->

<div class="rounded wh bg-danger "> 默认圆角 </div>

<div class="rounded-top wh border "> 指定圆角 </div>

<div class="rounded-circle wh border"> 圆形 </div>

<div class="rounded-0 wh border "> 取消圆角 </div>

</div>

<div class="container">

### <!-- 2. float-left/right浮动 -->

<!--

a) 普通浮动:

float-left/right: 左右浮动

clearfix: 清除浮动

b) 响应式浮动

float-sm-right: 屏幕大于sm时为右浮动

float-sm/md/lg/xl-right

float-none: 无浮动

-->

<div class="clearfix">

<div class="float-left border wh">111</div>

<div class="float-right border wh">222</div>

</div>

<h3 class="clearfix" > 响应式浮动 </h3>

<div class="clearfix">

<span class="float-sm-right"> 屏幕大于sm时为右浮动 </span>

<span class="float-none" style="color:red;" > 无浮动 </span>

</div>

<h3 class="clearfix" > 水平剧中 </h3>

### <!-- 3. mx-auto水平剧中 -->

<div class="mx-auto wh border border-danger"> 水平剧中 </div>

</div>

### <!-- 4. w-\*,h-\*宽度/高度 -->

<!--

w-\*: 控制宽度,w-25 相当于 width:25%;

h-\*: 控制高度,w-100 相当于 height:100%;

mw-100/mh-100: 最大宽度/高度

-->

<div class="container" style="height:500px;" >

<div class=" w-25 h-50 bg-danger float-left"></div>

<div class=" w-25 h-100 bg-info float-left "></div>

</div>

## <!-- media多媒体对像 -->

<!--

a) 第一印象,感觉并不常用

b) 父类: media

子类: media-body

c) media 容器默认开启flex模式,所以align-self-\*可以操控子元素垂直方向

d) media容器内,内容按照flex来搞就可以

-->

<div class="media border p-3">

<img src="https://static.runoob.com/images/mobile-icon.png" alt="John Doe" class="mr-3 mt-3 rounded-circle" style="width:60px;">

<div class="media-body">

<h4>菜鸟教程</h4>

<p>学的不仅是技术，更是梦想！！！</p>

<div class="media border p-3">

<img src="https://static.runoob.com/images/mobile-icon.png" alt="John Doe" class="mr-3 mt-3 rounded-circle" style="width:60px;">

<div class="media-body">

<h4>菜鸟教程</h4>

<p>学的不仅是技术，更是梦想！！！</p>

</div>

</div>

</div>

</div>

<!---->

</div>

## <!-- 0. flex弹性布局 -->

<!--

a) 开启弹性布局:

d-flex: 相当于 display:flex;

d-inline-flex: 相当于 display:inline-flex;

b) flex排序方向:

flex-row,flex-row-reverse: 水平方向排序布局 - flex-wrap:row; - flex-wrap:row-reverse;

flex-column,flex-column-reverse: 垂直方向排序布局 - flex-wrap:column; - flex-wrap:column-reverse;

c) 内容排序:

x方向:

0. 水平方向排序控制:

justify-content-start: 相当于 justify-content:flex-start;

\*: justify-content-start/end/center/between/around( 我已经理解flex,所以这里有不做备注啦 )

1. 等宽:

注意设定到子元素

flex-fill: 相当于 flex:1 1 auto;

2. 扩展:

占领剩余区域

flex-grow-1: flex-grow:1;

3. 数字排序:

order-1: 相当于 order:1;

4. 换行:

flex-wrap: 换行 - flex-wrap:wrap;

flex-nowrap: 不换行 - flex-wrap:nowrap;( 默认 )

flex-wrap-reverse: 反转换行 - flex-wrap:wrap-reverse;

y方向:

0. 垂直方向排序控制:

align-content-start: 相当于 align-content:flex-start;

\*: align-content-start/end/center/between/around

1. 单行垂直方向排序控制:

align-items-stretch: 相当于 align-items:stretch; 高度填充完容器

\*: align-items-start/end/center/between/around/stretch

指定元素:

0. 子元素垂直方向单独控制排序

align-self-start: 相当于 align-self:flex-start;

align-self-baseline: 文字于文字进行对齐

\*: align-self-start/end/center/between/around/baseline/stretch

d) 外/内边距:

mt-1: 相当于 margin-top:0.25rem;

mr-auto: 相当于 margin-right:auto; 配合flex的子元素,能实现更自由的排序位置操控

pt-1: 相当于 padding-top:0.25rem;

e) 响应式,弹性布局

0. 无疑是中间加入,不过要注意源语法完整性 \*: sm/md/lg/xl

1. 如: justify-content-sm-end: 当屏幕大于sm时,为flex-end排序方式

-->

#### <!-- 等宽 -->

<div class="d-flex" >

<div class="flex-fill bg-danger ">1</div>

<div class="flex-fill bg-info">2</div>

<div class="flex-fill bg-success">3</div>

</div>

#### <!-- 扩展 -->占领剩余区域

<div class="d-flex">

<div class="bg-success">test1</div>

<div class="bg-info">test2</div>

<div class="flex-grow-1 bg-danger">3</div>

</div>

</div>

## ---至此BootStrap已经全部研习完毕，开始实战笔记---

# 2019.7.11

## Bootstrap-内外边距

my-lg-4: 屏幕大于lg时有margin: 1.5rem 0px;小于lg则没有

mx-4: 有margin: 0px 1.5rem;

m-4: 相当于margin: 1.5rem;

注意所有的m-num/p-num其中num<=5超出范围则无效

## HTML5-锚点跳转

<a href=”#test”> 单击跳转 </a> <!— 注意与id号对应即可 -->

<div id=”test”>

会直接跳转到此标签

</div>

# 2019.7.12

## Bootstrap实战练习项目1

### BootStrap初始模板在: “ 2019.7.8/main\_templates/templates.html ”

去GitHub

### Bootstrap-导航监控以及滑动动画效果

#### 滚动条滑动效果核心步骤:

1. 默认的bootstrap滚动监控弄上
2. 直接上js脚本，抓取导航栏的a标签，即可，脚本如下

<script>

// 创建滚动效果

$(".navbar a").on( // 抓取跳转的a标签,要与目标位置对应

'click',

function(event){

if(this.hash !== ""){

event.preventDefault();

const hash = this.hash;

$('html, body').animate({

scrollTop: $(hash).offset().top

},

800, // 动画执行时间

function(){

window.location.hash = hash;

}

);

}

});

</script>

例:

<body data-spy="scroll" data-target=".navbar" data-offset="50" >

<!--nav-BGN-->

<nav class="navbar bg-dark navbar-dark navbar-expand-sm fixed-top">

<div class="container">

<a href="#" class="navbar-brand">

LOGO

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#main-nav" >

<span class="navbar-toggler-icon" ></span>

</button>

<div class="navbar-collapse collapse" id="main-nav" >

<ul class="navbar-nav ml-auto ">

<li class="nav-items">

<a href="#home-section" class="nav-link">主页</a>

</li>

<li class="nav-items">

<a href="#explore-head-section" class="nav-link">探索</a>

</li>

<li class="nav-items">

<a href="#create-head-section" class="nav-link">制造</a>

</li>

<li class="nav-items">

<a href="#share-head-section" class="nav-link">分享</a>

</li>

</ul>

</div>

</div>

</nav>

<!--nav-END-->

<!—main\_body-BGN -->

<div id=” home-section”></div>

<div id=” explore-head-section”></div>

<div id=” create-head-section”></div>

<div id=” share-head-section”></div>

<!—main\_body-END -->

<script>

// 创建滚动效果

$(".navbar a").on( // 抓取跳转的a标签,要与目标位置对应

'click',

function(event){

if(this.hash !== ""){

event.preventDefault();

const hash = this.hash;

$('html, body').animate({

scrollTop: $(hash).offset().top

},

800, // 动画执行时间

function(){

window.location.hash = hash;

}

);

}

});

</script>

### HTML5-使背景图片静止不动, 锁定背景图片fiexd, 控制背景颜色透明度

background-attachment: fixed; /\*\* 使用背景图静止不动 \*\*/

background-color: rgba(0,0,0,0.7); /\*\* background-color:rgba(rgb颜色,透明度);可以控制背景颜色的透明度 \*\*/

### bootstrap4-控制不同分辨率显示或者不显示

核心:

1. D-none, d-lg-block
2. D-none: 相当于display:none; 不被浏览器渲染显示的标签
3. D-lg-block: 在屏幕分辨率为大于lg时为display:block;( d-sm/md/lg/xl-block都可以)

例:

<div container>

<div class=”row”>

<div class=”col-md-8 d-none d-md-block”></div>

<div class=”col-md-4”>

</div>

</div>

## Bootstrap4-响应式布局核心

那么我们以实战为演示：

<div id=”home-section”> <!— 模块化布局建设ID -->

<div class=”container”> <!— 必备，为写row打基础 -->

<div class=”row”> <!—必备, 为写col打基础 -->

<div class=”col-md-6”></div> <!— 当分辨率大于md时为占据6/12 -->

<div class=”col-md-6”></div>

</div>

</div>

</div>

## Jquery-部分使用

#### Jquery-cdn:

1. <script src="http://code.jquery.com/jquery-3.4.1.min.js" integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFlBw8HfCJo=" crossorigin="anonymous"></script>

#### Jquery-显示当前时间, 年

<script>

// 获取时间,使用时" <span id="year"></span> "对应id是核心

$('#year').text(new Date().getFullYear());

</script>

#### Jquery-创建滚动条动画滑动效果

// 创建滚动效果

$(".navbar a").on( // 抓取跳转的a标签,要与目标位置对应

'click',

function(event){

if(this.hash !== ""){

event.preventDefault();

const hash = this.hash;

$('html, body').animate({

scrollTop: $(hash).offset().top

},

800, // 动画执行时间

function(){

window.location.hash = hash;

}

);

}

});

</script>

## Bootstrap-注意引入图片要加入响应式属性

<img src=”1.jpg” class=”img-fluid” >

#### Bootstrap-display的快捷使用

d-inline 相当于 display:inline;

d-block 相当于 display:block;

d-none 相当于 display:none;

### bootsrap-全部字体属性

#### BOOTSRAP-改变字体颜色

Text-danger: 即可改变字体颜色

#### BOOTSRAP-改变字体对齐方式

0. text-right/text-center/text-left - 右对齐,剧中,左对齐

1. text-justify - 文本自动换行( 默认开启 )

2. text-nowrap - 文本不换行

#### BOOTSRAP-改变字体大小写

**0. text-lowercase - 全部小写**

**1. text-uppercase - 全部大写**

**2. text-capitalize - 开头字母大写**

## **CDN-ICON小图标**

### 加入代码

<!-- cdn\_icon -->

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.6.3/css/all.css" integrity="sha384-UHRtZLI+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ8lWUE00s/" crossorigin="anonymous">

#### Icon实列搜索查看

<https://fontawesome.com/icons/>

## Bootstrap-自适应图片原理

Img-fluid{

Max-width:100%;

Height:auto;

}

# 2019.7.15

### Bootstrap-变量修改

如修改颜色变量，直接在\_variables.scss中修改对应变量值，即达到bootstrap全局修改

注意: scss文件名称开头为”下划线”则scss不会编译此文件

### Bootstrap-不同分辨率内外边距设定

<!-- mb-5,mb-lg-0与d-none与d-lg-block配合一个道理 -->

Mb-lg-0：只要屏幕分辨率大于lg则外边距为0，即可达到不同分辨率所设定外边距/内边距不同

### CSS3-margin负数用法

margin-top:-50px; /\*\* margin竟然哈可以使用负数，竟然有惊奇的效果，可以越父类容器的界 \*\*/

### CSS3-指定分辨率执行属性

通常@media( max-width:分辨率 ){ } 用在移动端屏幕分辨率上

/\*\* 移动端样式: 当屏幕分辨率<768px时生效 \*\*/

@media( max-width:768px ){

/\*\* 移动端时,加外边距 \*\*/

.mb-resp{

margin-bottom: 1.5rem;

}

}

### CSS3-文本不换行，以及溢出文本变为”…”( 省略符 )

1. 文本换行/不换行核心:
   1. word-break:break-all; // 自动换行
   2. word-break:keep-all; // 不换行
   3. white-space:nowrap; // 规定段落中的文本不进行换行

.text-oneLine{

display:block; /\*内联对象需加\*/

word-break:keep-all; /\* 不换行 \*/

white-space:nowrap; /\* 不换行 \*/

overflow:hidden; /\* 内容超出宽度时隐藏超出部分的内容 \*/

text-overflow:ellipsis; /\* 当对象内文本溢出时显示省略标记(...) ；需与overflow:hidden;一起使用。\*/

}

# 2019.7.18

## BT4猜想印证

### bootstrap-在同一个row中可以方多行col吗？( 超过12 )以达到自适应效果

* 1. 能超出12,超出12后会自动换行，且换行后依然为12格
  2. 核心抽象：col为一个面，md也为一个面…等其它也如此
     1. col-num与col-num 在任何分辨率不会换行
     2. col-md-num: 达到md分辨率将换行，其它类似

### bootstarp-轮播图的备注内容可以先嵌入container -> carousel-caption

目的: 把carousel-item嵌入container在嵌入carousel-caption也是可以的，能更加有校的限制显示内容

例如:

<div class="carousel-item active banner\_0">

                        <div class="container d-none d-sm-block "> *<!-- 把carousel-item嵌入container在嵌入carousel-caption也是可以的，能更加有校的限制显示内容 -->*

                            <div class="carousel-caption text-right ">

                                <h2 class="display-3" >

                                    Banner\_One

                                </h2>

                               </div>

                        </div>

                    </div>

### Bootstrap-banner轮播图设置时间

// 控制banner特效( carousel动态效果 )

$('.carousel').carousel({

interval:3500, // 3.5/s一换图

pause:'hover' // 鼠标悬停时,停止动画

});

### Bootstrap-如何控制标签高度

min-height:200px; /\*\* 为兼容响应式,没用特殊要求,不能固定高度 \*\*/

## Bootstrap-modal弹窗获取youtube视频，并自动加载（ 共3步 ）

#### 按钮:

<!-- data-video="https://www.youtube.com/embed/视频编号，即可获取对应视频数据 -->

<a href="#" class="video" data-toggle="modal" data-target="#test1" data-video="https://www.youtube.com/embed/7QskmAsU9kc" >

#### Modal-body中:

<!-- 关于iframe -->

<!--

a) frameborder="0": 视频边框为0

b) allowfullscreen: 允许全屏

-->

<iframe src="" frameborder="0" width="100%" height="350px" allowfullscreen ></iframe>

#### 设置Jquery:

// 目的是自动加载视频( 只能使用在上方所演示的固定模式中 )

$(function () {

// 抓取目标class=”.video”

$(".video").click(function () {

var theModal = $(this).data("target"),

videoSRC = $(this).attr("data-video"),

videoSRCauto = videoSRC + "?modestbranding=1&rel=0&controls=0&showinfo=0&html5=1&autoplay=1"; // 自动播放核心代码

$(theModal + ' iframe').attr('src', videoSRCauto);

$(theModal + ' button.close').click(function () {

$(theModal + ' iframe').attr('src', videoSRC);

});

});

});

## css3-浏览器取消x滚动轴

overflow-x: hidden; /\*\* 取消显示x滚动轴 \*\*/

## html5-跨站跳转到指定页面指定ID位置（ 非常增加用户体验 ）

核心: href="指定页面/#ID" 即可跳转 [ 单击按钮 -> 跳转指定页面 -> 滑动到指定ID ]

用处: 增加用户体验，比如产品在切换页面时，例如“鸿伟酒业”产品跳转,但是ID号要统一

例: href="/tel.html/#tel" ( 跨页面跳转 )

### css3-自适应背景图

核心: background-size:cover; /\*\* 二者竟然可以配合使用,从而达到完全覆盖显示效果，且不变形 \*\*/

background:url(../images/image1.jpg) no-repeat top center ;

background-size: cover;

### css3-锁定背景图片( 使背景图有fixed效果 )

#ready{

background: url(../images/lights.jpg) center center no-repeat ;

background-attachment: fixed; /\*\* 锁定背景图片 \*\*/

}

### Css3-精确改变背景图片位置

background-position: x y ; /\* 精确改变背景图片的相对位置，background-position: x y ; \*/

## Bootstrap-灯箱效果

1. 步骤:
2. 引入CDN
3. 去官网：<http://ashleydw.github.io/lightbox/> （ 在官网 ）
4. CDN链接,注意引入位置：

<!-- 灯笼弹窗效果-与Bt4搭配,放置头部bt4之下-->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/ekko-lightbox/5.3.0/ekko-lightbox.css" integrity="sha256-HAaDW5o2+LelybUhfuk0Zh2Vdk8Y2W2UeKmbaXhalfA=" crossorigin="anonymous" />

<!-- 灯笼弹窗效果js需文件-与Bt4搭配, 放置尾部bt4之下-->

<script src="https://cdnjs.cloudflare.com/ajax/libs/ekko-lightbox/5.3.0/ekko-lightbox.js" integrity="sha256-jGAkJO3hvqIDc4nIY1sfh/FPbV+UK+1N+xJJg6zzr7A=" crossorigin="anonymous"></script>

1. 设置jquery

// 灯箱效果

$(document).on('click', '[data-toggle="lightbox"]', function (event) { // 抓取属性执行效果

event.preventDefault();

$(this).ekkoLightbox();

});：

#### <!-- 灯箱弹窗效果,需要上方cdn.css与底部cdn.js+jquery配合，在配合bootstarp -->

<!--

data-toggle="lightbox" // 开启灯箱效果( 核心 )

data-gallery="img-gallery" // 保证灯箱效果下能切换图片

data-width/height="500" // 控制弹窗大小

data-title=” 图片标题 ” // 单击图片打开后显示标题

href=”” 灯箱效果会将链接的内容加载出来，并且显示，可以是本地图片或者远程链接

-->

<a href="https://source.unsplash.com/random/1200x650" data-toggle="lightbox" data-gallery="img-gallery" data-width="1200" data-height="650" >

<img src="https://source.unsplash.com/random/567x567" alt="" class="img-fluid" >

</a>

### <!—灯箱多图按钮可切换重要属性 - 实战 -->

1. data-gallery="example-gallery" 目标值属性相同,则可以在灯效效果开启时直接进行按钮切换查看图片
2. 链接分析: <https://unsplash.it/1200/768.jpg?image=251>
   1. 指定图片编号索引,此意能索引相同图片，不同大小
3. data-max-width (or data-max-height) 限制灯箱显示图片大小
4. 别忘记1个css，1个js，1个jq

<div class="row">

<a href="https://unsplash.it/1200/768.jpg?image=251" data-toggle="lightbox" data-gallery="example-gallery" class="col-sm-4">

<img src="https://unsplash.it/600.jpg?image=251" class="img-fluid">

</a>

<a href="https://unsplash.it/1200/768.jpg?image=252" data-toggle="lightbox" data-gallery="example-gallery" class="col-sm-4">

<img src="https://unsplash.it/600.jpg?image=252" class="img-fluid">

</a>

<a href="https://unsplash.it/1200/768.jpg?image=253" data-toggle="lightbox" data-gallery="example-gallery" class="col-sm-4">

<img src="https://unsplash.it/600.jpg?image=253" class="img-fluid">

</a>

</div>

#### 必备jq：

$(document).on('click', '[data-toggle="lightbox"]', function(event) {

event.preventDefault();

$(this).ekkoLightbox();

});

# 2019.7.21

### 滑动轮播模块

效果如下：按下按钮可以来回切换



#### 滑动轮播必备插件

<!-- 滑动轮播模块-共二个css一个js -->放在头部

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/slick-carousel/1.9.0/slick.css" />

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/slick-carousel/1.9.0/slick-theme.css" />

<!-- 滑动轮播模块 -->放在尾部

<script src="https://cdnjs.cloudflare.com/ajax/libs/slick-carousel/1.9.0/slick.js" integrity="sha256-zUQGihTEkA4nkrgfbbAM1f3pxvnWiznBND+TuJoUv3M=" crossorigin="anonymous"></script>

// 滑动轮播模块jquery必备,滑动轮播必备Jquery: 放在尾部的尾部，一定在引入的js后面

$('.slider').slick({

infinite: true, // 开启滑动

slideToShow: 1, // 初试位置为1

sliderToScrool: 1 // 滑动数为1

});

#### 滑动模块使用方式

<!-- 滑动轮播图特效 -->

<!--

a) 添加class=".slider"后在js抓取.slider( 别忘记jquery抓取 )

b) 满足架构:

<div class="slider" >

<div>内容1<div>

<div>内容2<div>

</div>

// 抓取标签

<script>

$('.slider').slick({

infinite: true, // 开启滑动

slideToShow: 1, // 初试位置为1

sliderToScrool: 1 // 滑动数为1

});

</script>

c) 此效果简单明了，重点在于jquery

-->

### 从unsplash调用随机图片引用方式

<!-- 随机图片CDN解析:

https://source.unsplash.com/random/宽度x高度/?关键词(以控制图片类型)

-->

<img src="https://source.unsplash.com/random/700x700/?technology" >

# 2019.7.21

### Bootstrap-一条x水平线

<hr/>

### <!-- 此次是card套装与collapse配合的

发现给card赋予颜色后，其内容自动调整合适的颜色，

在这里card-header通过collase控制card-body的父类容器#fq1...等 -->

实列:

<div id=”fq-left”>

<div class="card bg-dark text-light text-left">

<div class="card-header">

<!-- #fq\* -->

<a href="#fq1" class="d-block text-light" data-toggle="collapse" data-parent="#fq-left" >

问题1

</a>

</div>

<!-- #fq\* -->

<div id="fq1" class="collapse" >

<div class="card-body">

<p>

content

</p>

</div>

</div>

</div>

### Bootstrap-当内容都为card时可以用card-columns作为父类容器,使card自动排序

**card-columns配合card具有一定的流式布局效果**

// 此次布局有点特殊,只用一个col

<div class=”col”>

<div class=”card-colums”> //让card-columns( 父类 )他来处理card( 子类 )的关系，有他在card将自动排序

<div class=”card”></div>

<div class=”card”></div>

<div class=”card”></div>

</div>

</div>

### Bootstrap-引用字体格式

<small> 会使字体变小 </small>

<cite> 通常用于定义作品标题 <cite>

带颜色的标签为blockquote固定格式

<blockquote class="blockquote">

<p>

Content1111

</p>

<footer class="blockquote-footer">

<small class="text-muted">

部分内容参考文献于

<cite> <!-- cite标签用于，定义作品的标题 -->

菜鸟驿站

</cite>

</small>

</footer>

</blockquote>

## Bootstrap-row可以进行多此嵌套，起本质为在100%宽度下，创建12个间距表格

<div class="row"> <!-- row中可以在此嵌套row，row本质为在100%宽度下创建12个表格间距 -->

<div class=”col-md-12”></div>

<div class="row">

<div class=”col-md-6”></div>

<div class=”col-md-6”></div>

</div>

</div>

# 2019.7.23

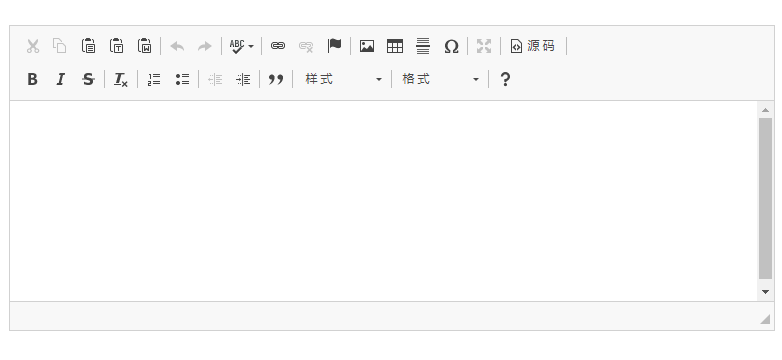
## Cdn-ckeditor – 超级文本编辑器

Cdn: <script src="https://cdn.staticfile.org/ckeditor/4.12.1/ckeditor.js"></script>

实列:

1. 设置jquery: <script> CKEDITOR.replace( 'editor1' ); </script> // editor1 抓取目标
2. 使用: <textarea class=”editor1”><textarea>

效果:



## Bootstrap-响应式表格注意事项:

<!--

表格响应式注意事项:

0. 表格不能嵌套在card-body中

1. .table嵌套在.table-responsive中开启响应式

2. 表格具有响应式,前提条件之一:

th/td的内容要设置为"不换行"( white-space:nowrap;)否则,无法使表格在手机端正常显示

3. table 为内联元素，所以在表格排序不一时，width:100%;最好

-->

实列:

<style>

Table{  
 th,td{

White-space:nowrap; // 不换行,响应式表格必备属性

}

}

</style>

<div class="table-responsive"> // 表格嵌套其中开启响应式

<table class=”table” >

<thead>

<th>

</thead>

<tbody>

<tr>

<td>

</tr>

</tbody>

</table>

</div>

# 2019.7.28

## 搜索引擎原理推测:

自动生成域名访问抓取网站标题,与关键词，自动收录

## HTML5-<fieldset>指向表单, 相当于在同一个form中的内容

<form id=”setting\_save” ><form>

<fieldset form="setting\_save" >

<legend> <!-- 与表单搭配的标题 -->

用户注册

</legend>

<div class="radio">

<label for="">

<input type="radio" value="yes" name="choose" id=""> Yes

</label>

</div>

</fieldset>

## <!—bootstrap-btn-block按钮变为块级元素 -->

<input type="submit" value="登陆" class="btn btn-primary btn-block">

## Bootstrap-flex布局正确用法

<!-- bootstrap- d-flex与flex-row配合使用,即可正常使用flex布局 -->

1. 重点在于，d-flex, flex-row 二者之间的配合
2. 默认嵌套之内的子标签不会换行，且可以在flex-row的淫威之下使用flex属性为所欲为
3. 甚至，在flex-row中，width:25%;于padding:;二者之间不会出现冲突,
4. 如图每个width:25%; 且具有p-4属性，但是二者并没用任何冲突



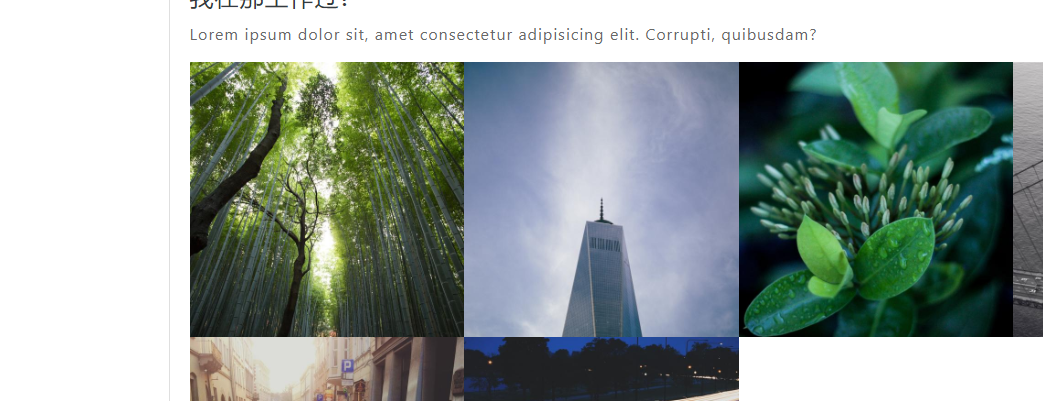
<div class="d-flex flex-row align-items-stretch text-center">

<div></div>

<div></div>

<div>

## <!-- bootstrap- no-gutters让标签的子类,左右无外边距,无缝结合 -->

1. 因为默认col之间是有间距的
2. No-gutters可以取消被嵌套标签的间距，从而达到无缝结合
3. 

<div class="row no-gutters">

<div class="col-md-3">

<!-- 灯箱效果 -->

<a href="https://unsplash.it/1440/800.jpg?image=666" data-toggle="lightbox" >

<img src="https://unsplash.it/600.jpg?image=666" class="img-fluid" alt="">

</a>

</div>

<div class="col-md-3">

<!-- 灯箱效果 -->

<a href="https://unsplash.it/1440/800.jpg?image=666" data-toggle="lightbox" >

<img src="https://unsplash.it/600.jpg?image=666" class="img-fluid" alt="">

</a>

</div>

</div>

Jquery-代替collapse-parinet

# 2019.8.2

### HTML5-关于对按钮禁用disabled属性的补充

// disabled = “disabled”; 禁用按钮

// disabled = “false”; 按钮恢复正常

<button class="btn-new" disabled >

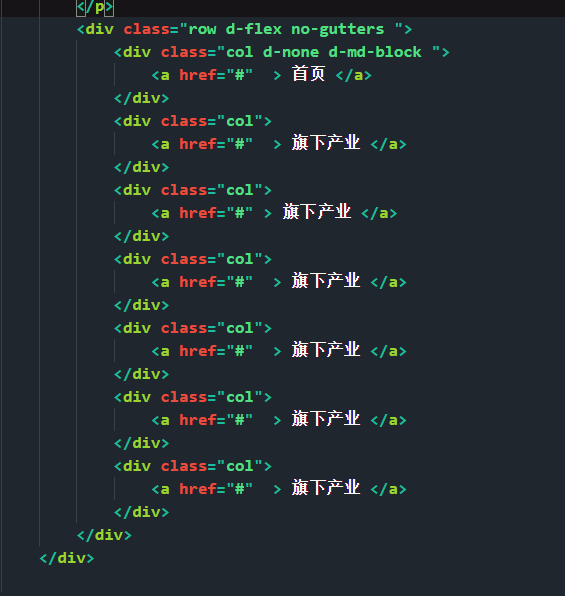
<i class="ion-ios-plus-outline"></i>New game

</button>

# 2019.8.21

## Bootstarp4-多标签自动换行

### 如下图写法:

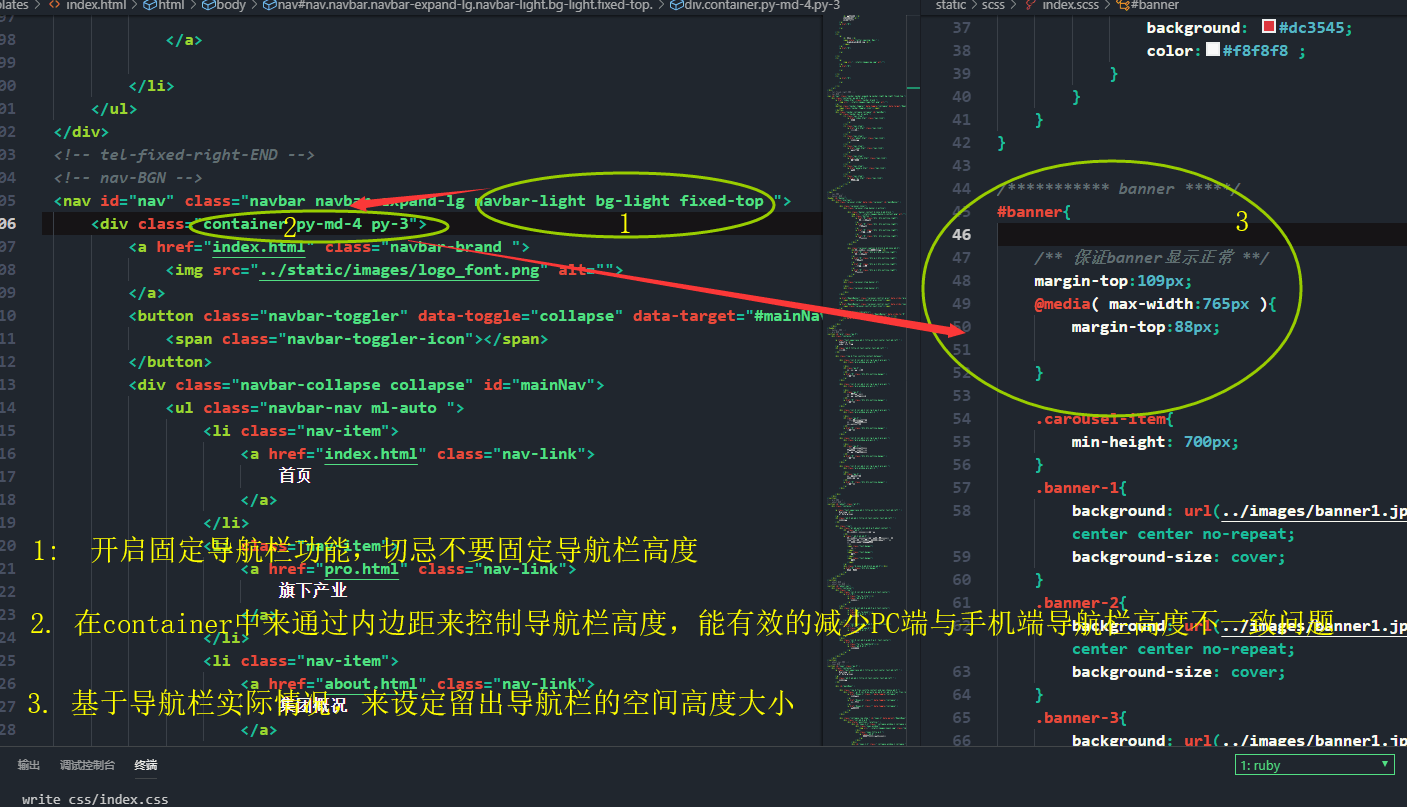


### 自动换行分析

1. 父类开启flex模式: .row .d-flex
   1. 不能有固定高度
2. 子类必须要有固定宽度:
   1. 写入.col
   2. 标签且有固定宽度如: width:200px;等
3. 满足以上条件即可开启自动根据大小在不同屏幕之间换行

## Bootstap4正确开启固定导航栏

### 如图:



### 固定导航栏分析

#### 开启固定导航栏功能

* 1. 写入 : .fixed-top/bottom,字体颜色( .navbar-light ),背景颜色( .bg-light )
  2. 切忌: 不能固定导航栏高度

#### 控制导航栏高度

* 1. 通过控制导航栏中.container的” 内间距 ”达到控制导航栏高度问题
  2. 优点: 此方法能有效的统一PC端 和 M端高度大小问题

#### 给导航栏留出位置

* 1. 根据实际情况留出恰当的外边距，注意PC/M端的问题
  2. 如果按照第二个方法进化，能更加方便操控外边距问题

# 2019.8.31

## 利用transform控制标签剧中对齐( 新的剧中姿势 )

### 原理解析:

0. top:50%; left:50%; 保证图片的右上角在剧中位置

1. transform: translate( x,y )本身就为控制图片相对位置的函数

### 实列:

HTML:

<div>

<div class=”test”>1</div>

</div>

CSS:

.test{

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

}

# 2019.10.9

## Css3属性选择器 [ 属性=”值” ]

1. 抓取标签属性及属性值

<style>

#### // 0. 只抓取所有标签含有test值的标签

[class="test"] {

background: #ffff00;

}

#### // 1. 抓取所有标签类中含有test的标签 ，包括class=” xxx\_test ”

[class\*="test"] {

background: #ffff00;

}

#### // 2. 抓取含有对应属性的标签

[data-itemid=’123’]{

// 及可抓取对应标签

}

</style>

<div class="test" data-itemid = ‘123’ >The third div element.</div>

<p class="test" data-itemid = ‘222’>This is some text in a paragraph.</p>