

OHModel UI 框架使用文档

OHModel UI 框架使用文档

1. 开始使用
2. 使用CSS样式
 - 2.1 整体布局类
 - 2.2 图标类 glyphicon
 - 2.3 辅助样式类
 - 2.3.1 浮动
 - 2.3.2 内联块级
 - 2.3.3 水平分割线
 - 2.3.4 字体颜色
 - 2.3.5 背景颜色
 - 2.3.6 小三角符号
 - 2.3.7 居中块级
 - 2.3.8 显示/隐藏
 - 2.3.9 徽章
 - 2.3.10 标记文本
 - 2.3.11 标题
 - 2.3.12 改变大小写
 - 2.4 排版布局 g
 - 2.4.1 logo样式
 - 2.4.2 调整布局
 - 2.4.3 顶部导航栏
 - 2.4.4 侧边导航
 - 2.4.5 页面主体部分
 - 2.6 按钮样式 u
 - 2.7 表单样式
 - 2.8 表格样式
 - 2.9 tab切换

结语

1. 开始使用

- 头部必须要有 `<!DOCTYPE html>`，不然会出现意想不到的bug
- 引用 `CSS` 样式表

```
<link rel="stylesheet" href="../css/OHModel.css">
```

- 应用 `JavaScript` 文件，顺序不要错

```
<script type="text/javascript" src="../js/mTools.js"></script>
<script type="text/javascript" src="../plugIn/jquery-1.8.3.min.js"></script>
<script type="text/javascript" src="../js/mToolsJQ.js"></script>
```

2. 使用CSS样式

目前css分为 整体布局类、 图标类、 辅助类样式、 排版布局样式、 按钮样式、 表单样式、 表格样式、 `tab`切换 几个板块。

2.1 整体布局类

这部分主要目的是清楚浏览器自带的样式，不用在意。

2.2 图标类 glyphicon

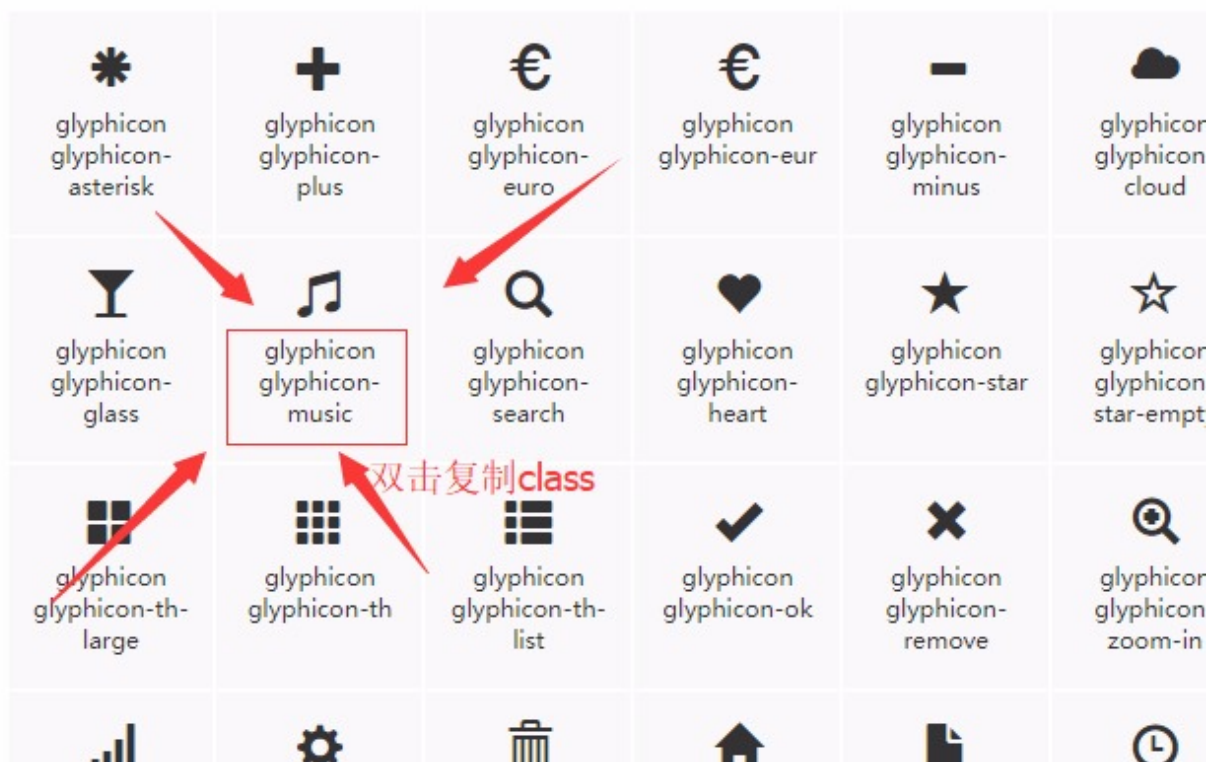
框架使用了bootstrap内部的图标，基本上可以满足开发需求。

使用方法介绍：

1. 找到自己想要的图标，打开[图标展示表](#),寻找想要使用的图标
2. 复制class全称（连续点击3次即可全部选中）
3. 新建 `i` 标签，设置class

事例：

```
<i class="glyphicon glyphicon-music"></i>
```



2.3 辅助样式类

一般辅助样式包括一些定义好的 浮动、内联块级、水平分割线、字体颜色、背景色、小三角、居中块级、显示/隐藏、徽章等。

2.3.1 浮动

使用浮动对块级元素进行布局控制。使用浮动以后要给父级元素加上清楚浮动的标志。

- `f-fl` 向左浮动
- `f-fr` 向右浮动
- `clear-fix` 清楚浮动

事例：

```
<!--浮动模块-->
<div class="clear-fix">
  <div class="f-fl">左边模块</div>
  <div class="f-fr">右边模块</div>
</div>
```

2.3.2 内联块级

如果需要把一个 `div` 或者 `span` 成 `inline-block`。只需要加上 `f-ib` 的class即可

2.3.3 水平分割线

使用class `f-horizon` 即可

2.3.4 字体颜色

使用对应的字体颜色class，更改字体颜色

```
<span class="fc-muted">这是一段文字 muted 柔和 </span>

<span class="fc-primary">这是一段文字 primary 初始 </span>

<span class="fc-success">这是一段文字 success 成功 </span>

<span class="fc-info">这是一段文字 info 消息 </span>

<span class="fc-warning">这是一段文字 warning 警告 </span>

<span class="fc-danger">这是一段文字 danger 危险 </span>
```

这是一段文字 muted 柔和
这是一段文字 primary 初始
这是一段文字 success 成功
这是一段文字 info 消息
这是一段文字 warning 警告
这是一段文字 danger 危险

2.3.5 背景颜色

使用对应的背景颜色class,即可更改背景色

```
<div class="bg-primary">背景色展示 bg-primary 初始的</div>
<div class="bg-success">背景色展示 bg-success 成功</div>
<div class="bg-info">背景色展示 bg-info 信息</div>
<div class="bg-warning">背景色展示 bg-warning 警告</div>
<div class="bg-danger">背景色展示 bg-danger 危险</div>
```

背景色展示 bg-primary 初始的

背景色展示 bg-success 成功

背景色展示 bg-info 信息

背景色展示 bg-warning 警告

背景色展示 bg-danger 危险

2.3.6 小三角符号

代表需要展开的菜单，class为 `caret`

```
<i class="caret"></i>
```

2.3.7 居中块级

需要水平居中的块级元素，加上class `center-block`

```
<div class="center-block"></div>
```

2.3.8 显示/隐藏

需要隐藏的元素加class `hide`，需要显示的元素加class `show`

2.3.9 徽章

徽章的作用在于显示数量，次数。使用方法，新建一个 `span` 标签，class 设置为 `badge`，通常用来表示消息的数量。

```
<div class="fc-primary f-ib">徽章 <span class="badge">4</span> </div>
```

徽章 4

按钮中的 `badge` 有一点不同

```
<div class="u-btn f-ib">徽章 <span class="badge">4</span> </div>
```

2.3.10 标记文本

如果你想标记某些文本，可以使用 `mark` 标签

```
<mark>这是一段标记文本</mark>
```

这是一段标记文本

2.3.11 标题

标题可以使用 `h1~h6` 标签，也可以使用class 从h1~h6，字号由大到小。

```
<h1>这是h1</h1>
<h2>这是h2</h2>
<h3>这是h3</h3>
<h4>这是h4</h4>
<h5>这是h5</h5>
<h6>这是h6</h6>
```

这是一段标记文本

2.3.12 改变大小写

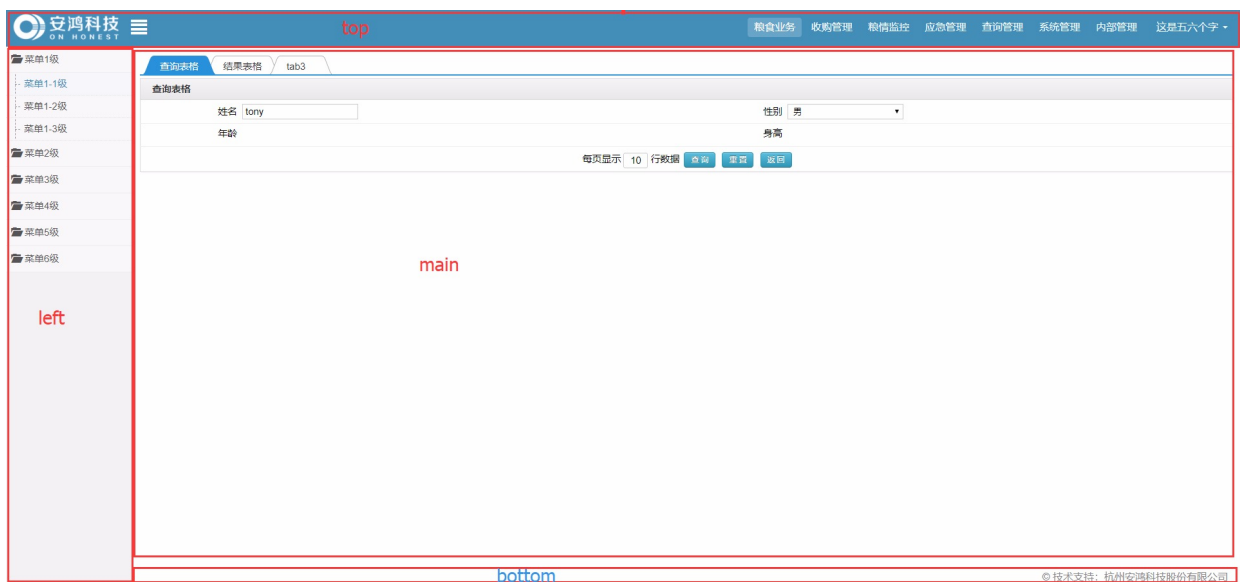
英文字母可以通过css来改变大小写，可以通过以下class来实现

- `text-lowercase` 切换为小写
- `text-uppercase` 切换为大写
- `text-capitalize` 首字母大写

2.4 排版布局 g

布局模块 g（grid 布局），布局模块控制着系统的布局

- `g-top` 顶部布局模块，用户，导航，logo展示区
- `g-left` 左部布局模块，侧边导航菜单
- `g-main` 主体布局模块，页面主体展示部分
- `g-bottom` 底部布局模块，系统，版权介绍部分



2.4.1 logo样式

`m-logo` 标签，包裹着logo元素。不过 `m-logo` 需要放在 `g-top` 元素里，才能实现样式。

```
<a class="m-logo f-fl"></a>
```

2.4.2 调整布局

`u-adjust` 代表调整布局的按钮，配合 `js` 使用，实现功能。

```
<!-- html -->
<div class="u-adjust f-fl" id="adjust"><i class="glyphicon glyphicon-align-justify"></i></div>
//js
adjust();
```

2.4.3 顶部导航栏

顶部导航 `HTML` 只需要调一下代码

```
<!-- 顶部导航 -->
<div class="m-nav f-fr" id="nav"></div>
```

另外需要调用 `js`，生成效果

```
//addTopNav(id,接口的url)
addTopNav('nav','../json/topNav.json');
```

事例接口形式,其中 `name` 代表导航的名称， `url` 代表侧边导航的数据接口地址。

```
[
  {"name": "粮食业务", "url": "../json/slideNav2.json"},
  {"name": "收购管理", "url": "../json/slideNav2.json"},
  {"name": "粮情监控", "url": "../json/slideNav2.json"},
  {"name": "应急管理", "url": "../json/slideNav2.json"},
  {"name": "查询管理", "url": "../json/slideNav2.json"},
  {"name": "系统管理", "url": "../json/slideNav2.json"},
  {"name": "内部管理", "url": "../json/slideNav2.json"}
]
```

2.4.4 侧边导航

侧边导航栏 **HTML**，只需要引用以下代码

```
<!-- 侧边栏 -->
<div class="g-left" id="slide_frame"></div>
```

js 代码部分

```
//addSlideNav(id,数据接口url)
addSlideNav('slide_frame','../json/slideNav.json');
```

接口事例，其中 **name** 表示导航名称，**href** 代表页面地址，**sub** 表示子菜单

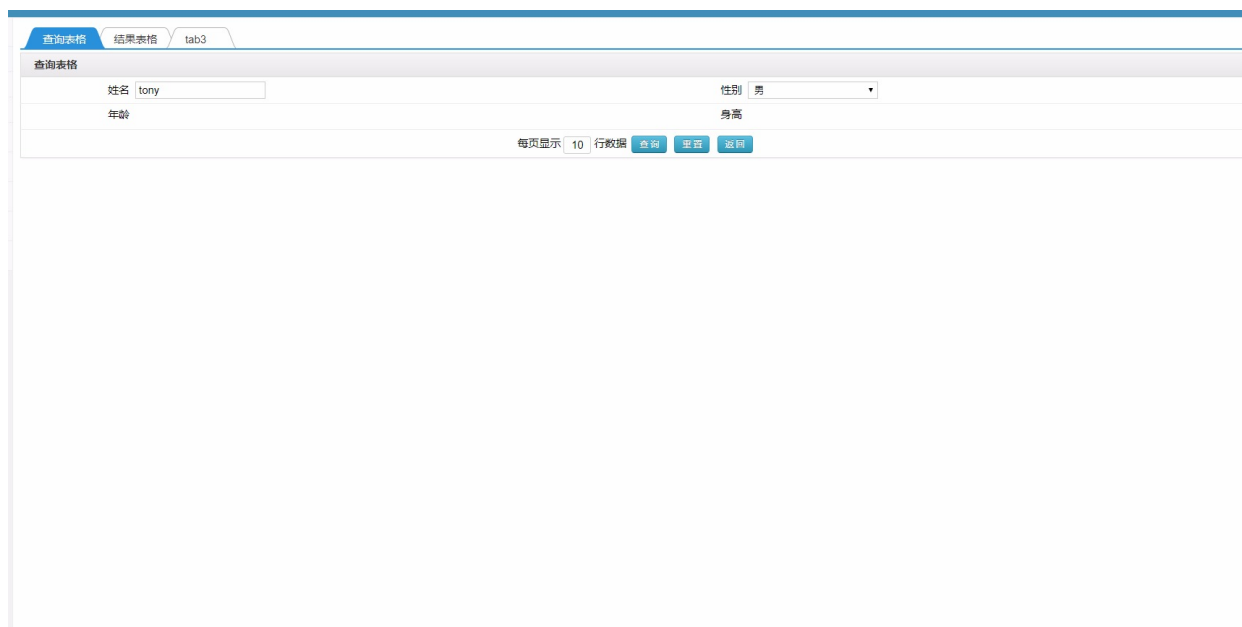
```
[
  {"name": "菜单1级", "sub": [
    {"name": "菜单1-1级", "href": "demo1.html"},
    {"name": "菜单1-2级", "href": "www.baidu.com"},
    {"name": "菜单1-3级", "href": "www.baidu.com"}
  ]},
  {"name": "菜单2级", "href": "www.baidu.com"},
  {"name": "菜单3级", "sub": [
    {"name": "菜单3-1级", "href": "www.baidu.com"},
    {"name": "菜单3-2级", "href": "www.baidu.com"},
    {"name": "菜单3-3级", "href": "www.baidu.com"}
  ]},
  {"name": "菜单4级", "href": "www.baidu.com"},
  {"name": "菜单5级", "href": "www.baidu.com"},
  {"name": "菜单6级", "href": "www.baidu.com"}
]
```




2.4.5 页面主体部分

页面主体使用 `iframe` 元素存放页面，具体使用方法参照以下代码

```
<div class="g-main" id="main_frame">  
  <iframe name="main_frame" frameborder="0" width="100%" height="100%">  
</iframe>  
</div>
```



2.6 按钮样式 u

对于按钮元素，需要加上一个class `u-btn`，其中u 是unit 缩写 代表 元件。使用

```
<button class="u-btn">按钮1</button>
<input type="button" class="u-btn" value="按钮2">
```

按钮 1

按钮 2

2.7 表单样式

对于input,select等表单元素，需要加上class `u-ipt`

```
<input type="text" class="ipt">
<select name="age">
  <option value="20">20</option>
  <option value="21">21</option>
  <option value="22">22</option>
</select>
```

2.8 表格样式

表格样式需要给 `table` 标签添加一个 `m-tb` 的默认的表格样式的class。内部的样式有：

- `search` 定义在 `table` 标签上与 `m-tb` 共同使用，代表搜索表格
 - `result` 定义在 `table` 标签上与 `m-tb` 共同使用，代表结果表格
 - `tb-title` 定义在 `tr` 标签上，代表表格标题
 - `tb-head` 定义在 `tr` 标签上，代表表格表头
 - `tb-content` 定义在 `tr` 标签上，代表表格内容行
 - `odd` 与 `tb-content` 共同使用,用于设置奇偶不同的背景效果，如果 `table` 使用了 `result`，无需设置
 - `tb-bottom` 定义在 `tr` 标签上，代表表格底部
 - `tb-field-head` 定义在 `td` 标签上，用于搜索的字段名
 - `tb-field-content` 定义在 `td` 标签上，用于搜索的字段值
 - `page-num` 定义在bottom的页码跳转 `input` 上，用于跳转页码
- 具体事例

1. 普通表格

```
<table class="m-tb">
  <tr class="tb-title">
    <td colspan="5">
      默认表格
```

```

        </td>
    </tr>
    <tr class="tb-head">
        <td>id</td>
        <td>name</td>
        <td>sex</td>
        <td>age</td>
        <td>height</td>
    </tr>
    <tr class="tb-content">
        <td>1</td>
        <td>jhon</td>
        <td>male</td>
        <td>20</td>
        <td><a href="">操作</a></td>
    </tr>
    <tr class="tb-content">
        <td>2</td>
        <td>tom</td>
        <td>male</td>
        <td>22</td>
        <td><a href="">操作</a></td>
    </tr>
    <tr class="tb-content">
        <td>3</td>
        <td>mey</td>
        <td>female</td>
        <td>21</td>
        <td><a href="">操作</a></td>
    </tr>
    <tr class="tb-bottom">
        <td colspan="5">
            <span></span>
            <button class="u-btn">首页</button>
            <button class="u-btn" disabled>上页</button>
            <button class="u-btn">下页</button>
            <button class="u-btn">尾页</button>
            <input type="text" class="page-num" value="1" />
            <button class="u-btn">GO</button>
        </td>
    </tr>
</table>

```

默认表格				
id	name	sex	age	height
1	jhon	male	20	操作
2	tom	male	22	操作
3	mey	female	21	操作

[首页](#)
[上页](#)
[下页](#)
[尾页](#)

[GO](#)

2. 结果表格 result

```

<table class="m-tb result">
  <tr class="tb-title">
    <td colspan="5">默认表格</td>
  </tr>
  <tr class="tb-head">
    <td>id</td>
    <td>name</td>
    <td>sex</td>
    <td>age</td>
    <td>height</td>
  </tr>
  <tr class="tb-content">
    <td>1</td>
    <td>jhon</td>
    <td>male</td>
    <td>20</td>
    <td><a href="">操作</a></td>
  </tr>
  <tr class="tb-content">
    <td>2</td>
    <td>tom</td>
    <td>male</td>
    <td>22</td>
    <td><a href="">操作</a></td>
  </tr>
  <tr class="tb-content">
    <td>3</td>
    <td>mey</td>
    <td>female</td>
    <td>21</td>
    <td><a href="">操作</a></td>
  </tr>
  <tr class="tb-bottom">
    <td colspan="5">
      <span></span>
      <button class="u-btn">首页</button>
      <button class="u-btn" disabled>上页</button>
      <button class="u-btn">下页</button>
      <button class="u-btn">尾页</button>
      <input type="text" class="page-num" value="1" />
      <button class="u-btn">GO</button>
    </td>
  </tr>
</table>

```

结果表格

id	name	sex	age	height
1	jhon	male	20	操作
2	tom	male	22	操作
3	mey	female	21	操作

首页
上页
下页
尾页
1
GO

3. 搜索表格 search

```
<table class="m-tb search">
  <tr class="tb-title">
    <td colspan="4">查询表格</td>
  </tr>
  <tr>
    <td class="tb-field-head">姓名</td>
    <td class="tb-field-content">
      <input type="text" value="tony" />
    </td>
    <td class="tb-field-head">性别</td>
    <td class="tb-field-content">
      <select name="sex" id="">
        <option value="male">男</option>
        <option value="female">女</option>
      </select>
    </td>
  </tr>
  <tr>
    <td class="tb-field-head">年龄</td>
    <td class="tb-field-content"></td>
    <td class="tb-field-head">身高</td>
    <td class="tb-field-content"></td>
  </tr>
  <tr class="tb-bottom">
    <td colspan="4">
      <span>每页显示</span>
      <input type="text" class="page-num" value="10">
      <span>行数据</span>
      <button class="u-btn">查询</button>
      <button class="u-btn">重置</button>
      <button class="u-btn">返回</button>
    </td>
  </tr>
</table>
```

查询表格

姓名 tony

性别 男

年龄

身高

每页显示 10 行数据 查询 重置 返回

2.9 tab切换

tab切换需要使用样式 `m-tab` ,内部有 `tab-head` , `tab-body` , `item` , `tab-box` 标签。

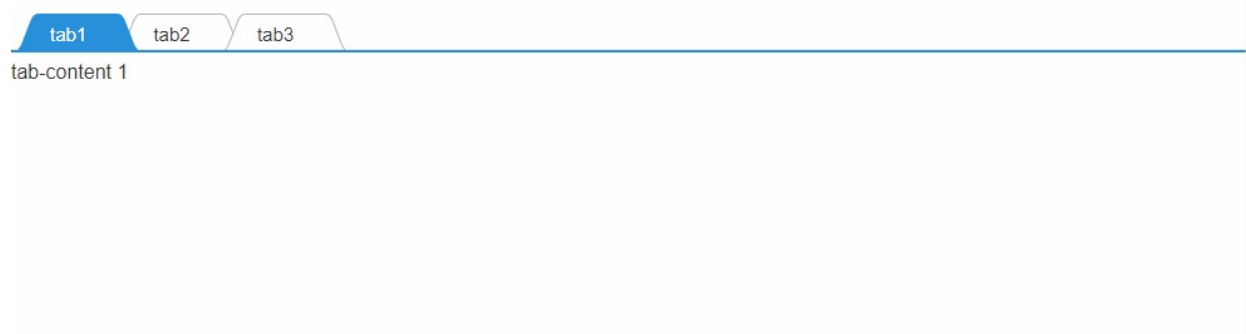
- `tab-head` tab头部标签
- `item` 是 `tab-head` 内部单元
- `tab-body` tab主体标签

- `tab-box` 是 `tab-body` 内部单元
使用事例 html部分

```
<!-- html部分 -->
<div class="m-tab">
  <ul class="tab-head" id="tab_head">
    <li class="item">tab1</li>
    <li class="item">tab2</li>
    <li class="item">tab3</li>
  </ul>
  <ul class="tab-body" id="tab_body">
    <li class="tab-box">tab-content 1</li>
    <li class="tab-box">tab-content 2</li>
    <li class="tab-box">tab-content 3</li>
  </ul>
</div>
```

使用js

```
//tab切换 js调用
var tabDemo = new TabsChange('tab_head','tab_body');
tabDemo.init();
```



结语

文档也有地方写的不是很清楚，后续还会修正功能和文档。工作中如果遇到不清楚的地方，或者有什么需求，请及时与我沟通。