

1. 富文本编辑器的实现步骤

1. 添加如下的 `layui` 表单行:

```
1 <div class="layui-form-item">
2   <!-- 左侧的 label -->
3   <label class="layui-form-label">文章内容</label>
4   <!-- 为富文本编辑器外部的容器设置高度 -->
5   <div class="layui-input-block" style="height: 400px;">
6     <!-- 重要: 将来这个 textarea 会被初始化为富文本编辑器 -->
7     <textarea name="content"></textarea>
8   </div>
9 </div>
```

2. 导入富文本必须的 `script` 脚本:

```
1 <!-- 富文本 -->
2 <script src="/assets/lib/tinymce/tinymce.min.js"></script>
3 <script src="/assets/lib/tinymce/tinymce_setup.js"></script>
```

3. 调用 `initEditor()` 方法, 初始化富文本编辑器:

```
1 // 初始化富文本编辑器
2 initEditor()
```

2. 图片封面裁剪的实现步骤

1. 在 `<head>` 中导入 `cropper.css` 样式表:

```
1 <link rel="stylesheet" href="/assets/lib/cropper/cropper.css" />
```

2. 在 `<body>` 的结束标签之前, 按顺序导入如下的 js 脚本:

```
1 <script src="/assets/lib/jquery.js"></script>
2 <script src="/assets/lib/cropper/Cropper.js"></script>
3 <script src="/assets/lib/cropper/jquery-cropper.js"></script>
```

3. 在表单中, 添加如下的表单行结构:

```
1 <div class="layui-form-item">
2   <!-- 左侧的 label -->
3   <label class="layui-form-label">文章封面</label>
4   <!-- 选择封面区域 -->
5   <div class="layui-input-block cover-box">
6     <!-- 左侧裁剪区域 -->
7     <div class="cover-left">
8       
9     </div>
10    <!-- 右侧预览区域和选择封面区域 -->
11    <div class="cover-right">
```

```

12         <!-- 预览的区域 -->
13         <div class="img-preview"></div>
14         <!-- 选择封面按钮 -->
15         <button type="button" class="layui-btn layui-btn-danger">选择封面
16     </button>
17 </div>
18 </div>

```

4. 美化的样式：

```

1  /* 封面容器的样式 */
2  .cover-box {
3      display: flex;
4  }
5
6  /* 左侧裁剪区域的样式 */
7  .cover-left {
8      width: 400px;
9      height: 280px;
10     overflow: hidden;
11     margin-right: 20px;
12 }
13
14 /* 右侧盒子的样式 */
15 .cover-right {
16     display: flex;
17     flex-direction: column;
18     align-items: center;
19 }
20
21 /* 预览区域的样式 */
22 .img-preview {
23     width: 200px;
24     height: 140px;
25     background-color: #ccc;
26     margin-bottom: 20px;
27     overflow: hidden;
28 }

```

5. 实现基本裁剪效果：

```

1  // 1. 初始化图片裁剪器
2  var $image = $('#image')
3
4  // 2. 裁剪选项
5  var options = {
6      aspectRatio: 400 / 280,
7      preview: '.img-preview'
8  }
9
10 // 3. 初始化裁剪区域
11 $image.cropper(options)

```

3. 更换裁剪的图片

1. 拿到用户选择的文件

```
1    var file = e.target.files[0]
```

2. 根据选择的文件，创建一个对应的 URL 地址：

```
1    var newImgURL = URL.createObjectURL(file)
```

3. 先 **销毁** 旧的裁剪区域，再 **重新设置图片路径**，之后再 **创建新的裁剪区域**：

```
1    $image
2      .cropper('destroy')      // 销毁旧的裁剪区域
3      .attr('src', newImgURL)  // 重新设置图片路径
4      .cropper(options)        // 重新初始化裁剪区域
```

4. 将裁剪后的图片，输出为文件

```
1    $image
2      .cropper('getCroppedCanvas', { // 创建一个 Canvas 画布
3        width: 400,
4        height: 280
5      })
6      .toBlob(function(blob) {      // 将 Canvas 画布上的内容，转化为文件对象
7        // 得到文件对象后，进行后续的操作
8      })
```