

1. 基本使用步骤

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. 在卡片的 `layui-card-body` 主体区域中，定义如下的 HTML 结构：

```
1 <!-- 第一行的图片裁剪和预览区域 -->
2 <div class="row1">
3   <!-- 图片裁剪区域 -->
4   <div class="cropper-box">
5     <!-- 这个 img 标签很重要，将来会把它初始化为裁剪区域 -->
6     
7   </div>
8   <!-- 图片的预览区域 -->
9   <div class="preview-box">
10    <div>
11      <!-- 宽高为 100px 的预览区域 -->
12      <div class="img-preview w100"></div>
13      <p class="size">100 x 100</p>
14    </div>
15    <div>
16      <!-- 宽高为 50px 的预览区域 -->
17      <div class="img-preview w50"></div>
18      <p class="size">50 x 50</p>
19    </div>
20  </div>
21 </div>
22
23 <!-- 第二行的按钮区域 -->
24 <div class="row2">
25   <button type="button" class="layui-btn">上传</button>
26   <button type="button" class="layui-btn layui-btn-danger">确定</button>
27 </div>
```

4. 美化的样式：

```
1 /* 设置卡片主体区域的宽度 */
2 .layui-card-body {
3   width: 500px;
4 }
5
6 /* 设置按钮行的样式 */
7 .row2 {
8   display: flex;
9   justify-content: flex-end;
10  margin-top: 20px;
```

```

11     }
12
13     /* 设置裁剪区域的样式 */
14     .cropper-box {
15         width: 350px;
16         height: 350px;
17         background-color: cyan;
18         overflow: hidden;
19     }
20
21     /* 设置第一个预览区域的样式 */
22     .w100 {
23         width: 100px;
24         height: 100px;
25         background-color: gray;
26     }
27
28     /* 设置第二个预览区域的样式 */
29     .w50 {
30         width: 50px;
31         height: 50px;
32         background-color: gray;
33         margin-top: 50px;
34     }
35
36     /* 设置预览区域下方文本的样式 */
37     .size {
38         font-size: 12px;
39         color: gray;
40         text-align: center;
41     }
42
43     /* 设置图片行的样式 */
44     .row1 {
45         display: flex;
46     }
47
48     /* 设置 preview-box 区域的的样式 */
49     .preview-box {
50         display: flex;
51         flex-direction: column;
52         flex: 1;
53         align-items: center;
54     }
55
56     /* 设置 img-preview 区域的样式 */
57     .img-preview {
58         overflow: hidden;
59         border-radius: 50%;
60     }

```

5. 实现基本裁剪效果：

```
1 // 1.1 获取裁剪区域的 DOM 元素
2 var $image = $('#image')
3 // 1.2 配置选项
4 const options = {
5   // 纵横比
6   aspectRatio: 1,
7   // 指定预览区域
8   preview: '.img-preview'
9 }
10
11 // 1.3 创建裁剪区域
12 $image.cropper(options)
```

2. 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
```

3. 将裁剪后的图片，输出为 base64 格式的字符串

```
1 var dataURL = $image
2   .cropper('getCroppedCanvas', { // 创建一个 Canvas 画布
3     width: 100,
4     height: 100
5   })
6   .toDataURL('image/png') // 将 Canvas 画布上的内容，转化为 base64 格式的字符串
```