1. **布局实现**

3层+结果

第一层：img，原始文件，通过fileReader直接读取，输出DataURL赋给img。

第二层：div，美化crop过程。边框采用半透明色，提示用户当前非选中区域。

第三层：div，美化。边框非透明，提示用户选中的裁剪区域。

结果：裁剪后的结果（因为每次重选原始图片，需要清空前次结果，需要设置scr为空（而不是ng-scr，因为ng-scr赋值后，是**操作**scr来显示图片），而直接设置scr为空，会有默认图片显示，所以把结果也纳入操作范围，即选择图片后，结果的scr清空，并隐藏，裁剪之后，显示结果

1. **裁剪实现**

鼠标移动/滚轮滚动：计算currentCropZone的4个属性，top/left/Width/height，以便计算实际需要从原始图片文件裁剪的位置。同时以currentCropZone为base，计算第二三层对应的参数（第二层的borderWidth，第三层的left/top/WH）

1. **参数：**

DefaultOptions:初始化时输入的option都会和defaultOptions合并，最终使用的是defaultOptions。

* 1. elementId：3层+结果 对应的元素Id。用来读取原始img的参数，设置二三层以及结果层
  2. L1origImgMaxWH：最大能处理的img size
  3. L3BorderWidth：第三层div的border（提示用户选中区域）
  4. cropImgWH：最终裁剪成的size
  5. zoomStep：滚轮滚动时，缩放的size
  6. bindedEvent：各种事件的绑定（第二三层移动，滚轮缩放）

1. **内部参数**
   1. bindState：是否可以移动currentCropZone
   2. allowBindEvent：是否可以在原始图片上绑定事件（移动然后剪切，小图片无此操作）
   3. Container：mousemove绑定的元素。默认是body元素。意味着鼠标在整个页面移动时，还是可以进行位置计算。
   4. **currentCropZone**：当前cropZone的参数tlwh，第二三层根据它进行计算
   5. allElement：根据元素id，获得元素
   6. L1\_origImgPos：如命，原始img的tlwh
2. **函数**
   1. isInt：判断是否为整数。
   2. init\_part1：读取options，merge到defaultOptions，并判断其中是否合格（元素是否存在，WH是否为整数，并为正）
   3. init\_part2: 读取origImg的参数，并初始化currentCropZone，L2L3
   4. calcSetL2BorderWidth：根据L1\_origImgPos和currentCropZone计算并设置L2的border
   5. calcSetL3Pos：根据L1\_origImgPos和currentCropZone计算并设置L3的位置
   6. ~~calcSetL2BorderWidthWhenZoom：缩放cropZone时，计算L2 border~~
   7. ~~calcSetL3PosWhenZoom：缩放cropZone时，根据L1\_origImgPos和currentCropZone计算并设置L3的位置~~
   8. **calcCropZoneMove**：根据鼠标位置，计算当前currentCropZone，
   9. **calcCropZoneZoom：**根据鼠标滚动，计算当前currentCropZone
   10. nextOp：绑定、非绑定cropZone的move/zoom和body的click属性
   11. cropGenerateDataURL：产生最终dataURL

**代码**

**Html：**

<**div** id=**"container"**>  
 <**img** id=**'L1\_origImg'** alt=**""** style=**"**float:**left**;**"** onchange=**"angular**.**element**(**this**).**scope**().chooseImg(**this**)**"**/>  
 <**div** id=**'L2\_coverZone'** style=**"**float:**left**;position:**absolute**;display:**none**;border-style:**solid**;border-color:**rgba**(128,128,128,0.6);display:**none**;**"** draggable=**"true"**></**div**>  
<**div** id=**"L3\_cropImgBorder"** style=**"**float:**left**;position:**absolute**;display:**none**;border-color:**silver**;border-style:**dashed**;**"**>  
</**div**><**img** id=**"croppedImg"** ng-src=**"{{cropedDataURL}}"**></**div**>

**Angular controller**

**var** crop=Crop.create({  
 **elementId**:{  
 **L1\_origImg**:**'L1\_origImg'**,  
 **L2\_coverZone**:**'L2\_coverZone'**,  
 **L3\_cropImgBorder**:**'L3\_cropImgBorder'**,  
 **croppedImg**:**'croppedImg'**,  
 },  
 **L1origImgMaxWH**:{  
 **width**:1376,  
 **height**:768,  
 },  
 **L3BorderWidth**:{  
 **borderLeftWidth**:1,  
 **borderTopWidth**:1,  
 },  
 *//最终裁剪出来的图片size* **cropImgWH**:{  
 **width**:300,  
 **height**:200,  
 },  
 *//滚轮滚动时，WH* **zoomStep**:{  
 **horizontal**:5,*//左右每边* **vertical**:5,*//上下每边* },  
**~~bindedEvent~~**~~:{~~ **~~zoomZone~~**~~:~~**~~'mousewheel DOMMouseScroll'~~**~~,~~ **~~moveZone~~**~~:~~**~~'mousemove'~~**~~,~~*~~//bind to body, when this event binded, the choose img in L1\_origImg show in L3\_cropImgBorder~~* **~~cropChooseImg~~**~~:~~**~~'click'~~**~~,~~ *~~//bind to L3\_cropImgBorder, this event define if moveZone still be binded(L3\_cropImgBorder still show choosn img or not)~~* ~~}~~  
})  
  
$scope.chooseImg=**function**(){  
 asyncFunc.readFile(**'userIcon'**,**'dataURL'**,20000000).then(**function**(data) {**if**(data.**rc**){  
 alert(data.**msg**)  
 }  
 **else**{  
 **var** img = **document**.getElementById(**'L1\_origImg'**)  
 img.onload=**function**(e){**var** result=crop.init()  
 **if**(result.**rc**>0){  
 alert(result.**msg**)  
 }  
 }  
 img.**src** = data;}  
 })  
}

$scope.crop=**function**(){  
 **var** result=crop.cropGenerateDataURL()  
 **if**(result[**'rc'**]){  
 alert(result.**msg**)  
 }**else**{  
 $scope.**cropedDataURL**=result  
 }  
  
}