程序的主要结构：

1. 训练图像，以获得各部件可能的位置（train\_model\_components）
2. 创建训练部件模型（create\_trained\_component\_model）
3. 查找部件，获得各部件位置参数（find\_component\_model）
4. 显示部件
5. 训练图像
   1. 训练图像主要函数是train\_model\_components，其需要提供几个重要参数：1模板图像（ModelImage）、2初始部件位置区域（InitialComponents），即在模板图像中，各部件的位置区域，是一个区域数组，3训练图像集（TraniningImage），是一个图像数组，输出包括一个模版图像各部件的轮廓（ModelComponents）和一个训练结果的句柄（ComponentTrainingID）及其它参数。
   2. 模板图像的获得，read\_image。
   3. 初始部件位置区域的获得，绘制好区域后，使用concat\_obj。
   4. 训练图像集的获得，使用gen\_empty\_obj (TrainingImages)，创建一个空的集合，然后使用read\_Image读取图像，再用concat\_obj将其合并。
   5. 模版图像各部件的轮廓是一个数组，每个元素对应一个部件的轮廓。
   6. 训练结果的句柄（ComponentTrainingID）在后续的create\_trained\_component\_model函数中会用到。
6. 创建训练部件模型

2.1 输出的主要参数是模型句柄ComponentModelID。

2.2 创建成功后，训练结果句柄ComponentTrainingID不再有用，应清除，方法是调用clear\_training\_components (ComponentTrainingID)。

1. 查找部件

使用函数find\_component\_model从图像中查找模型匹配，查找的综合匹配分数保存在参数Score中，注意它表示一组结果的分数，而不是单个部件的匹配分数，如果在一幅图像中查到到多组匹配，Score便成为一个数组，每个元素代表对应组的匹配分数。

1. 显示部件

调用get\_found\_component\_model函数，从查找结果中读取结果，由于可能查找到多组匹配，因此，应指定结果的序号。

1. 其它函数

5.1 get\_component\_relations 得到指定部件（部件序号为ReferenceCompoent）与模板图像中其他各个部件之间的位置关系和角度关系，如果要得到第i幅训练图像中的关系（而不是模版图像），应将参数Image设成i。各部件的位置在图中用3像素的圆表示，参考部件之外的其他部件与参考部件的关系在图中被表示出来。位置关系用矩形表示，方向关系用扇形表示，中心圆置于平均相对位置处，矩形描述了各部件参考点相对于参考部件姿态的运动，而扇形描述了相对方向的变动。如果各部件在所有的图像中表现出相同的相对方向，扇形退化成一条直线。

5.2 get\_mbutton获得鼠标位置，可以用来等待鼠标响应