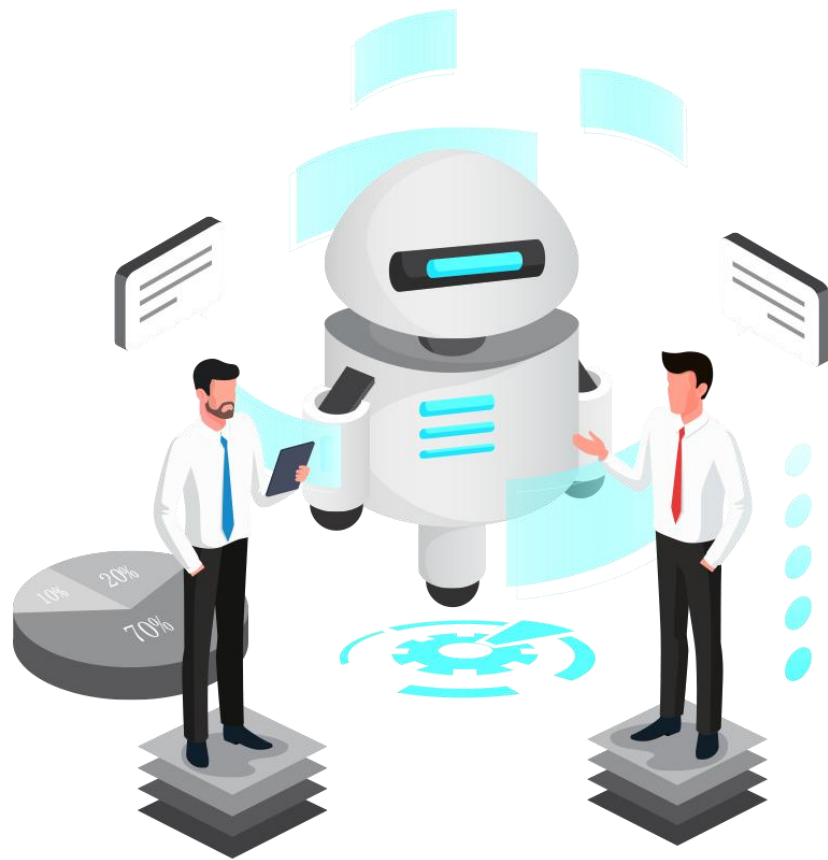


人脸比对

mixly (米思齐)



什么是人脸识别

人脸识别是基于人的脸部特征信息进行身份识别的一种生物识别技术。针对输入的人脸图像或者视频流，首先判断其是否存在人脸，如果存在人脸，则进一步的给出每个脸的位置、大小和各个主要面部器官的位置信息。然后依据这些信息，进一步提取每个人脸中所蕴涵的身份特征数据，并将其与已知的人脸进行对比，从而识别每个人脸的身份。



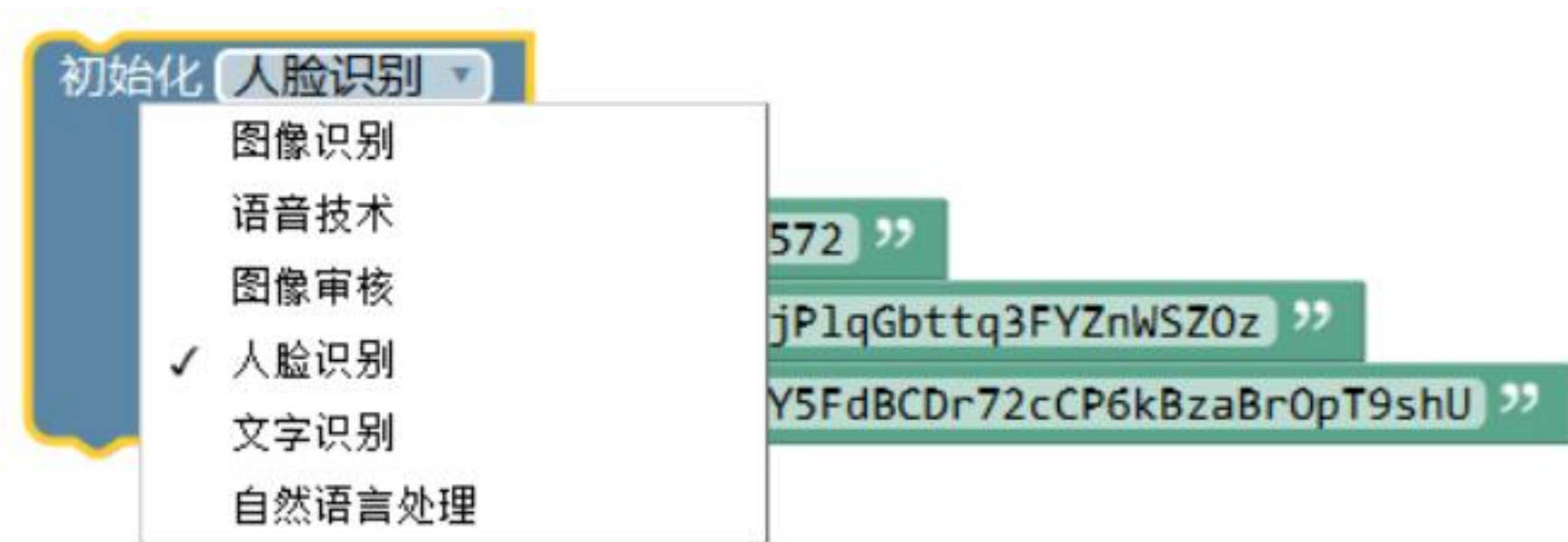
人脸识别的应用领域

序号	领域	说明
(1)	通关领域	护照认证、身份证认证、工作认证，这些可作为通关判断条件之一，再加人脸认证作为通关判断条件之一，均符合要求则，则自放行。涉及到门禁控制，闸机控制等。
(2)	考试领域	确认报名考试的身份证与参加考试的人身份证及人为同一人。
(3)	在线教学领域	人脸认真考勤、人脸认证巡查、远程认证。
(4)	金融行业	支付人脸认证识别、VIP客户识别、商业智能。
(5)	园区管理	园区出入口管理，刷卡入区，加一重人脸认证即可自行出入。
(6)	地铁	刷脸进站，乞讨人员布控，在逃人员布控。
(7)	其他应用领域	人脸验证驾照，签证；监控公园监控、街道监控；人机交互式游戏，主动计算等。

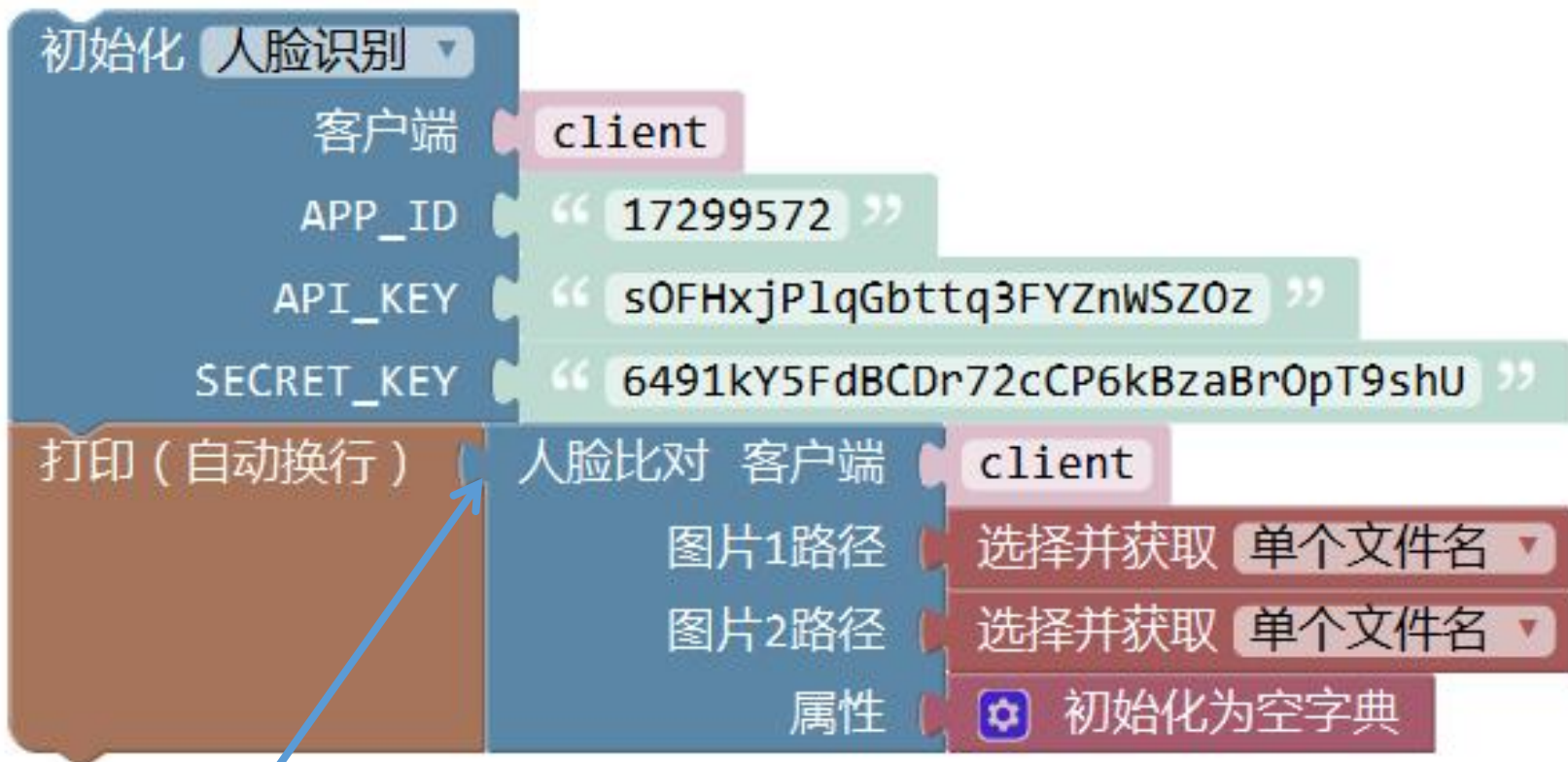
人脸比对

- 01 两张人脸图片之间比对
 - 02 使用摄像头拍照并比对
 - 03 语音播报对比结果(语音合成)
 - 04 使用摄像头拍照与多张人脸比对
 - 05 输出每一张图片的对比结果
 - 06 显示出比对成功的图片
 - 07 多张人脸图片之间比对
 - 08 任务扩展：找出你所在的班级
-
-
-

初始化，选择“人脸识别”



选择同一个人的两张人脸图片、不同的两个人的人脸照片进行比对。



输出

想一想：能否把图片1路径改为选择并获取“多个文件名”？

选择并获取 单个文件名 ▾

- ✓ 单个文件名
- 多个文件名
- 文件目录

试一试、比一比：

打印

选择并获取 单个文件名 ▾

打印

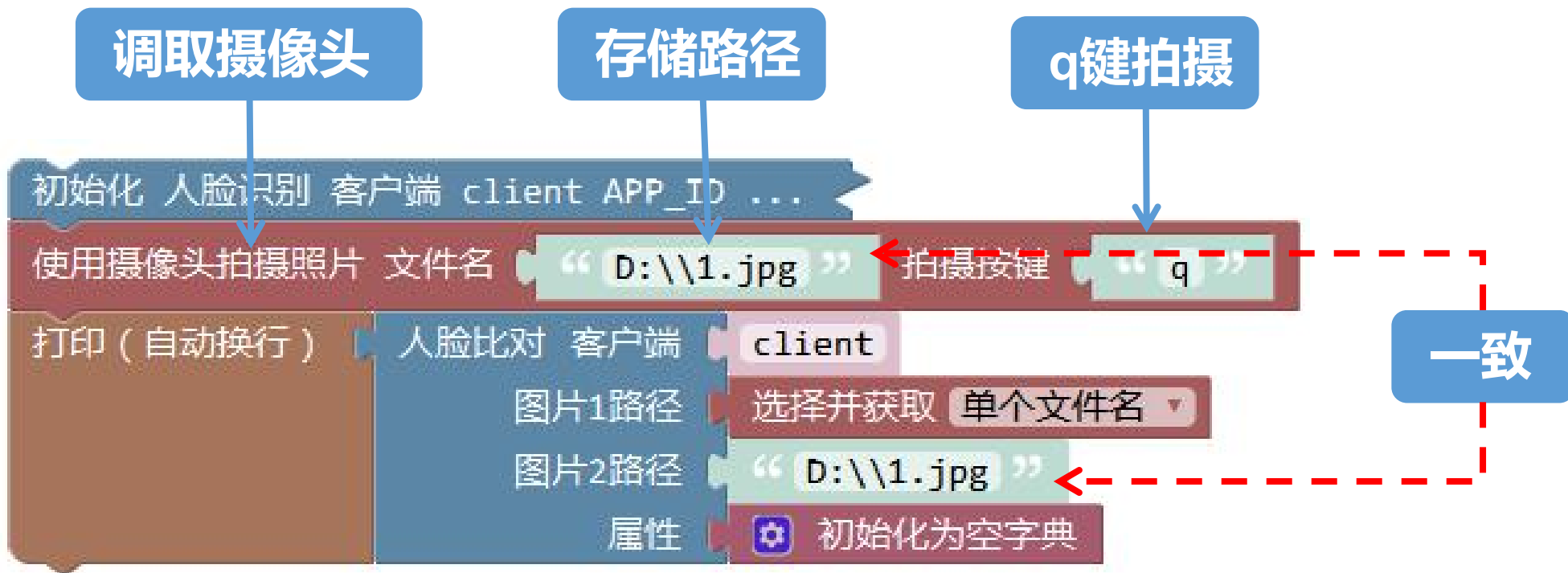
选择并获取 多个文件名 ▾

摄像头拍摄的照片存放于D盘下。

注意：电脑是否有D盘，若没有，请更改路径，存放于其他盘。

The image shows a Scratch script with the following blocks:

- 初始化 人脸识别** (Initialize Face Recognition) block containing:
 - 客户端** (Client): client
 - APP_ID**: "17299572"
 - API_KEY**: "s0FHxjPlqGbttq3FYZnWSZOz"
 - SECRET_KEY**: "6491kY5FdBCDr72cCP6kBzaBr0pT9shU"
- 使用摄像头拍摄照片** (Use Camera to Take Photo) block containing:
 - 文件名** (Filename): "D:\\\\1.jpg"
 - 拍摄按键** (Shoot Button): "q"
- 打印 (自动换行)** (Print) block containing:
 - 人脸比对** (Face Comparison) block containing:
 - 客户端** (Client): client
 - 图片1路径** (Image 1 Path): 选择并获取 单个文件名 (Select and Get Single Filename)
 - 图片2路径** (Image 2 Path): "D:\\\\1.jpg"
 - 属性** (Properties): 初始化为空字典 (Initialize as Empty Dictionary)



结果解读

```
{'score': 85.2461853,  
  'face_list':  
  [{'face_token': 'e5b917496fa45dcd1c256ae348182748'},  
   {'face_token': '6ff99615ebb396468cd1b9c63399a924'}  
  ]}
```

获取需要的结果

获取“人脸比对”结果，即“**score**”。

方法一：使用获取结果“人脸比对”模块。



The image shows a sequence of Scratch-style blocks for a face comparison task:

- 初始化 人脸识别 客户端 client APP_ID ...** (Initialize face recognition client)
- 使用摄像头拍摄照片** (Use camera to take photo) with sub-blocks:
 - 文件名: "D:\\1.jpg"
 - 拍摄按键: "q"
- 打印 (自动换行)** (Print) block containing:
 - 获取结果** (Get results) block with a dropdown menu:
 - 图像识别
 - 语音识别
 - ✓ 人脸比对 (checked)
 - 文字识别
 - 短文本相似度
 - 情感倾向分析
 - 人脸比对 客户端** (Face comparison client) block with value: client
 - 选择并获取 单个文件名** (Select and get single filename) block with value: "D:\\1.jpg"
 - 属性** (Properties) block with value: 初始化为空字典 (Initialize as empty dictionary)

获取需要的结果

方法二：从“result”中获取键 “score” 对应值。



变量

result 赋值为

变量赋值模块：通过该模块，我们能够重新实现重新变量赋予一个值的功能。

result 赋值为

result

获取键

“ score ”

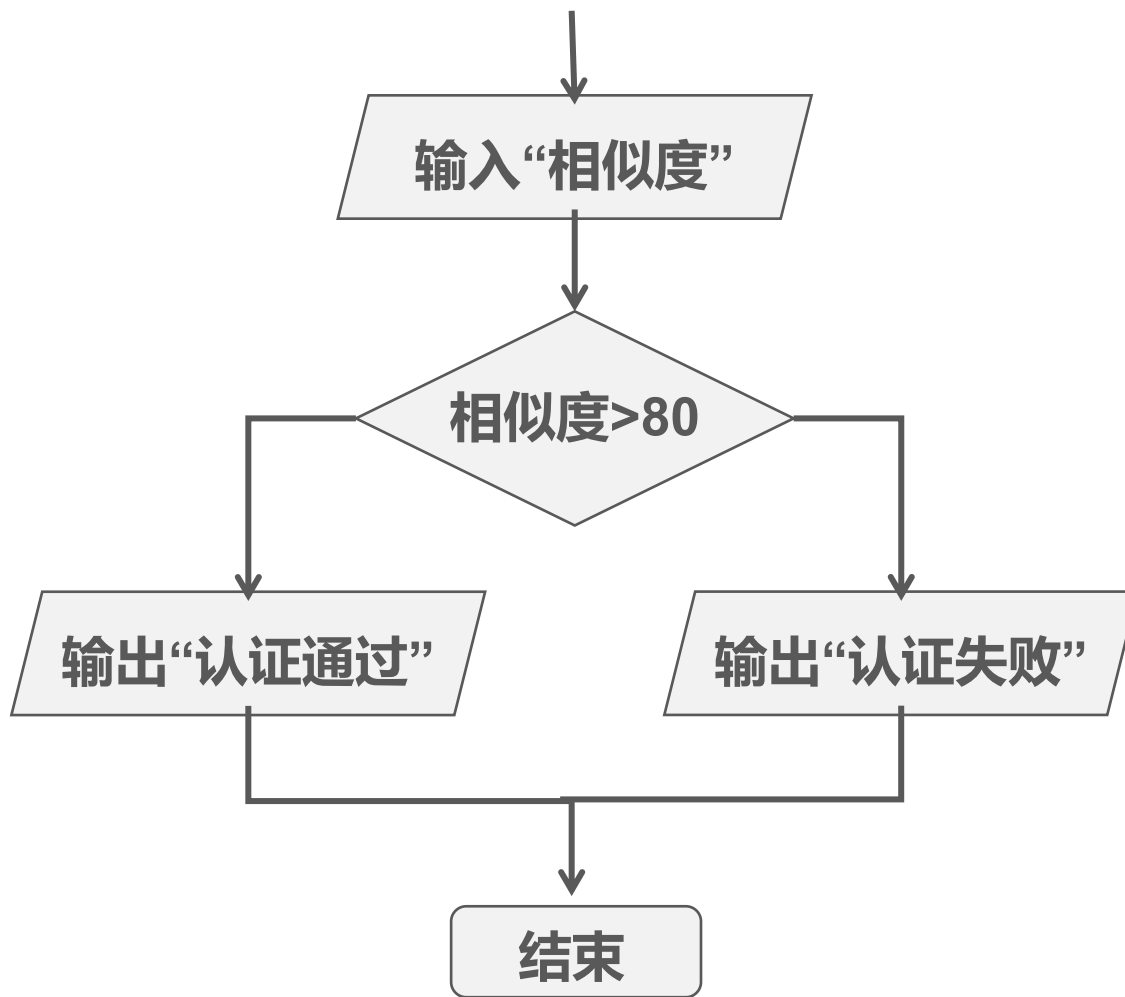
对应值

通过变量赋值模块，我们已经把“85.2461853”重新赋值给了变量“result”。

result

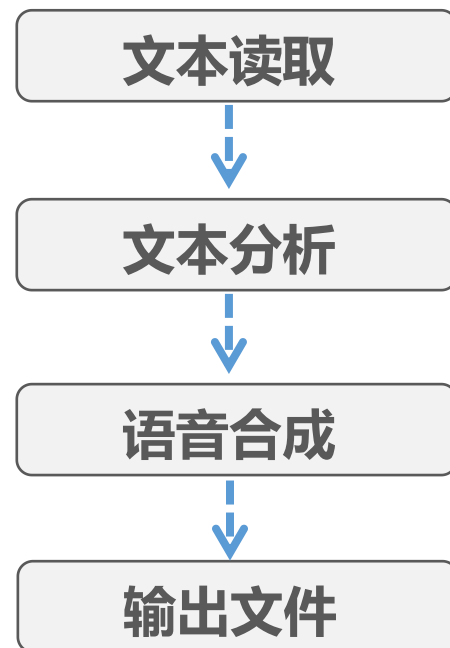
变量值模块：我们能够提供变量“result”的值。

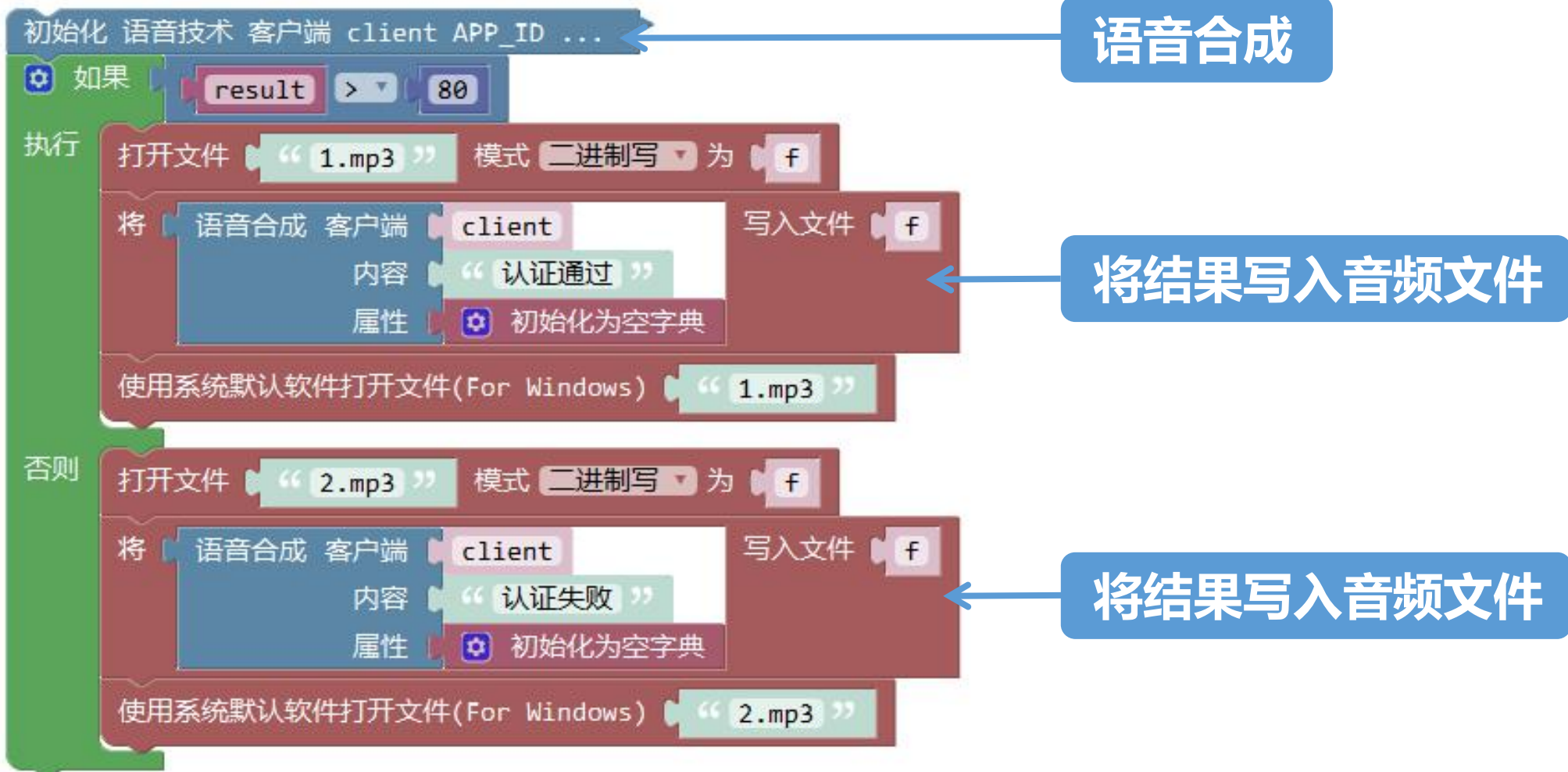
设置相似度为80%，如果比对结果 >80 ，语音播报认证通过；否则语音播报认证失败。





语音合成一般流程







控制

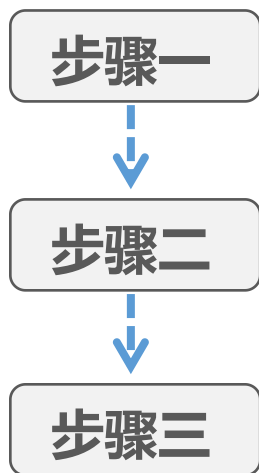
时间延迟、条件执行、循环执行、获取时间、跳出循环等。

控制模块中主要执行的内容是对程序结构进行的相应控制。有了控制，才能实现不同程序模块的选择和跳转。

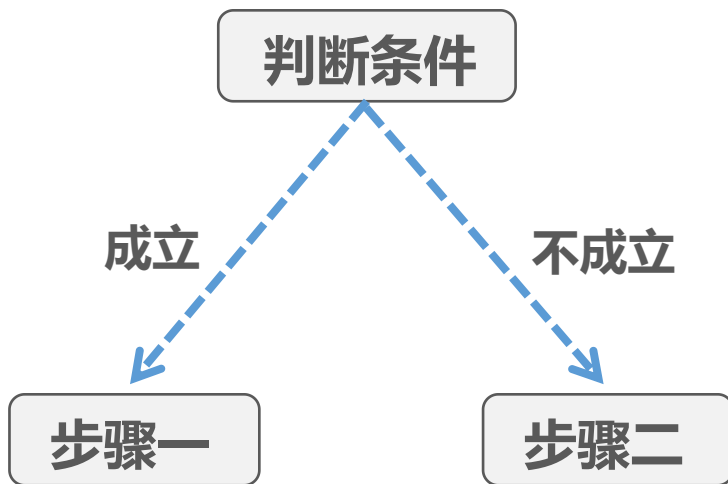
程序的控制住主要有三种结构：1、**顺序结构** 2、**判断结构** 3、**循环结构**

程序结构

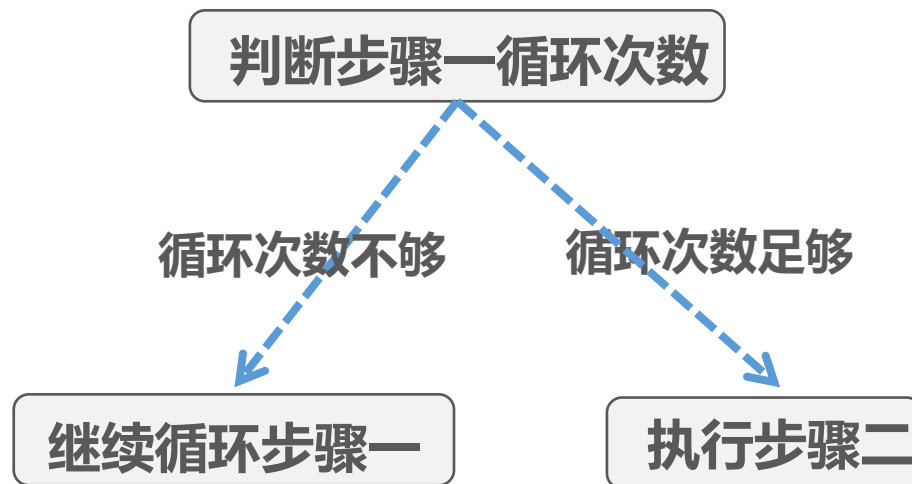
顺序结构



判断结构



循环结构



顺序结构的程序是由上至下，逐一执行。判断结构通过条件语句进行判断。

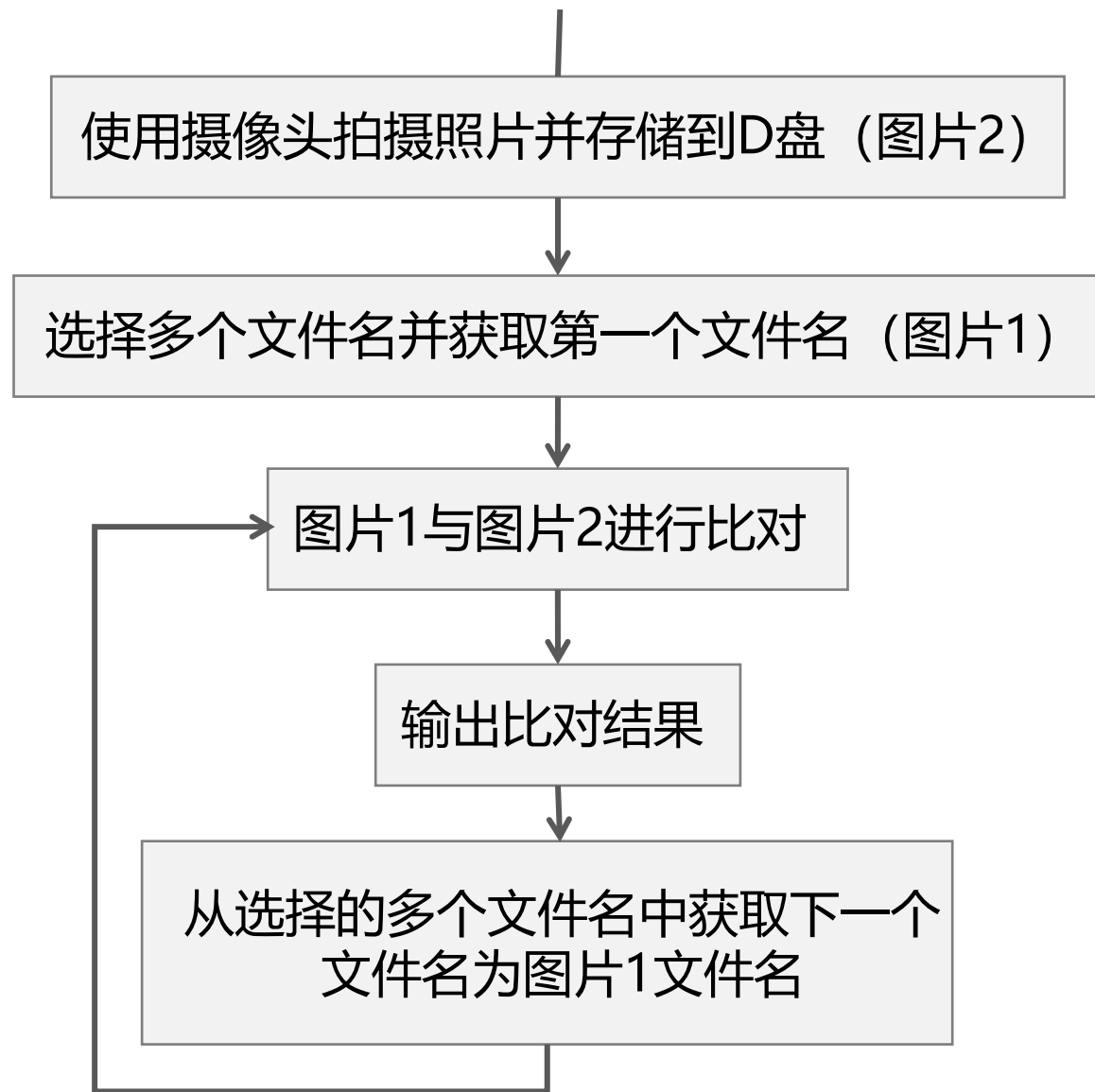
条件语句



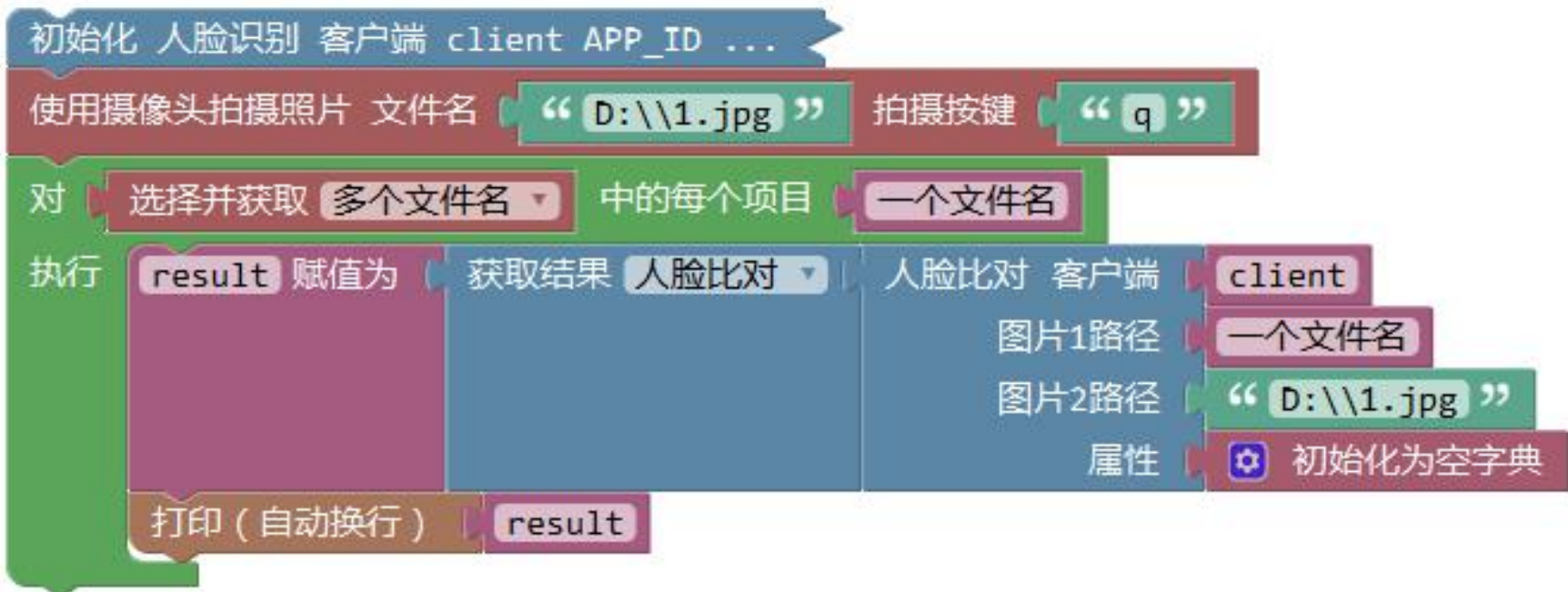
点击小齿轮展开“如果”模块



设置“如果-否则”语句



1:N比对功能，将比对的图片存放在同一级目录下。



判断每一张人脸图片是不是你自己。

将你的图像和图像比对，根据比对分数输出比对结果成功与否



字符串



“ Mixly ”

字符串模块：该模块会提供一个字符串，内容直接在双引号中输入即可。



“ Hello ” 连接 “ Mixly ”

文本连接模块：该模块能够将两个字符串结合成一个字符串。

尝试与实践：

找出选择获取的多个人脸图片中一共有“多少个”你自己。

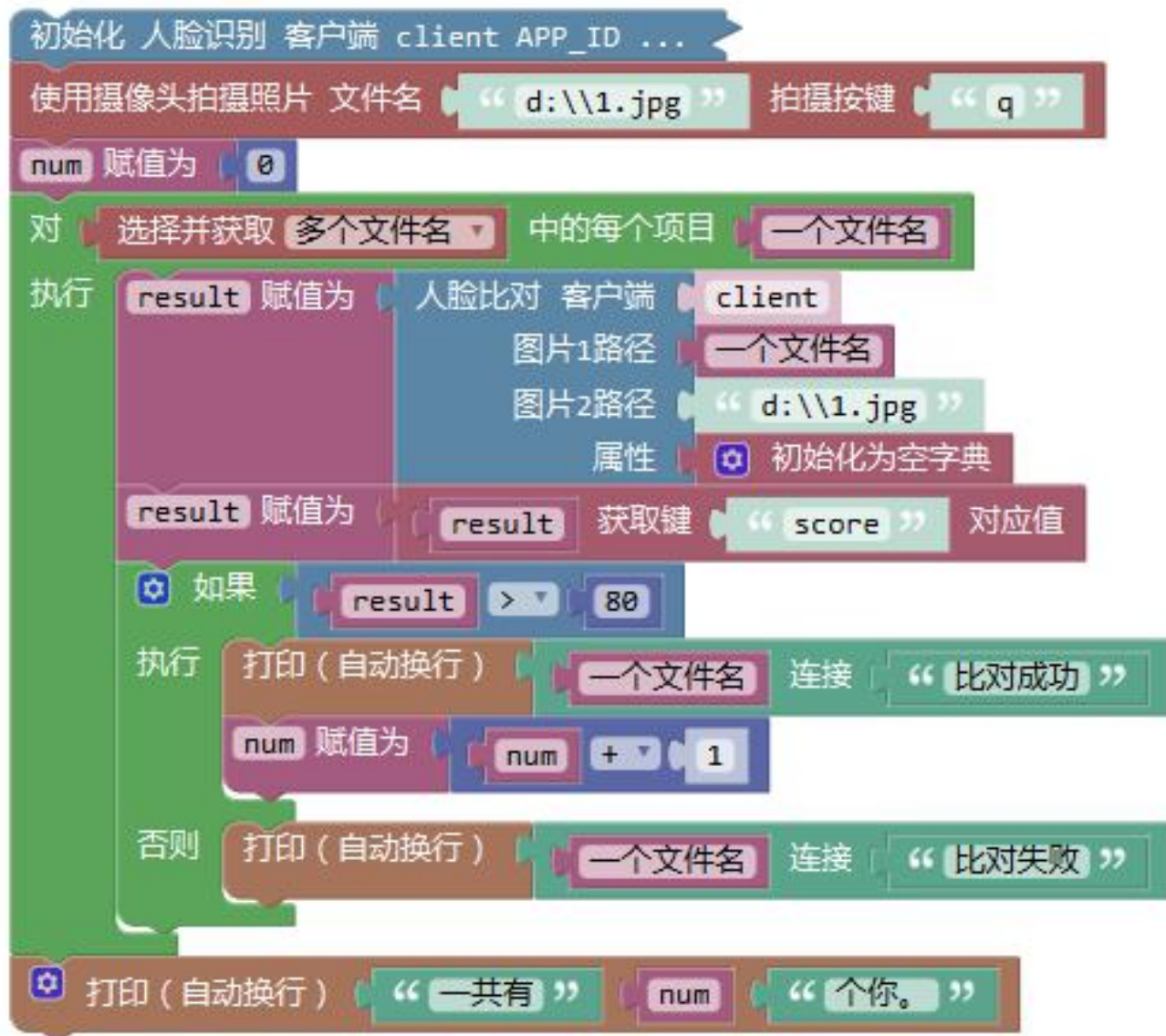
累计个数：



输出结果：



一共有“多少个”你自己:





读取比对成功的图片的内容

将结果写入图片文件

写入文件“f”的不是文件名，而是文件内容。



python文件操作

新建/打开文件



进行读/写文件



关闭文件

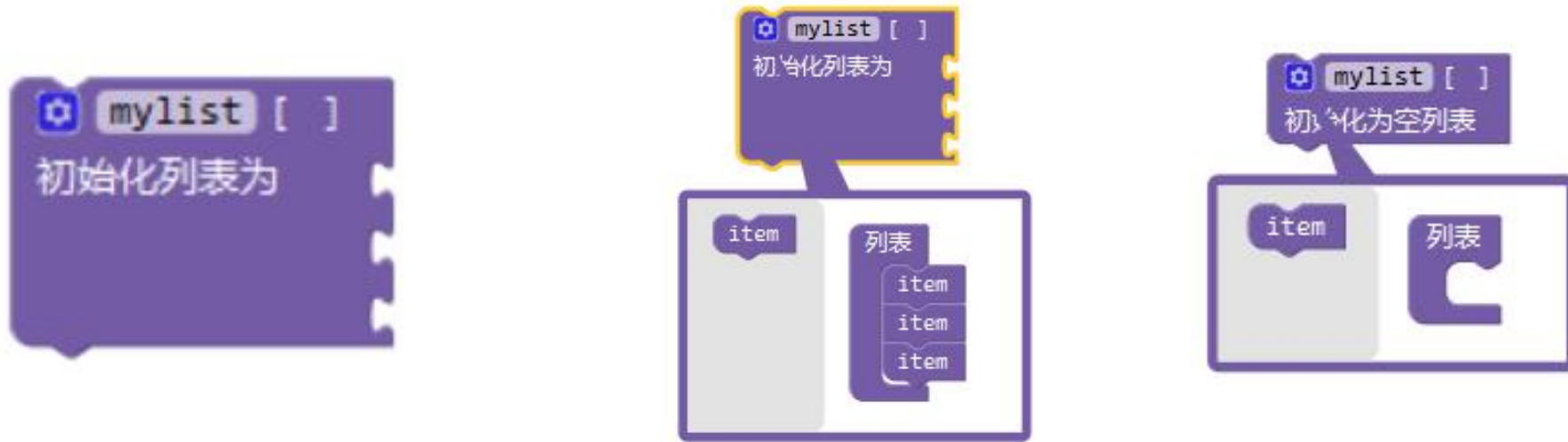
制作所有需要比对的人脸图像路径列表，与使用摄像头拍摄的照片进行比对。



```
初始化 人脸识别 客户端 client APP_ID ...
mylist 赋值为 初始化为空列表
使用摄像头拍摄照片 文件名 "d:\\1.jpg" 拍摄按键 "q"
对 选择并获取 多个文件名 中的每个项目 一个文件名
执行 mylist 增加 一项 一个文件名 到末尾
对 range 从 0 到 获取长度 mylist 步长为 1 中的每个项目 i
执行
  result 赋值为 人脸比对 客户端 client
  图片1路径 mylist 获取第 i 项
  图片2路径 "d:\\1.jpg"
  属性 初始化为空字典
  result 赋值为 result 获取键 "score" 对应值
  打印 (自动换行) mylist 获取第 i 项
  打印 (自动换行) result
```

列表

列表像一个大盒子，可理解为遗传用来存储数据的空间，可以储存一定的数字或字符串。“列表”分类中的模块就是实现**放入数据**到存储空间或从存储空间中**取出数据**的操作。



初始化列表模块：该模块能够创建一个列表的功能。该模块，默认列表中数据的数量为3个，点击模块小齿轮。我们能够自由增加列表数据的数量。减少列表的数据位置，初始化为空列表。

列表

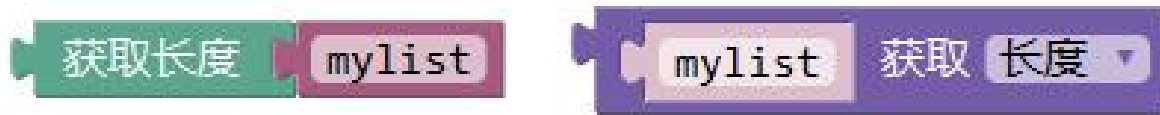


获取列表数据模块：该模块能够实现，在列表中取出数据的功能。



我们在列表中存入了数据之后，我们需要取出数据。通过获取数据模块，我们能够取出列表中数据，数据在列表中的排列序号1从开始，然后一直递加。

列表



获取列表长度模块：该模块能够实现，计算列表长度的功能。获取指定列表包含多少个元素，得到数值。



我们在往列表里面存入数据之后，我们想知道里面有多少数据就需测量列表的长度。通过打印获取列表mylist的长度，我们能够获得列表的长度。

列表



在列表末尾添加数据模块：在列表的末尾添加一个数据。

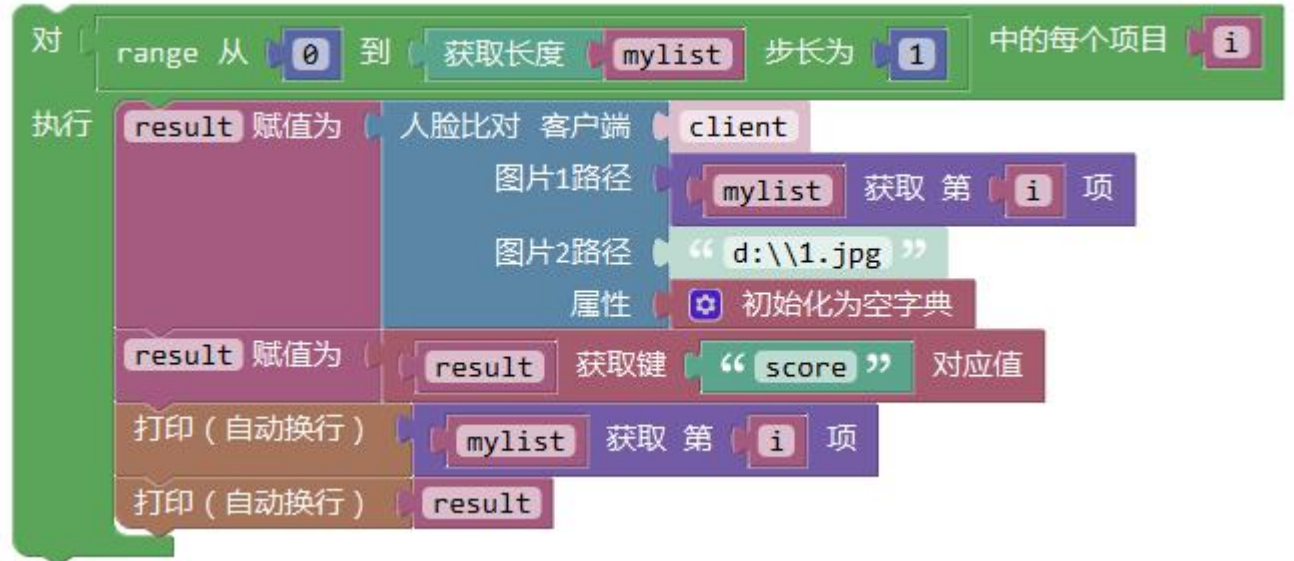


列表mylist存储了我们获取的多个文件名的路径。

列表

mylist:

```
['D:/Mixly/人脸/01.jpg',  
'D:/Mixly/人脸/02.jpg',  
'D:/Mixly/人脸/03.jpg',  
'D:/Mixly/人脸/04.jpg',  
'D:/Mixly人脸/05.png',  
'D:/Mixly/人脸/06.jpg']
```



这里共选择了6张人脸图片，列表mylist的长度为6。使用了0到[mylist的长度]的循环语句以此获得每张人脸图片（图片1）的路径，随后与图片2路径下的图片进行对比。

```
初始化 人脸识别 客户端 client APP_ID ...
mylist1 赋值为 初始化为空列表
mylist2 赋值为 初始化为空列表

对 选择并获取 多个文件名 中的每个项目 一个文件名
执行 mylist1 增加 一项 一个文件名 到末尾

对 选择并获取 多个文件名 中的每个项目 另一个文件名
执行 mylist2 增加 一项 另一个文件名 到末尾

对 range 从 0 到 获取长度 mylist1 步长为 1 中的每个项目 i
执行
  对 range 从 0 到 获取长度 mylist2 步长为 1 中的每个项目 j
  执行
    result 赋值为 人脸比对 客户端 client
    图片1路径 mylist1 获取 第 i 项
    图片2路径 mylist2 获取 第 j 项
    属性 初始化为空字典
    result 赋值为 result 获取键 "score" 对应值
    打印 (自动换行) result
```

列表 `mylist1` 的第1项与列表 `mylist2` 的第1项到最后1项逐一比较；`mylist1` 的第2项与 `mylist2` 的第1项到最后1项逐一比较.....，直到 `mylist1` 的**最后1项**与 `mylist2` 的第1项到最后1项逐一比较，最终完成比对。

假如你是初一的一名学生，在同一级目录下建立初一所有班级的各班图片文件夹，如初一（1）班，使用摄像头拍照找出你在那个班级。



谢谢大家

