PyCon China 2020

PyConChina 2020
PYTHON 中国开发者大会 2020

Python For Good

K-12 STEM 教育里的 Python + AI 实践

刘敬韬 / Tommy Jing-Tao Liu

CTO, 可可乐博（深圳）科技有限公司

About

*: STEAM 是 Science, Technology, Engineering, Art, Math 的缩写，该理念在传统的 STEM 教育（国内也称创客／科创教育）上延伸出一个艺术的概念，旨在加强 K-12 学生对于科学、技术、工程、艺术以及数学的能力。

- 全栈工程师 & 设计师，tjtl.io
- 玩了 5 年的嵌入式硬件，从 Arduino ATMega 到 Espressif ESP
- 4 年前开始创业（可可乐博），主要做教育科技产品，专注 STEAM* 教育解决方案的研发，产品为粤港澳大湾区超过 200 所中小学校所使用
- 目前在研发针对中小学科创教育课堂的 Python 和入门级 AI 教学产品

K-12 教育里的 Python + AI 实践

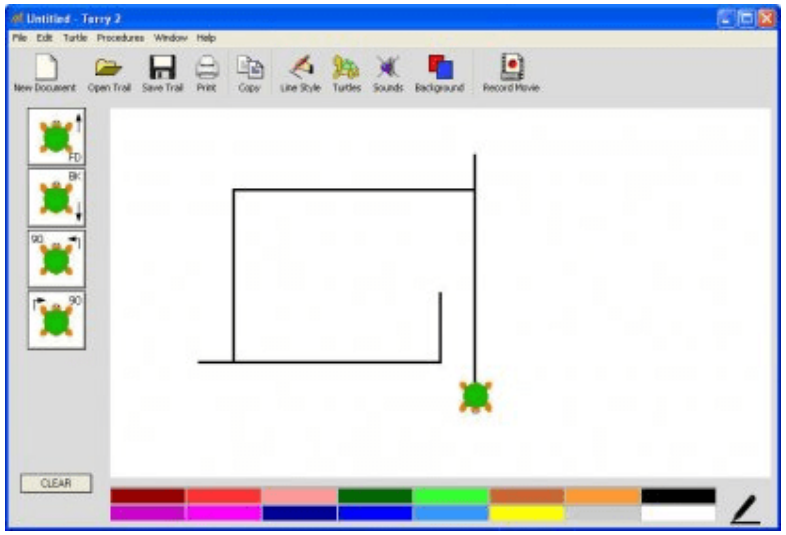
1. 教些什么？
2. 怎么去教？
3. 用什么教？

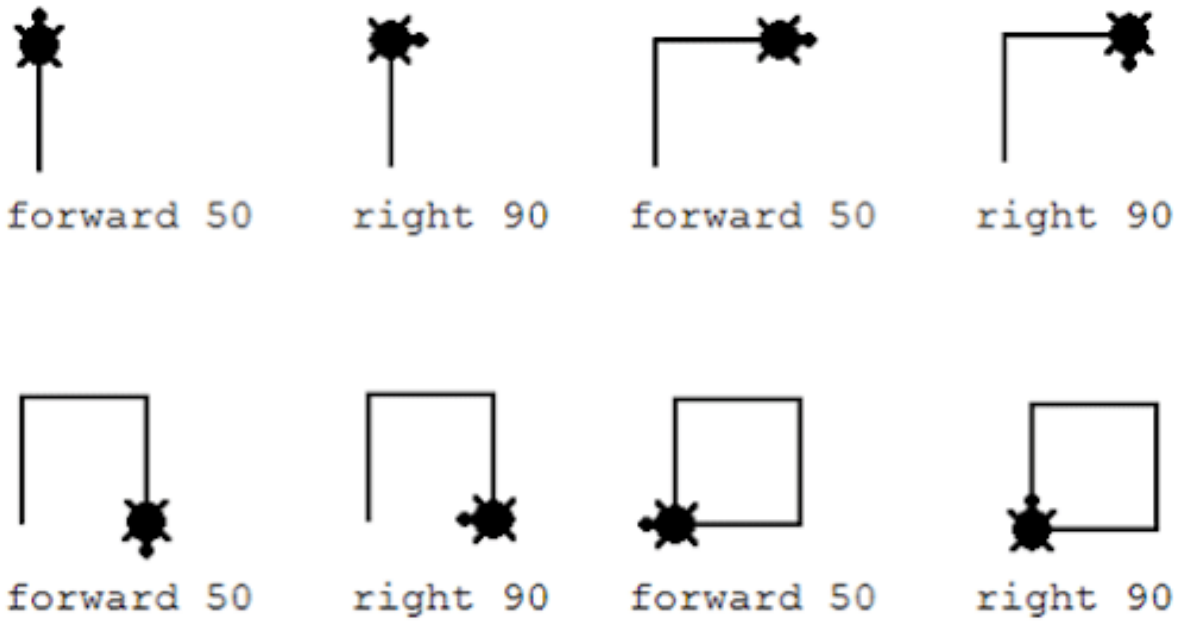
K-12 教育里的 Python + AI 实践

1. 教些什么？
2. 怎么去教？
3. 用什么教？

时代背景

PC Logo





“In the LOGO environment … the child, even at preschool ages, is in control: The child programs the computer” (Papert, 1980, p. 19).

时代背景

index people concepts projects courses events

the aesthetics + computation group

at the mit media laboratory aesthetics + computation group we work toward the design of advanced system architectures and thought processes to enable the creation of (as yet) unimaginable forms and spaces


John Maeda


Tom White

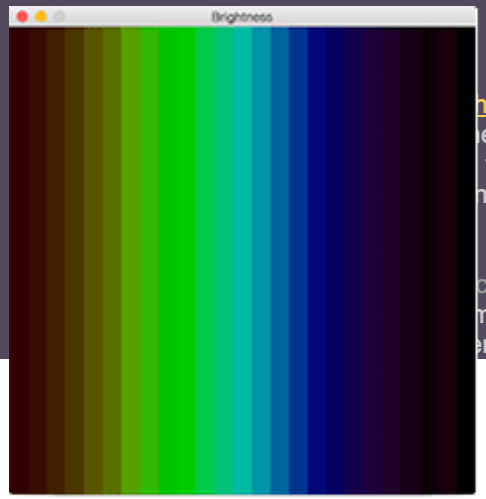

Ben Fry

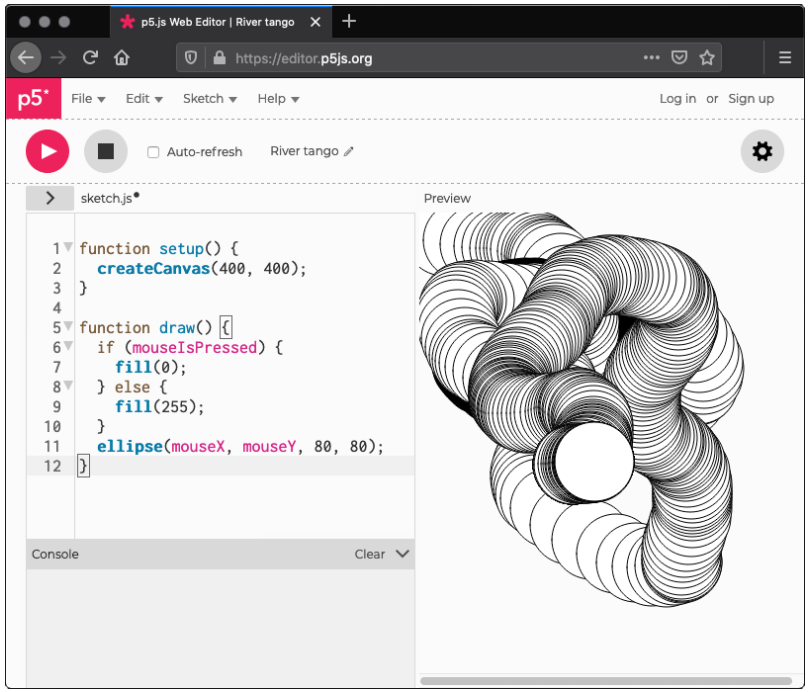

James Seo


Megan Galbraith


Simon Greenwold


Justin Manor

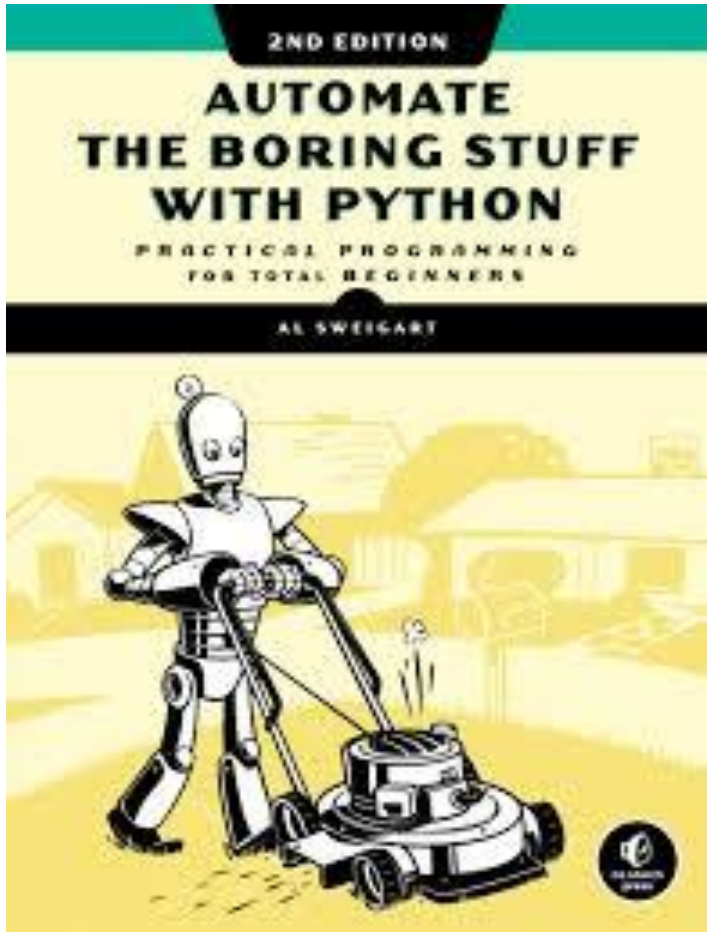




Message Area

Console/Errors

时代背景

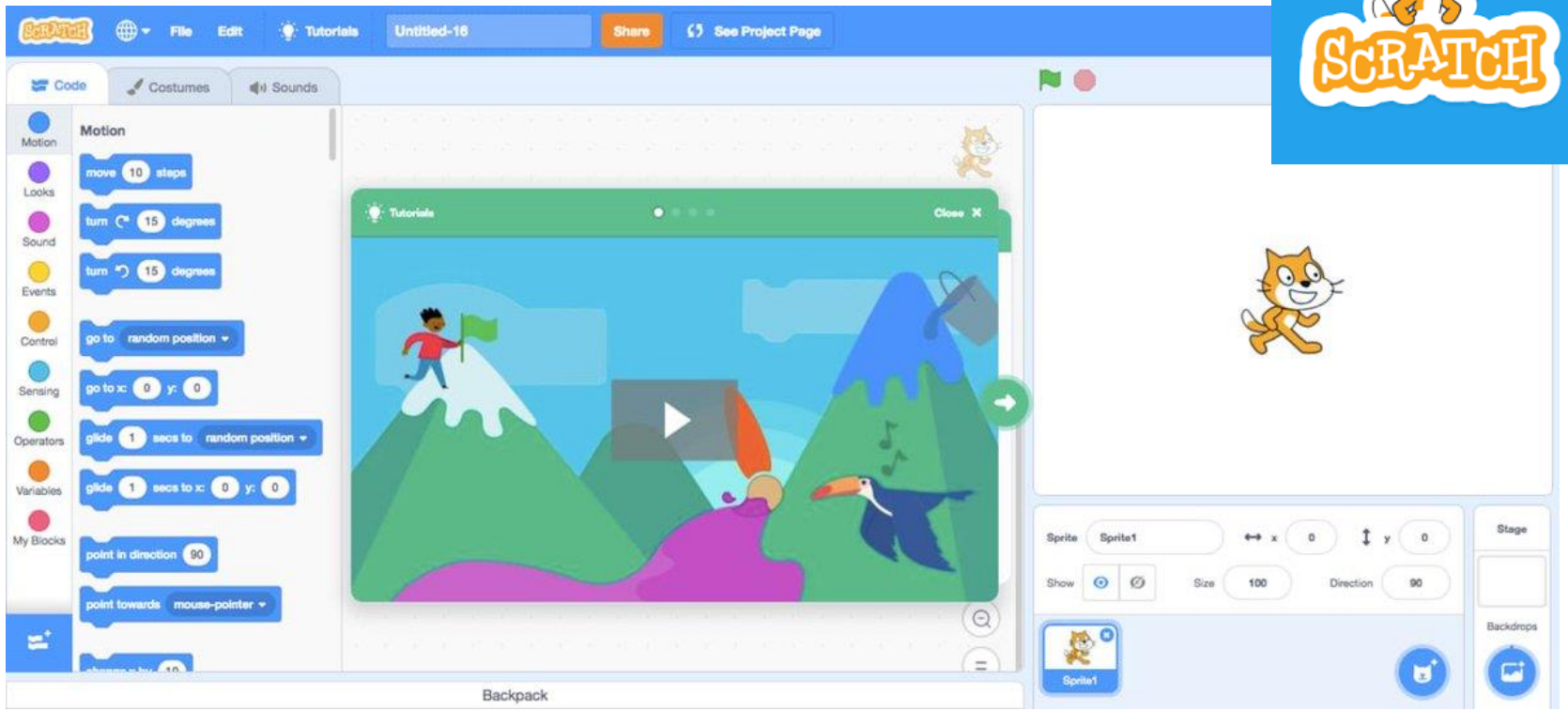


培养学生的逻辑思维能力

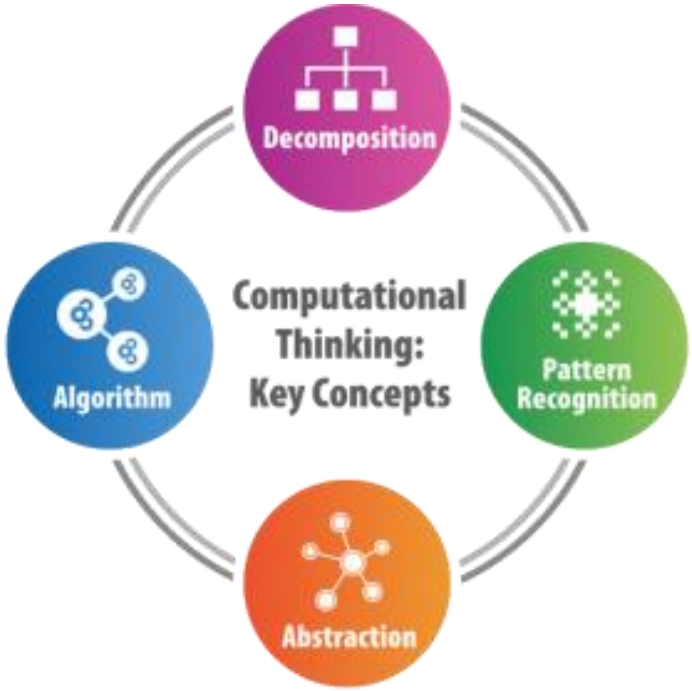
提供一种新的思维工具去解决事情

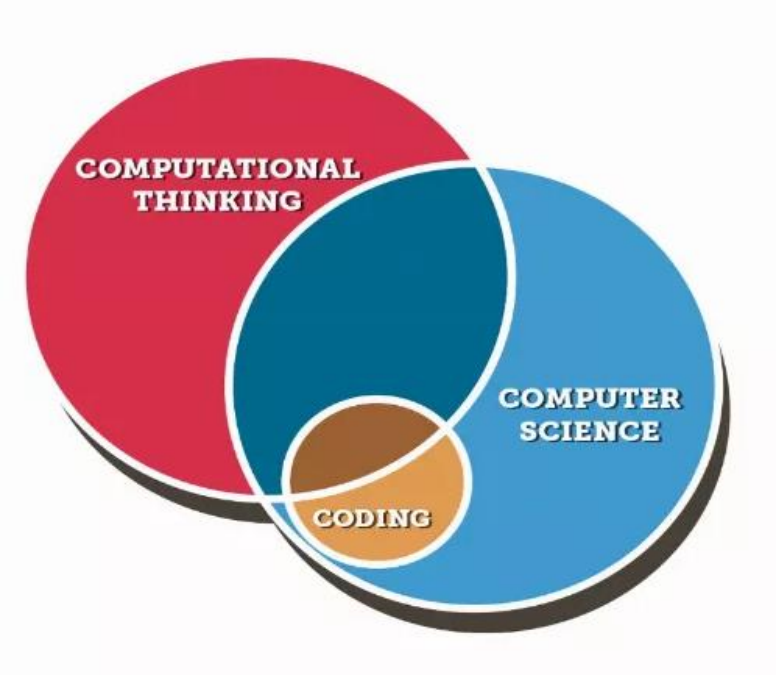
适用于画家、音乐家、建筑师、市场营销等各行各业

时代背景


File
Edit
Tutorials
Untitled-16
Share
See Project Page
Code
Costumes
Sounds
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
move 10 steps
turn 15 degrees
turn 15 degrees
go to random position
go to x: 0 y: 0
glide 1 secs to random position
glide 1 secs to x: 0 y: 0
point in direction 90
point towards mouse-pointer
Tutorials
Close
Backpack
Sprite
Sprite1
x 0
y 0
Show
Size 100
Direction 90
Stage
Backdrops
Sprite1

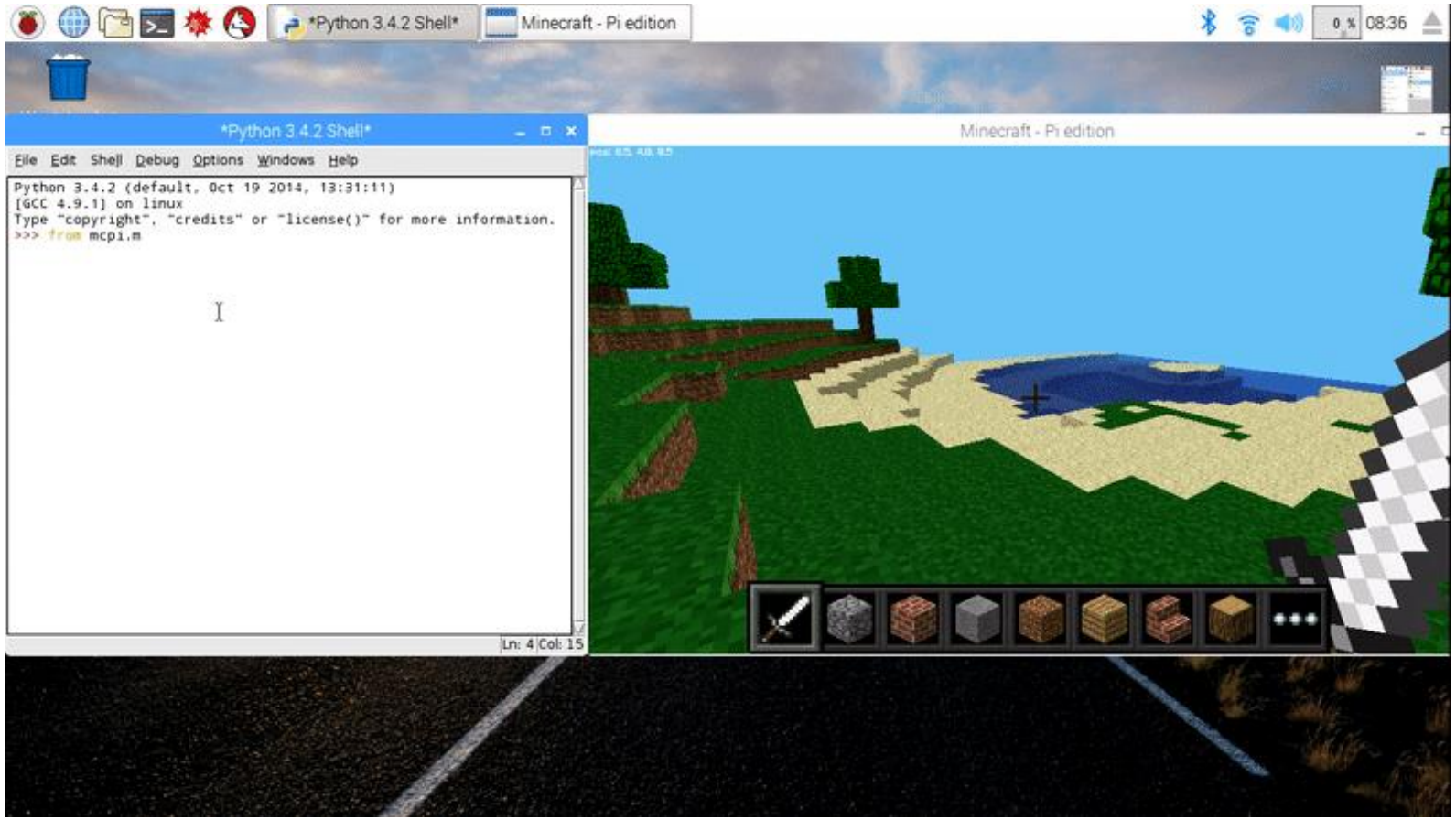
时代背景


Decomposition
Algorithm
Computational Thinking: Key Concepts
Pattern Recognition
Abstraction


COMPUTATIONAL THINKING
COMPUTER SCIENCE
CODING

教些什么：游戏化学习

MineCraft
脚本**编辑**器



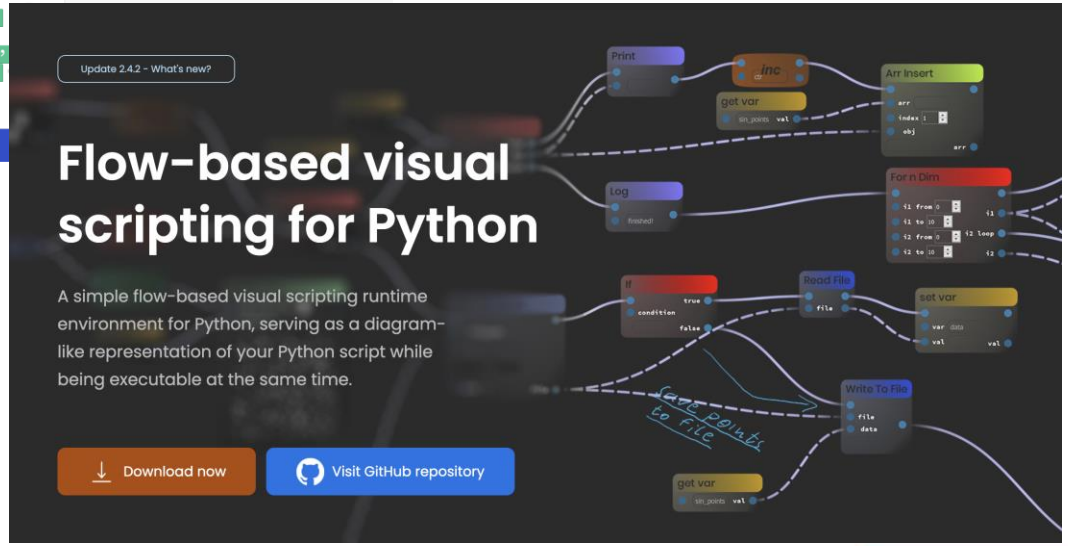
教些什么：游戏化学习

教些什么：图形化编程

CocoBlockly X

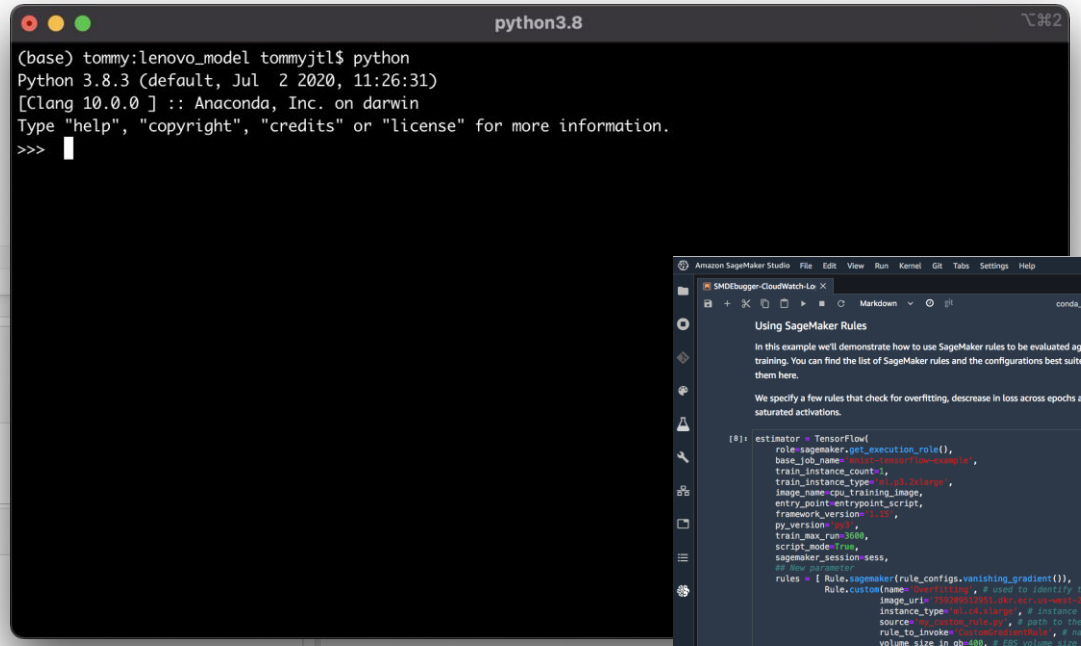
海龟编辑器

Ryven

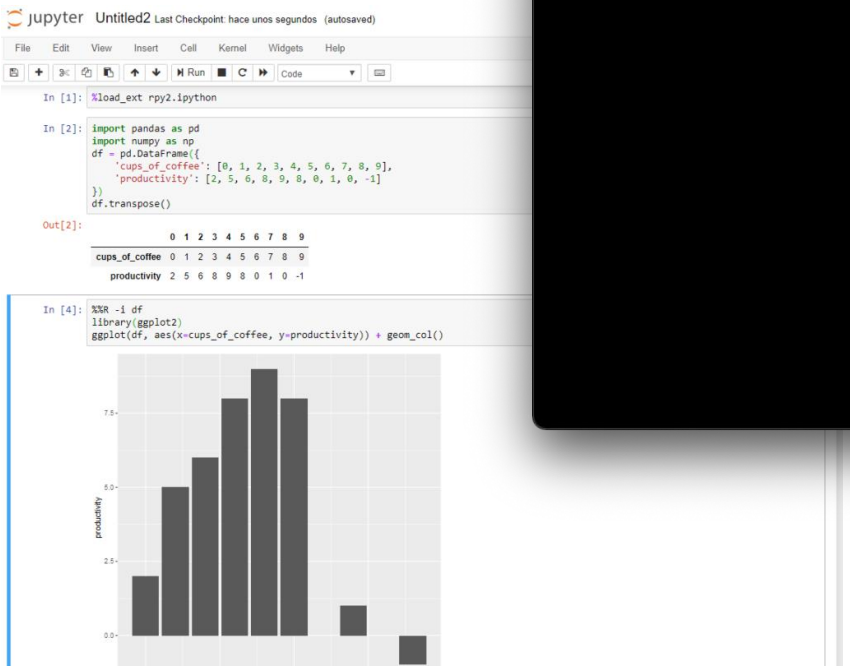


教些什么：源码式编程

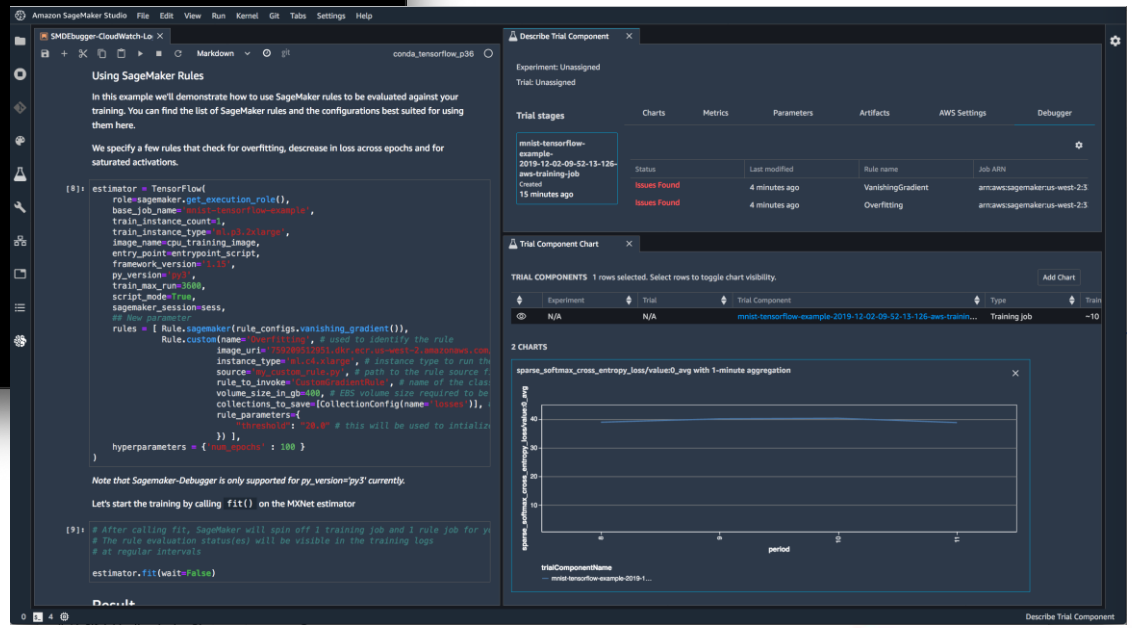
Python Official Build



Jupyter Notebook



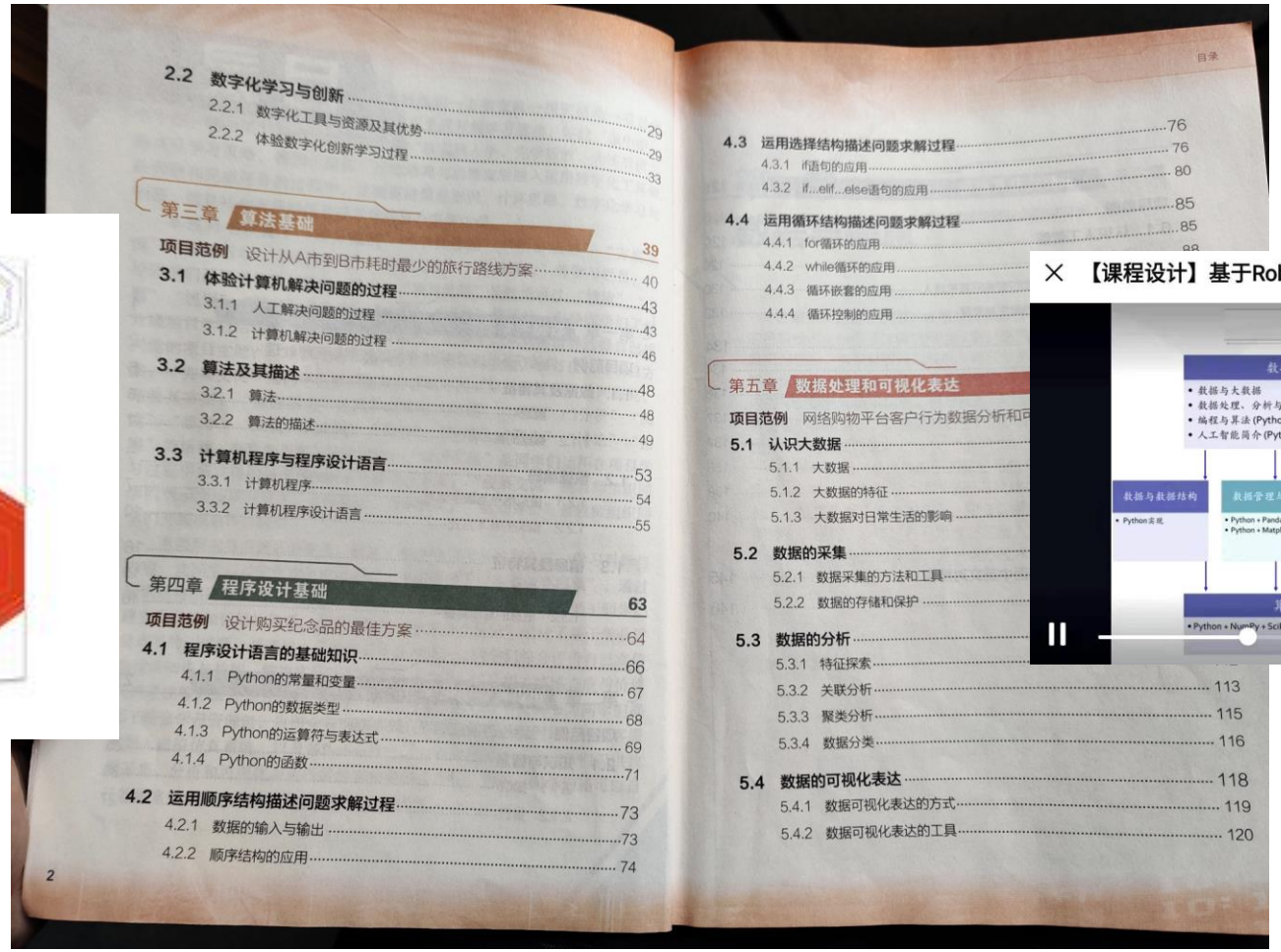
AWS SageMaker Studio

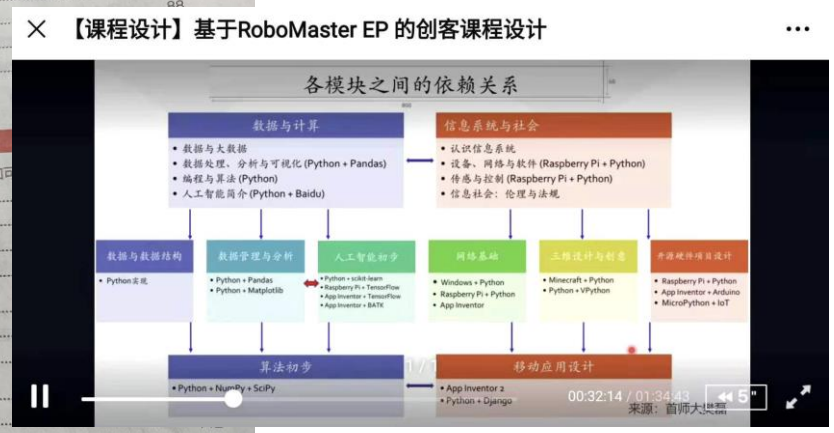


K-12 教育里的 Python + AI 实践

1. 教些什么？
2. 怎么去教？
3. 用什么教？

怎么教？(Python & AI)





怎么教？(Python & AI)

小学阶段

体验

通过交互装置、故事、漫画或者游戏的形式，认识编程和人工智能是什么

调参

通过调整已完成项目的某个参数，让学生能够进一步了解程序，有参与感

理解

能够理解程序逻辑，能够编写简单的程序，能够进行创作

中学阶段

进阶

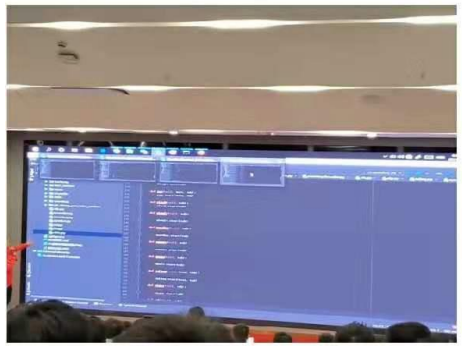
巩固基础知识，学习第三方库、数据分析、爬虫等进阶应用手段

自己玩

对于那些热爱计算机的学生，自己徒手生成 ML 模型、进行大数据分析也不成问题

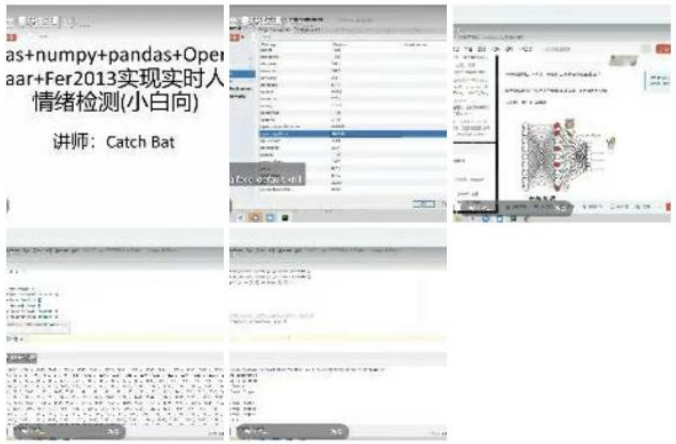
教些什么：怎么教？(Python & AI)

13岁的初中学生可以特别6地用tensorflow搞个小ai，未来少儿编程(api调用师)会像语文数学一样成为基础必修课



1小时前

中学生该如何学习人工智能，昨晚听了一位学生自己组织的技术分享会，完整展示了项目思路，理论讲解，代码实现等，做了一会AI小专家，非常棒。应该鼓励学生多角度思考问题，找到自己感兴趣的实践项目；试着自己从头写算法，思考算法能产生什么价值，兴趣才是最好的老师👍



时代背景

1. （持续）降低编程门槛，让更多的人可以以低成本的方式学习编程（Python）
2. 把软硬件产品和课程高度集成，以学生和老师体验为优先

K-12 教育里的 Python + AI 实践

1. 教些什么？
2. 怎么去教？
3. 用什么教？

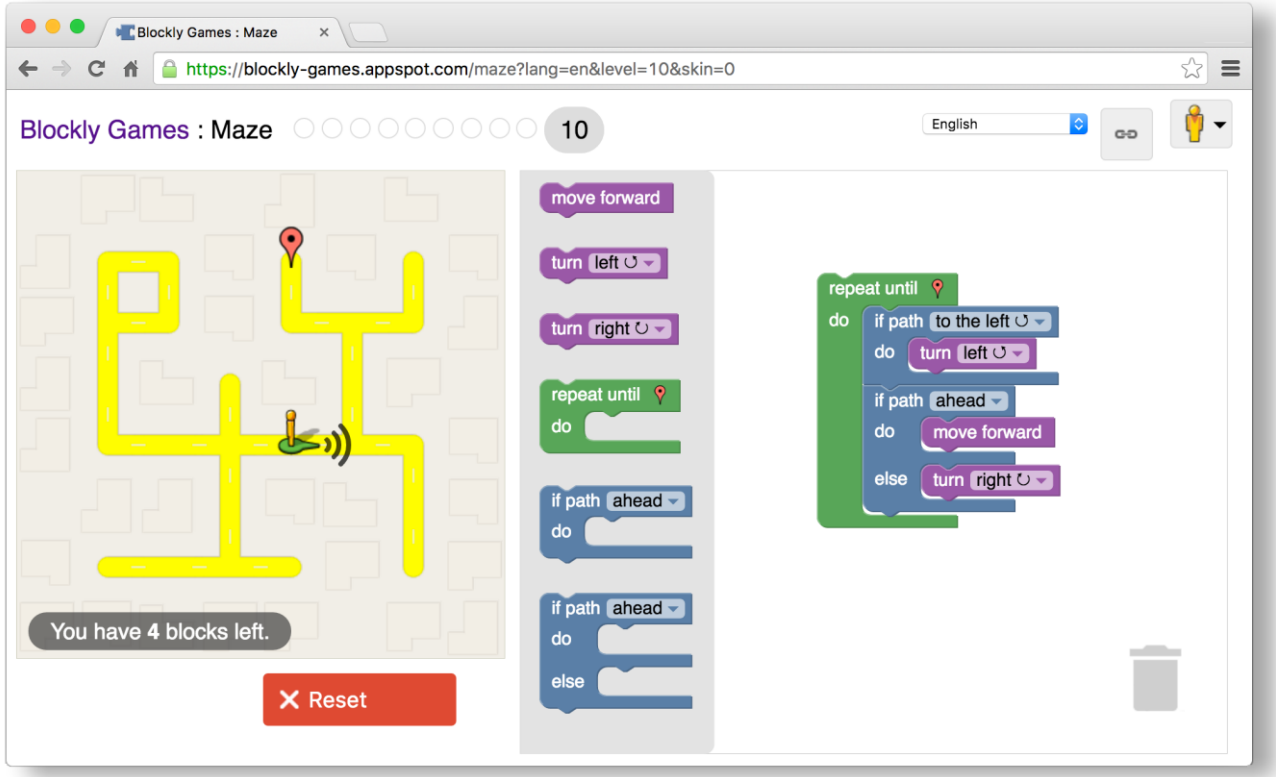
Agenda

- Python + Blockly = **积木式编程**
- Python + Embedded System = **电子编程教学**
- Python + Model Training = **人工智能教学**

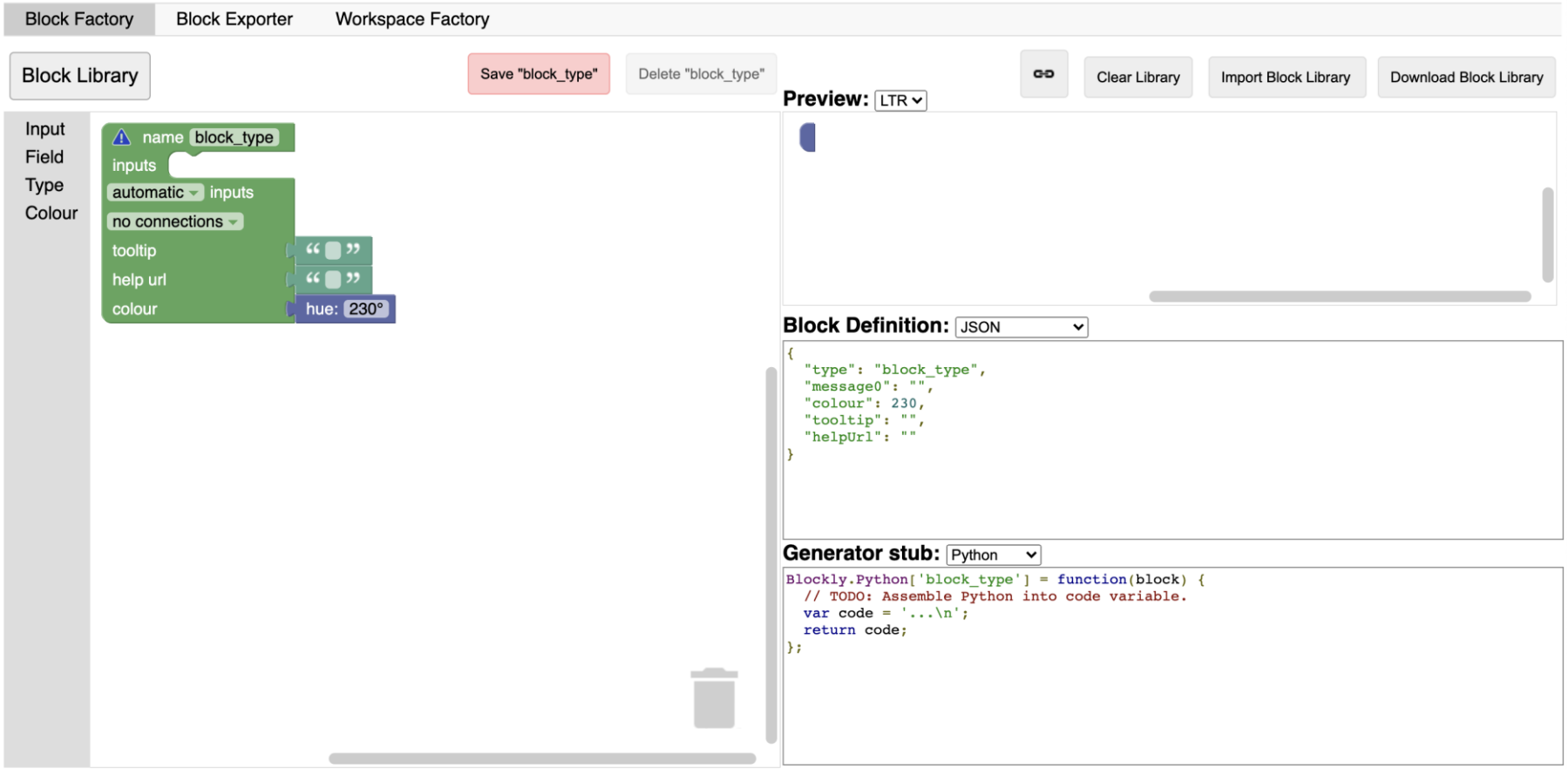
Agenda

- Python + Blockly = **积木式编程**
- Python + Embedded System = **电子&编程教学**
- Python + Model Training = **人工智能教学**

Google Blockly



Google Blockly

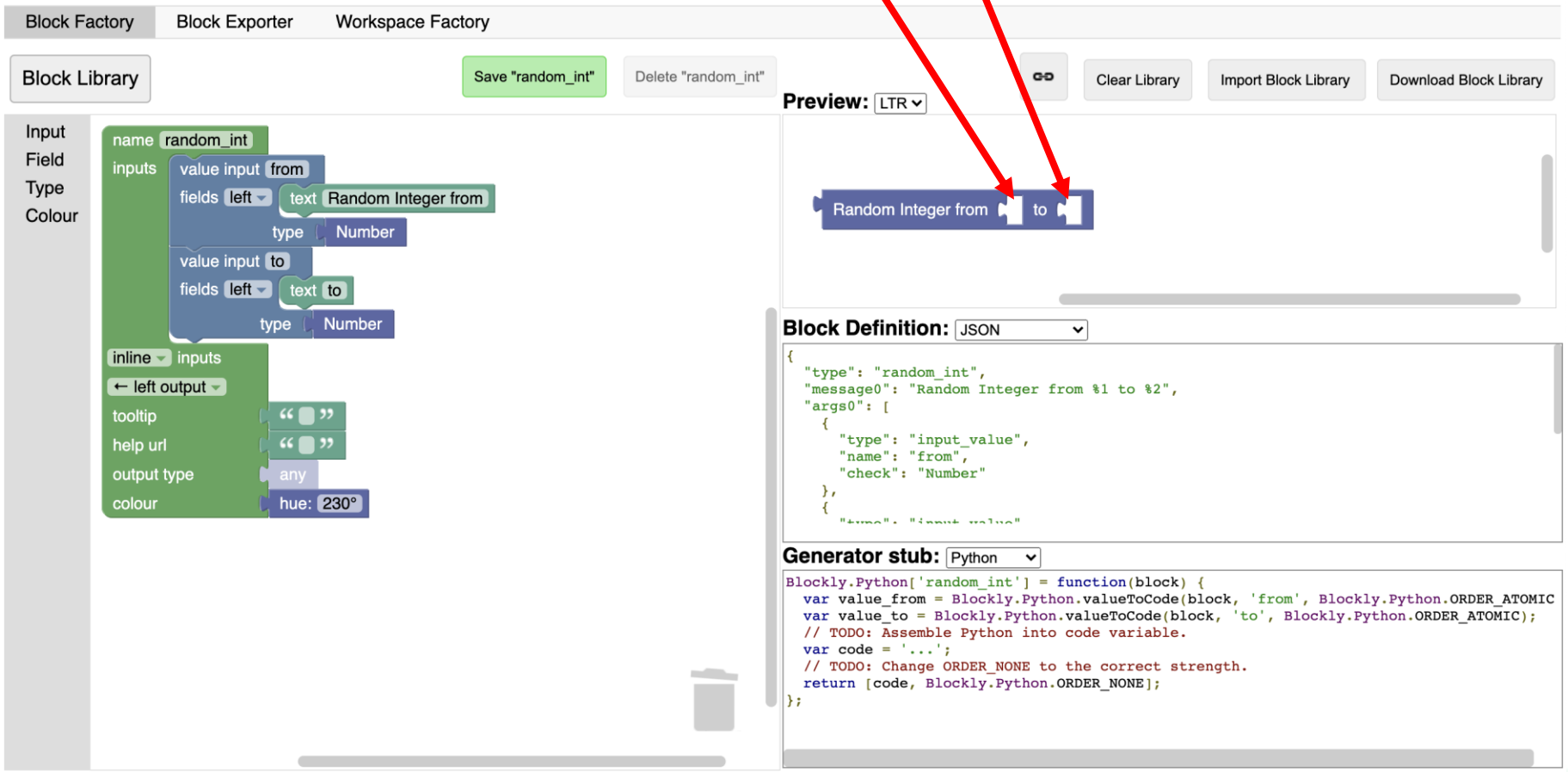


Google Blockly

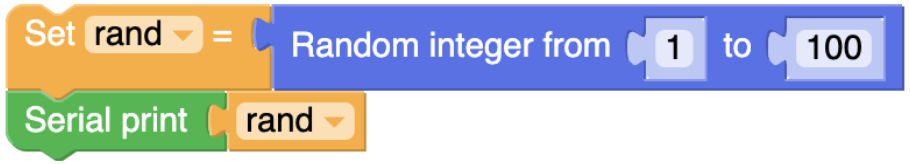

```
random.randint(1, 100)
```

Google Blockly


```
random.randint(1, 100)
```

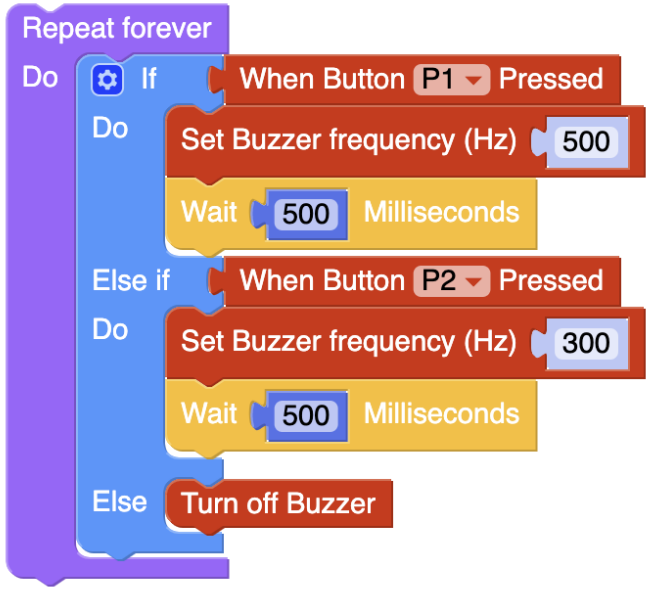


Google Blockly




```
1 import random
2
3
4
5 rand = random.randint(1, 100)
6 print(rand)
7
```

组合以后是怎么样的？



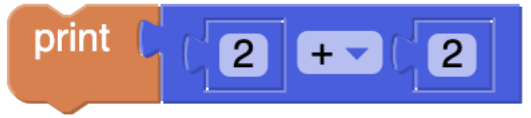
Code -> Block

```
14 while True:
15     if (_P1.value() == 1):
16         beeper.init()
17         beeper.freq(500)
18         time.sleep_ms(500)
19     elif (_P2.value() == 1):
20         beeper.init()
21         beeper.freq(300)
22         time.sleep_ms(500)
23     else:
24         beeper.deinit()
25
```

Google Blockly


```
1 print(2+2)
```

?



能不能把代码转成积木？

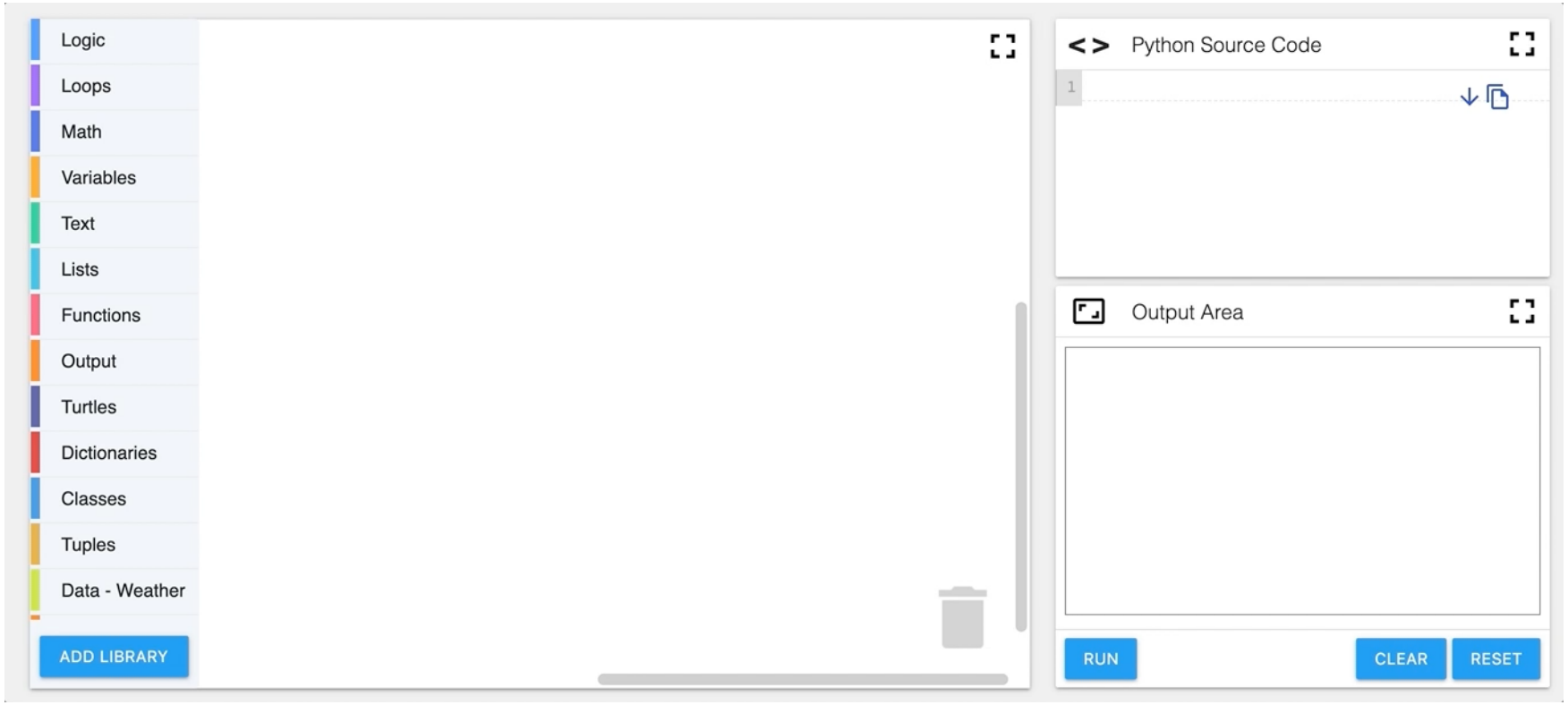
Google Blockly


```
1 print(2+2)
```

可以 ✅

能不能把代码转成积木？

Google Blockly



Google Blockly

如何提供真实的 Python 运行环境给用户使用／学习

服务端
（Python 沙盒进程）

纯前端
（Skulpt, Pydiode）

Google Blockly

纯前端

(Skulpt, Pydiode)

Google Blockly

无需服务端，无需沙盒，

在前端直接模拟 Python 3 环境

SKULPT

Skulpt is an *entirely in-browser* implementation of Python.

No preprocessing, plugins, or server-side support required, just write Python and reload.

PYODIDE
WA

Pyodide brings the Python 3.8 runtime to the browser via WebAssembly, along with the Python scientific stack including NumPy, Pandas, Matplotlib, parts of SciPy, and NetworkX.

The packages directory lists over 35 packages which are currently available.

Google Blockly

JavaScript		Python	Example
String	⟷	str	"Hello, Pyodide"
Uint8ClampedArray	⟷	bytes	"\xff\xf7"
Number	⟵	int	42
	⟶	float	3.1415926
Array	⟷	list	["first", "second"]
Object	⟵	dict	{"key": "value"}
	⟶	jsproxy	document.getElementById()
pyproxy	⟵	object	obj.do_something()
TypedArray	⟷	numpy.ndarray	2x2x2 array of int

https://hacks.mozilla.org/2019/04/pyodide-bringing-the-scientific-python-stack-to-the-browser/

为什么要做积木式编程？

Python 不是已经很简单了吗？

Google Blockly

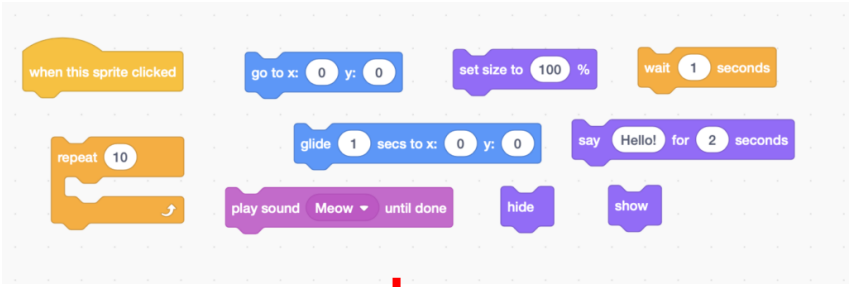

```
import pygame
```

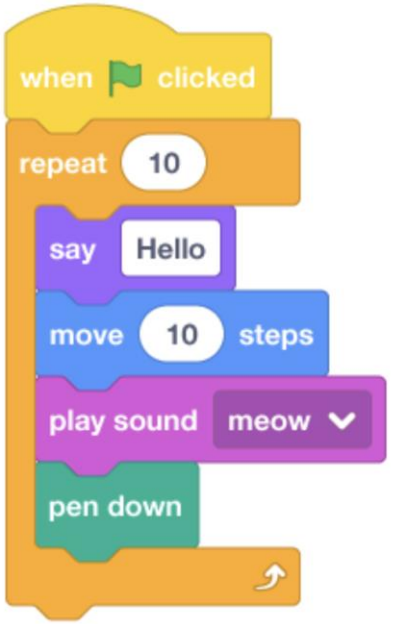
```
theApp = App()
```

```
pygame.display.set_mode
```

```
pygame.event.get()
```

```
import pygame
from pygame.locals import *

class App:
    def __init__(self):
        self._running = True
        self._display_surf = None
        self.size = self.weight, self.height = 640, 400

    def on_init(self):
        pygame.init()
        self._display_surf = pygame.display.set_mode(self.size
        self._running = True

    def on_event(self, event):
        if event.type == pygame.QUIT:
            self._running = False
    def on_loop(self):
        pass
    def on_render(self):
        pass
    def on_cleanup(self):
        pygame.quit()

    def on_execute(self):
        if self.on_init() == False:
            self._running = False

        while( self._running ):
            for event in pygame.event.get():
                self.on_event(event)
            self.on_loop()
            self.on_render()
        self.on_cleanup()

if __name__ == "__main__" :
    theApp = App()
    theApp.on_execute()
```



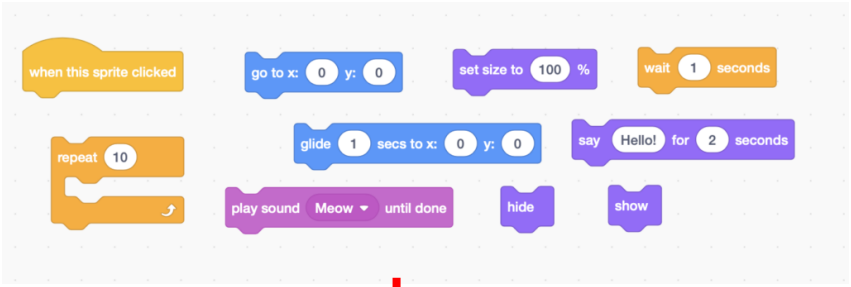


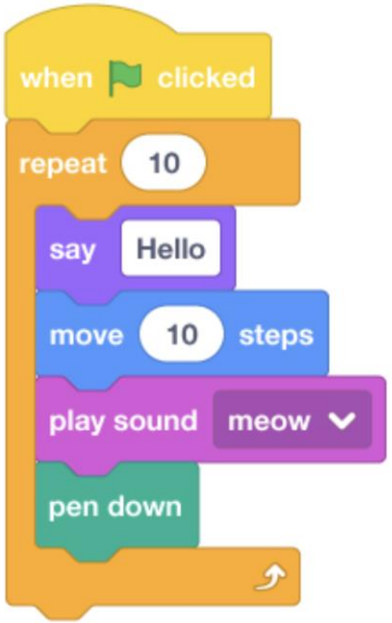
Google Blockly


```
import pygame
theApp = App()
pygame.display.set_mode
pygame.event.get()
```

```
import pygame
from pygame.locals import *

class App:
    def __init__(self):
        self._running = True
        self._display_surf = None
        self.size = self.weight, self.height = 640, 400
        self._display_surf = pygame.display.set_mode(self.size
        self._running = True
    def on_event(self, event):
        if event.type == pygame.QUIT:
            self._running = False
    def on_loop(self):
        pass
    def on_render(self):
        pass
    def on_cleanup(self):
        pygame.quit()

    def on_execute(self):
        if self.on_init() == False:
            self._running = False

        while( self._running ):
            for event in pygame.event.get():
                self.on_event(event)
            self.on_loop()
            self.on_render()
        self.on_cleanup()

if __name__ == "__main__" :
    theApp = App()
    theApp.on_execute()
```

不适合刚入门的用户学习



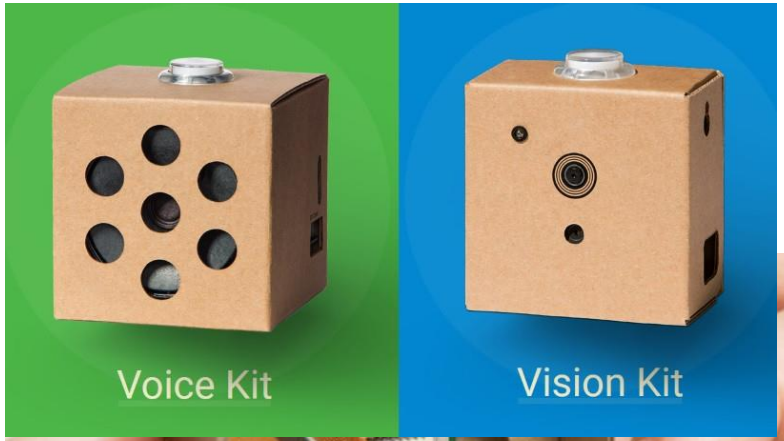


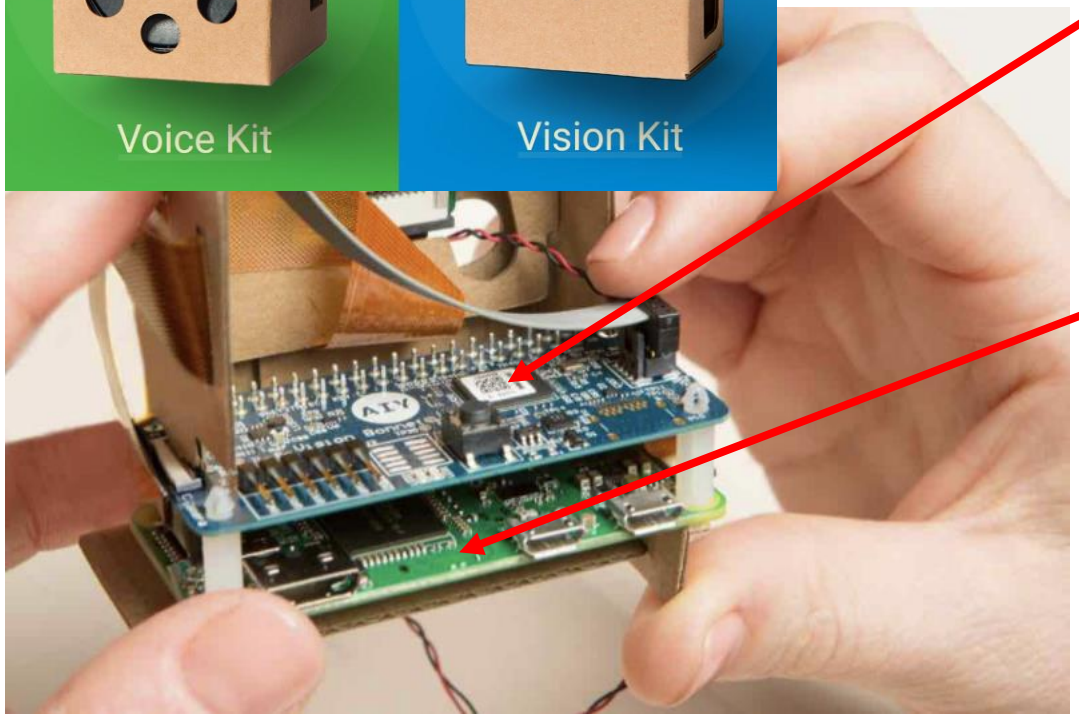
Google Blockly

基础的 Python 可以积木化,

Python + A.I. 能否也积木化？

Google AIY





Intel Movidius MA2450 processor

low-energy artificial neural network programming, with 30FPS processing image speed

Raspberry Pi Zero

CPU	1-GHZ, Broadcom BCM2835
RAM	512MB
Wireless (Pi Zero W only)	802.11n / Bluetooth 4.1 / LE
Ports	Micro USB, mini HDMI
I/O	40 GPIO Pins, CSI camera connector (not on version 1.2)

Google AIY

aiyprojects ▾ aiyprojects-raspbian / src / examples / vision / Go to file Add file ▾

dmitriykovalev Make joy demo sound playing more robust ... b383a27 5 days ago History

..		
joy	Make joy demo sound playing more robust	5 days ago
object_meter	Fix lint errors.	2 years ago
video_capture	Add more unit tests and verify all of them pass.	2 years ago
any_model_camera.py	Make all vision example scripts executable.	2 years ago
dish_classification.py	Cleanup vision examples.	2 years ago
dish_detection.py	Make all vision example scripts executable.	2 years ago
face_camera_trigger.py	Cleanup vision examples.	2 years ago
face_detection.py	Cleanup vision examples.	2 years ago
face_detection_camera.py	Cleanup vision examples.	2 years ago
face_detection_raspivid.py	Add count property to CameraInference class + cleanup.	3 years ago
image_classification.py	Cleanup vision examples.	2 years ago
image_classification_camera.py	Fix vision lint errors.	2 years ago
inaturalist_classification.py	Make all vision example scripts executable.	2 years ago
mobilenet_based_classifier.py	Cleanup vision examples.	2 years ago
object_detection.py	Cleanup vision examples.	2 years ago

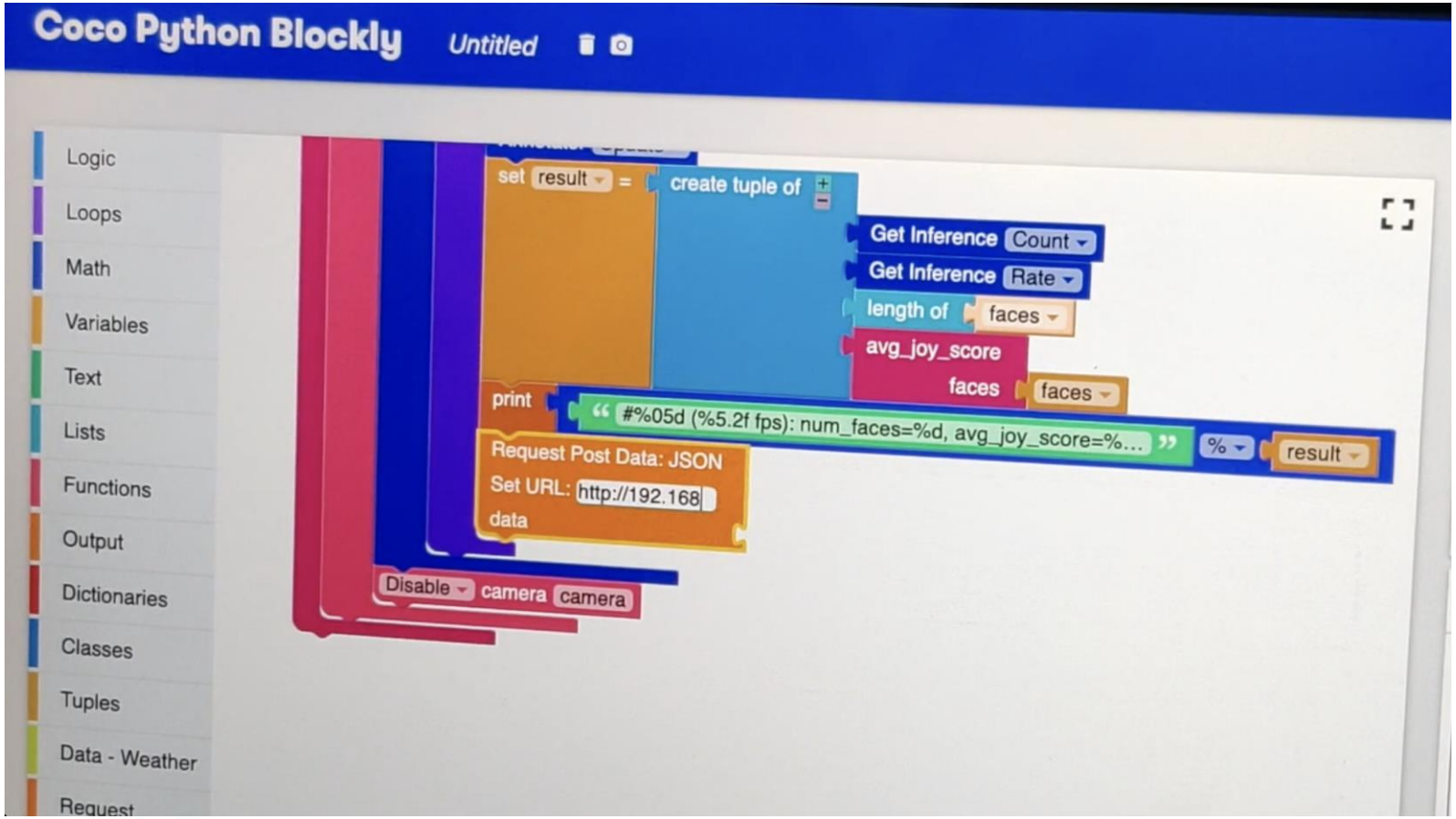
Google AIY


```
1   import argparse
2
3   from picamera import PiCamera
4
5   from aiy.vision.inference import CameraInference
6   from aiy.vision.models import face_detection
7   from aiy.vision.annotator import Annotator
```

...

```
41          with CameraInference(face_detection.model()) as inference:
42              for result in inference.run(args.num_frames):
43                  faces = face_detection.get_faces(result)
44                  annotator.clear()
45                  for face in faces:
46                      annotator.bounding_box(transform(face.bounding_box), fill=0)
47                  annotator.update()
48
49                  print('#%05d (%5.2f fps): num_faces=%d, avg_joy_score=%.2f' %
50                      (inference.count, inference.rate, len(faces), avg_joy_score(faces)))
```

Google AIY


Coco Python Blockly
Untitled
Logic
Loops
Math
Variables
Text
Lists
Functions
Output
Dictionaries
Classes
Tuples
Data - Weather
Request
set result =
create tuple of
Get Inference Count
Get Inference Rate
length of faces
avg_joy_score
faces faces
print
“ #%05d (%5.2f fps): num_faces=%d, avg_joy_score=%... ”
% result
Request Post Data: JSON
Set URL: http://192.168
data
Disable camera camera

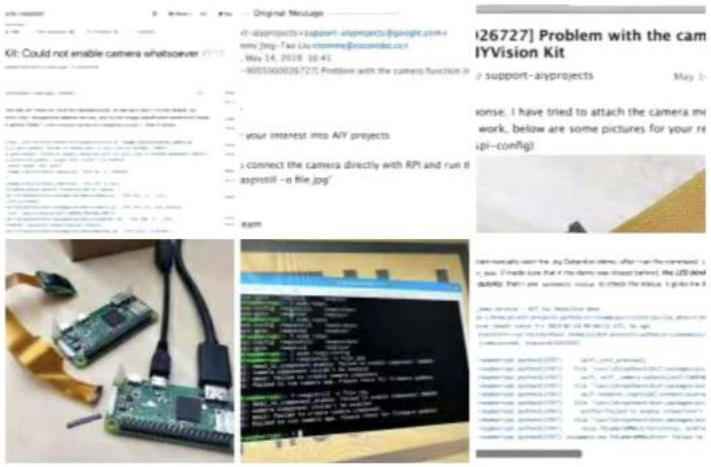
Google AIY

一切都看上去不错，但是…

Google AIY

Tommy

從來沒有對 Google 如此失望過，都已經推出了兩代原型，仍然有一堆不穩定的軟硬件情況：座子沒焊齊、排線座固定片稍一用力就斷、攝像頭一直 access 不到，更別提最初拿到第一個 vision kit 時，Raspbian 剛開機只出現彈窗，沒有 GUI，還得重新 flash 才正常，太震驚了。



2019年5月14日 20:23 删除

毕竟还是一个

开源 & 实验 & 针对开发者的项目，

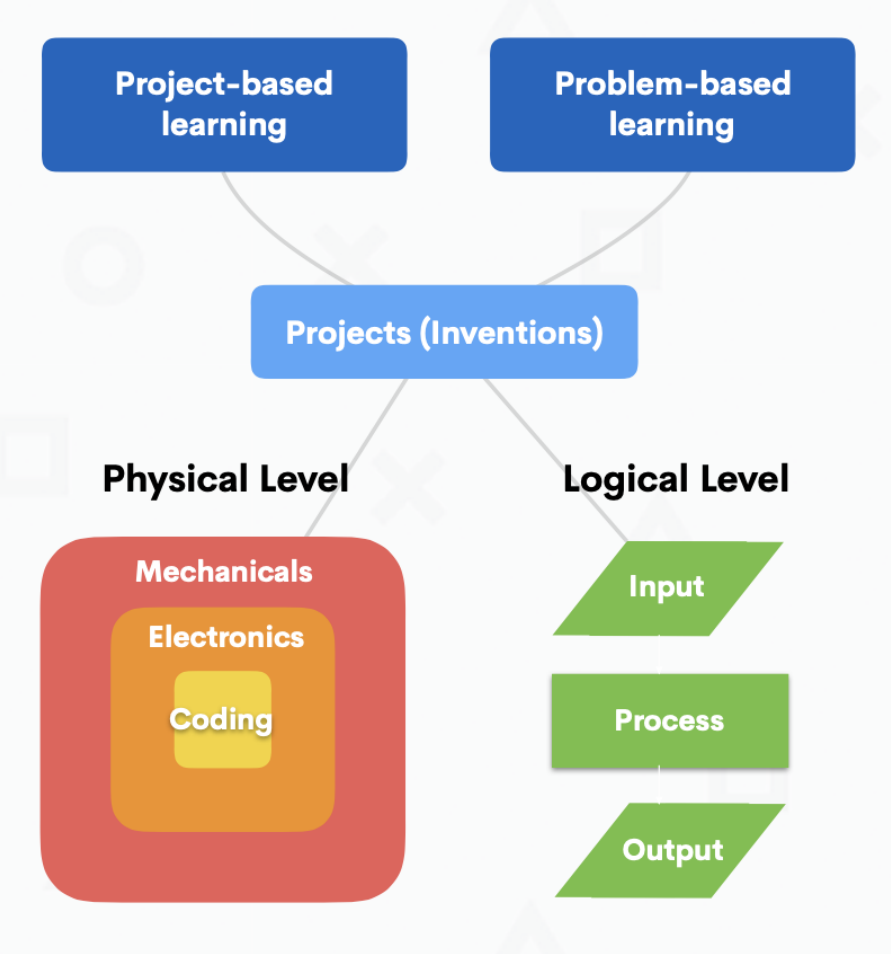
所以我们无法将其实际应用在课堂教学中

所以我们开始自己做硬件

Agenda

- Python + Blockly = 积木式编程
- Python + Embedded System = **电子编程教学**
- Python + Model Training = 人工智能教学

Agenda


Project-based learning
Problem-based learning
Projects (Inventions)
Physical Level
Logical Level
Mechanicals
Electronics
Coding
Input
Process
Output

MicroPython：嵌入式硬件版本 Python

Completely free, open source software

MicroPython is a lean and efficient implementation of the Python 3 programming language that includes a small subset of the Python standard library and is **optimised** to run on microcontrollers and in constrained environments.

MicroPython is packed full of advanced features such as an interactive prompt, arbitrary precision integers, closures, list comprehension, generators, exception handling and more. Yet it is compact enough to fit and run within just 256k of code space and 16k of RAM.

MicroPython

在精简的嵌入式硬件上运行 Python

STM32, SAMD21, ARM, ESP8266, ESP32…

MicroPython

在精简的嵌入式硬件上运行 Python

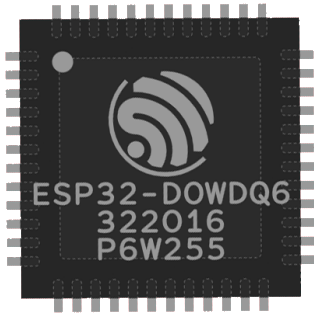
STM32, SAMD21, ARM, ESP8266, ESP32…

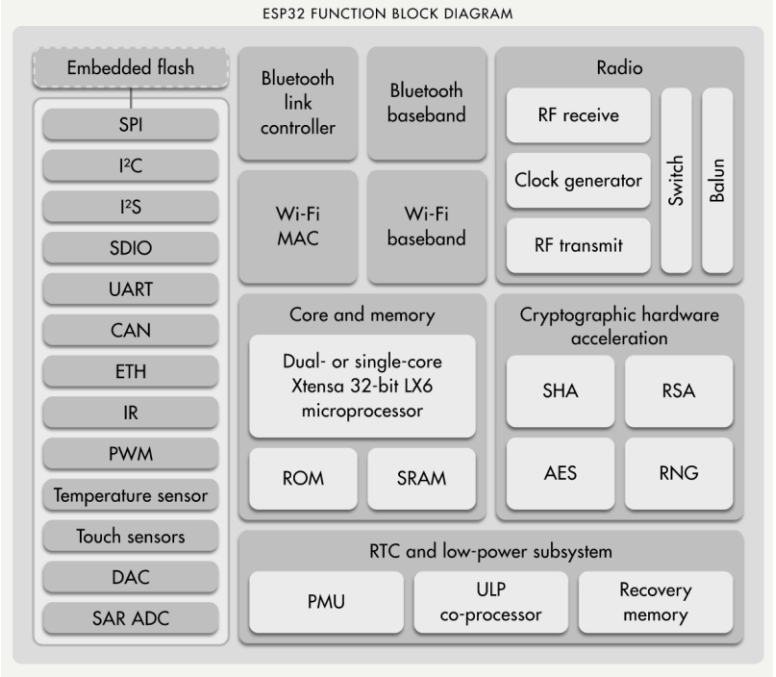
MicroPython

Manufacturer	Espressif Systems
CPU	**Tensilica Xtensa LX6 microprocessor @ 160 or 240**
MHz	
Memory	520 KiB SRAM
Power	**3.3 V DC**

Picture from: http://esp32.net

MicroPython

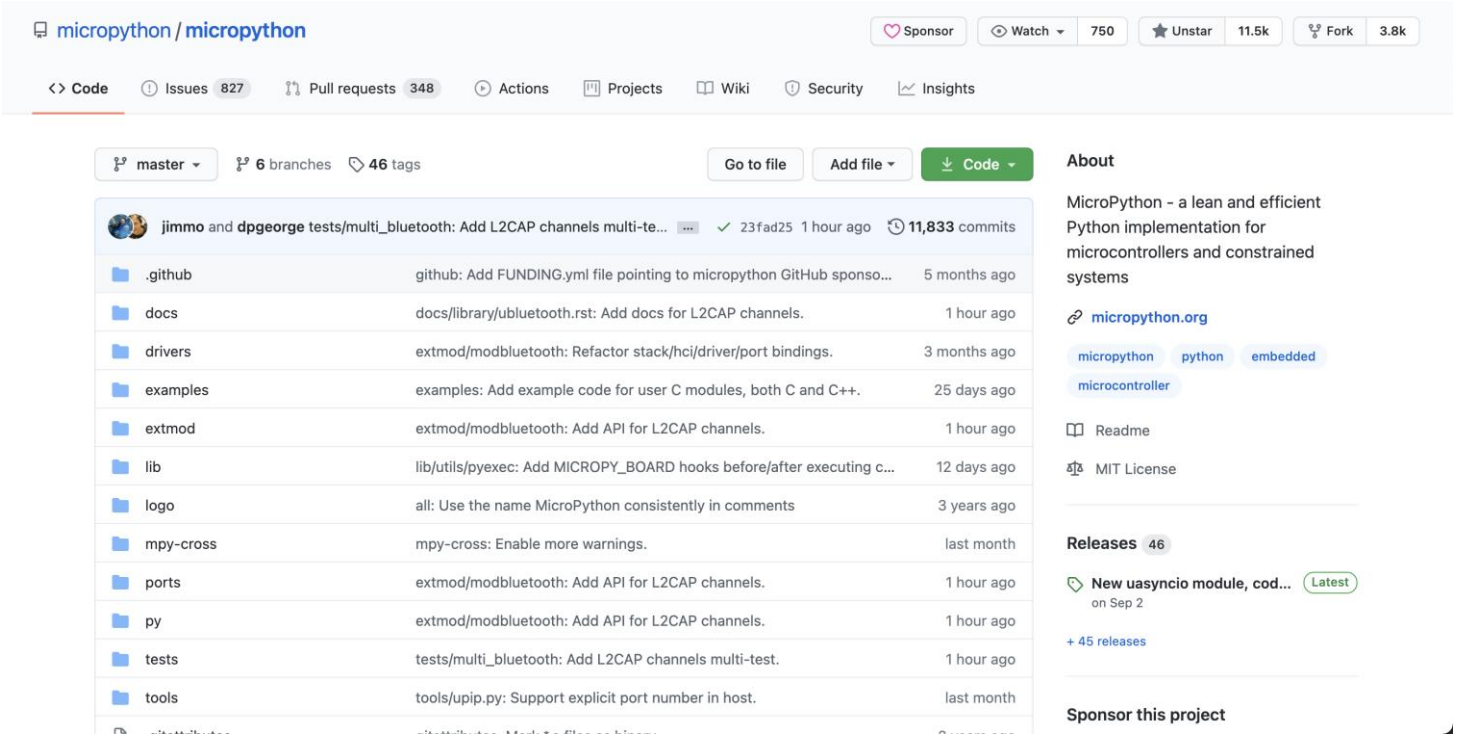




low-cost, low-power
system on a chip microcontrollers
with integrated Wi-Fi and dual-mode Bluetooth.

MicroPython

MicroPython is written in C99, and is highly customizable.



MicroPython

将 MicroPython 的功能封装为积木，

在嵌入式硬件上使用 Python 句法直接编写 Python 程序

MicroPython

Delay and timing

Use the `time` module:

```
import time

time.sleep(1)           # sleep for 1 second
time.sleep_ms(500)      # sleep for 500 milliseconds
time.sleep_us(10)       # sleep for 10 microseconds
start = time.ticks_ms() # get millisecond counter
delta = time.ticks_diff(time.ticks_ms(), start) # compute time difference
```

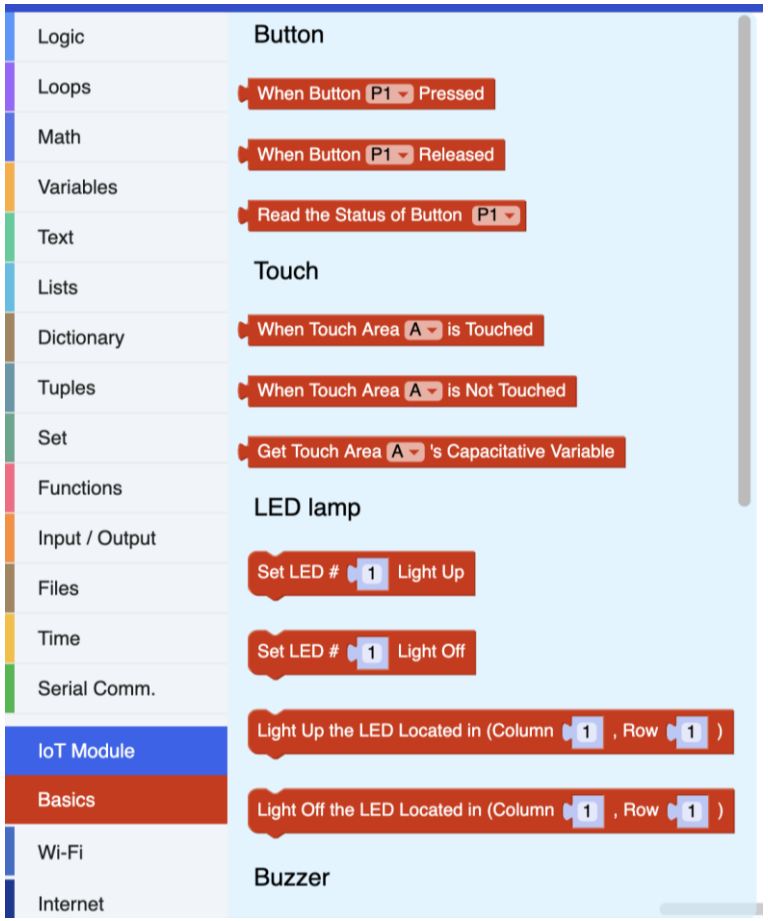
Timers

The ESP32 port has four hardware timers. Use the machine.Timer class with a timer ID from 0 to 3 (inclusive):

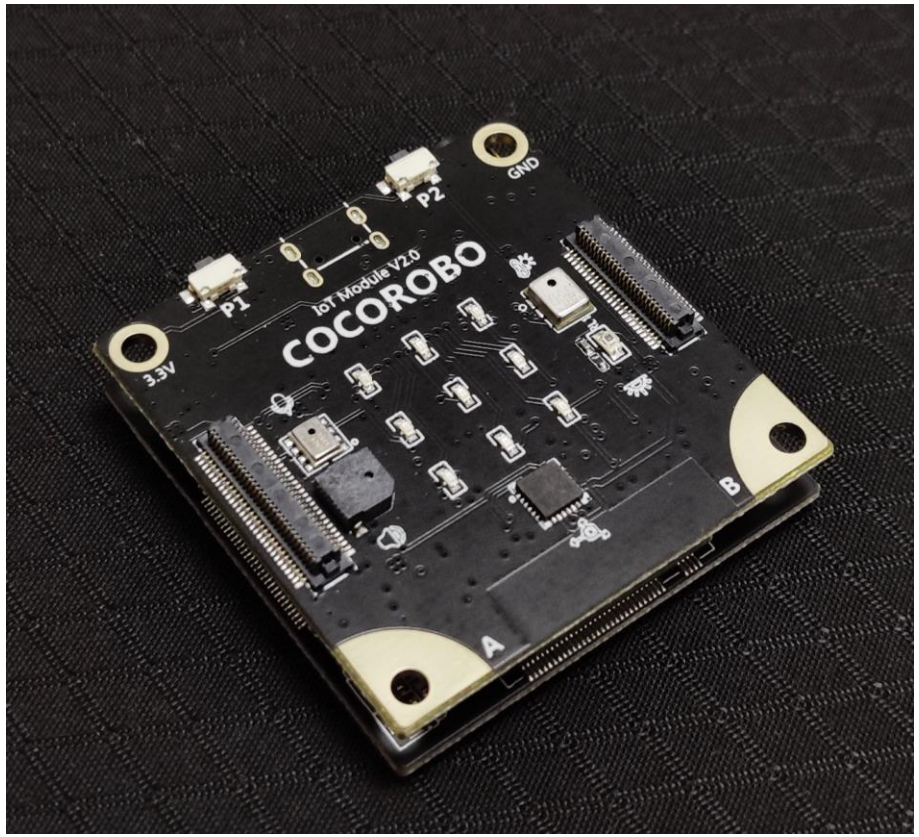
```
from machine import Timer

tim0 = Timer(0)
tim0.init(period=5000, mode=Timer.ONE_SHOT, callback=lambda t:print(0))

tim1 = Timer(1)
tim1.init(period=2000, mode=Timer.PERIODIC, callback=lambda t:print(1))
```

The period is in milliseconds.

Virtual timers are not currently supported on this port.



MicroPython



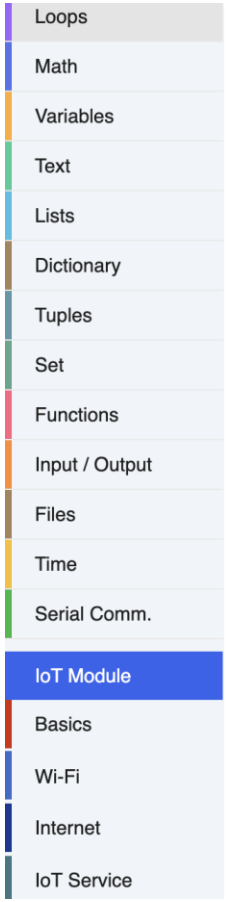
- Based on ESP32 (w/ WiFi & BLE)
- Buttons, LEDs, Buzzer, Microphone
- Capacitive Touch Sensor
- Light Sensitivity Sensor
- Temperature & Humidity Sensor
- Motion Sensor (Gyroscope)

MicroPython

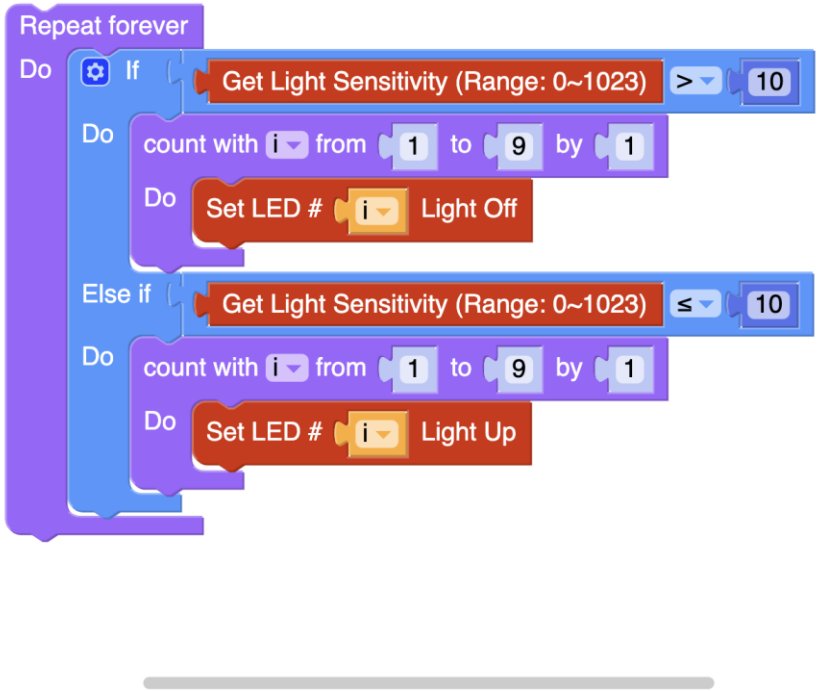
浏览器内上传程序

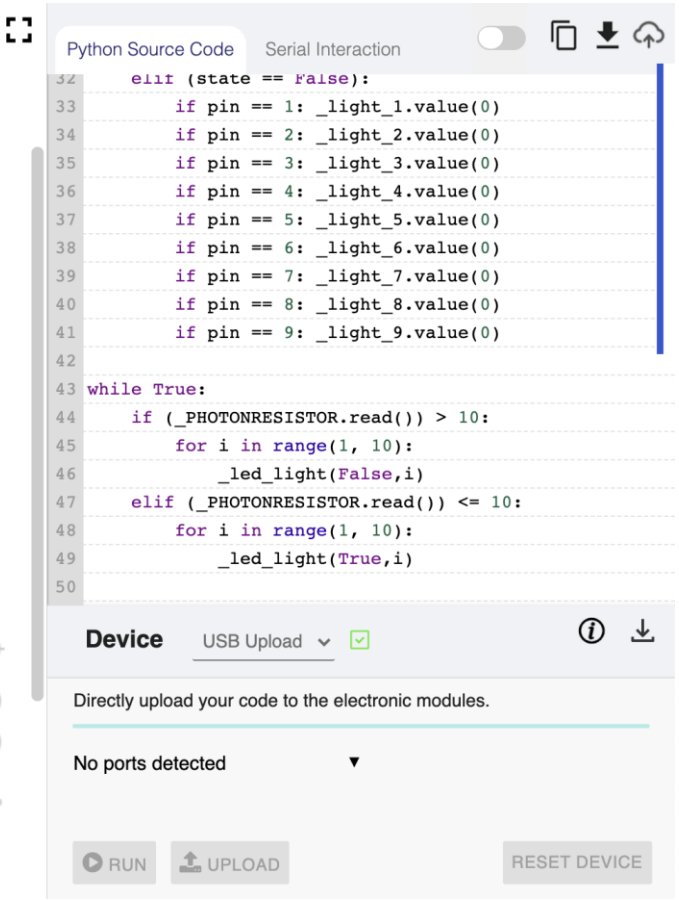

CocoBlockly X BETA
未命名项目
保存
样例
AI 模组
学习
设备
文件
语言
tonyemail2014
逻辑
循环
数学运算
变量
文字
列表
字典
元组
集合
函数
输入 / 输出
文件
时间
串口通信
人工智能
扩展功能
Python 源代码
串口交互窗
设备
有线上传
直接将代码上传到电子模组中。
COM15
更换启动模式
重启设备
CocoRobo LTD © 2020 Copyright
CocoBlockly X Beta

来看几个简单的示例



来看个简单的示例：光照感应灯

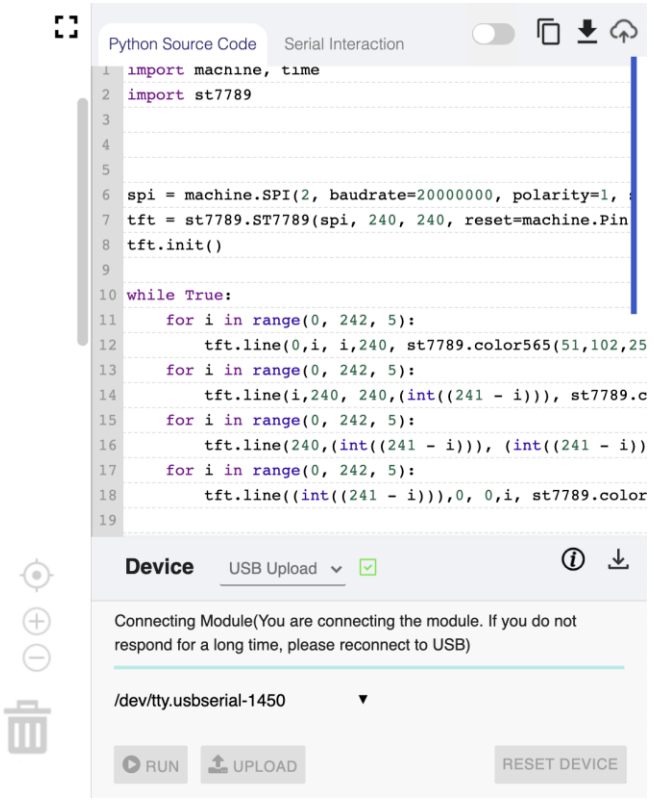




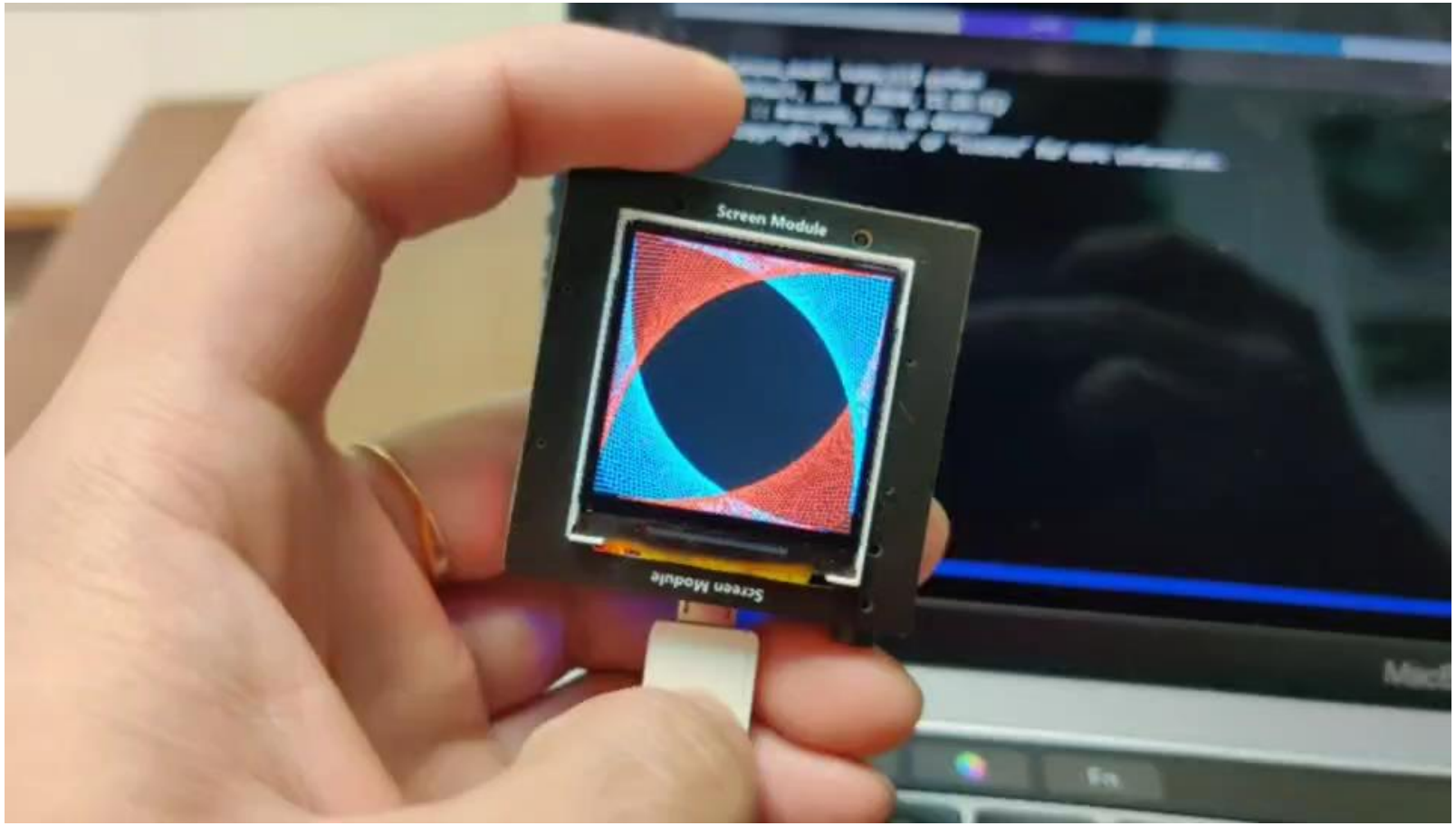
MicroPython

屏幕图案绘制

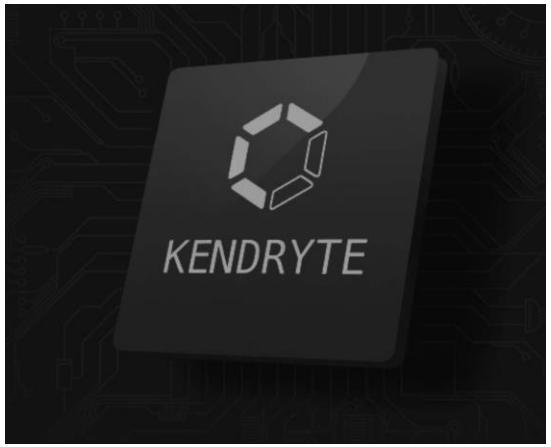

Loops
Math
Variables
Text
Lists
Dictionary
Tuples
Set
Functions
Input / Output
Files
Time
Serial Comm.
IoT Module
Basics
Wi-Fi
Internet
IoT Service
LCD Screen Setup
Repeat forever
Do
count with i from 0 to 241 by 5
Do
Draw Line
Set Start Coordinate: X: 0 Y: i
Set End Coordinate: X: i Y: 240
Color: Colour
count with i from 0 to 241 by 5
Do


Python Source Code
Serial Interaction
import machine, time
import st7789
spi = machine.SPI(2, baudrate=20000000, polarity=1,
tft = st7789.ST7789(spi, 240, 240, reset=machine.Pin
tft.init()
while True:
for i in range(0, 242, 5):
tft.line(0,i, i,240, st7789.color565(51,102,25
for i in range(0, 242, 5):
tft.line(i,240, 240,(int((241 - i))), st7789.c
for i in range(0, 242, 5):
tft.line(240,(int((241 - i))), (int((241 - i))
for i in range(0, 242, 5):
tft.line((int((241 - i))),0, 0,i, st7789.color
Device
USB Upload
Connecting Module(You are connecting the module. If you do not respond for a long time, please reconnect to USB)
/dev/tty.usbserial-1450
RUN
UPLOAD
RESET DEVICE

MicroPython



MaixPy & K210



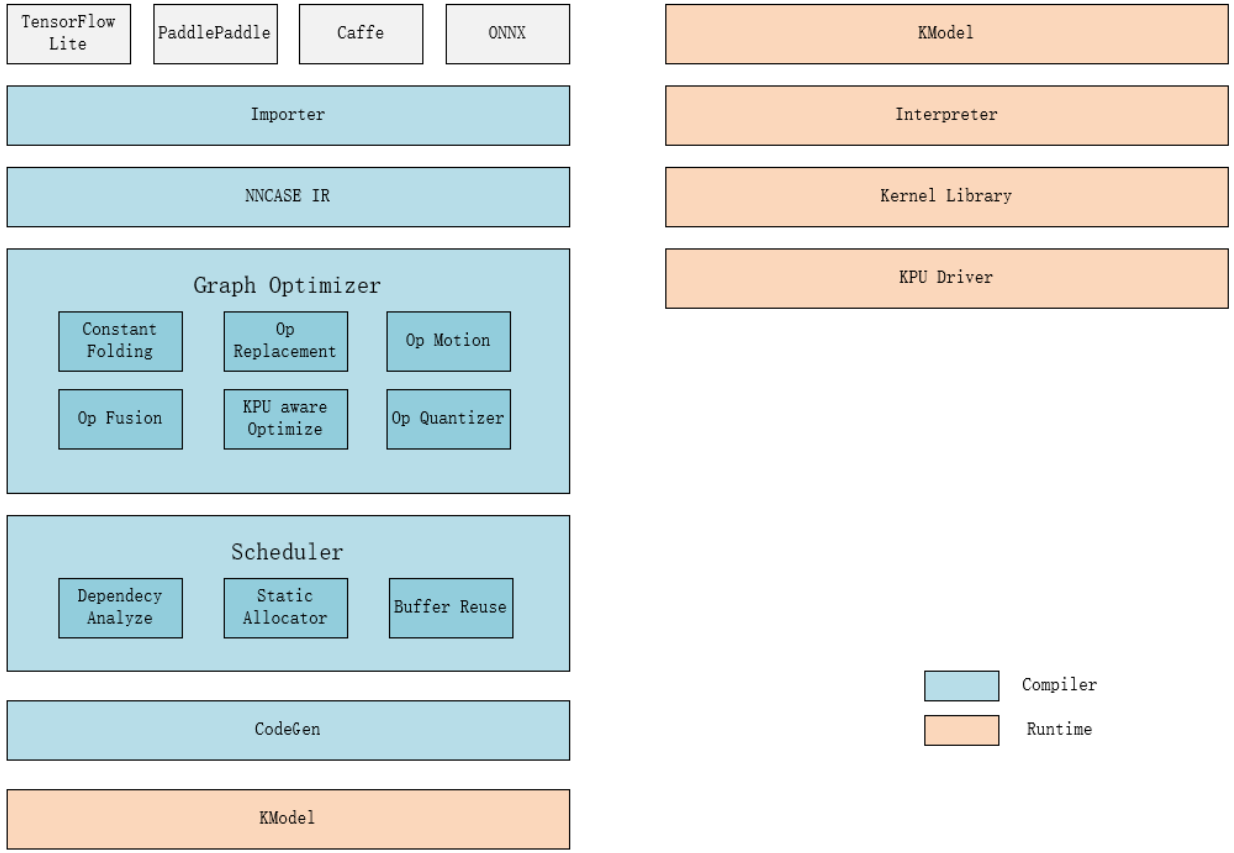
K210 是一款 RISC-V 双核 64 位 CPU，可以仅凭低功耗进行边缘计算，处理机器学习模型（功耗 1W，算力 1TOPS）。

MaixPy

SiPEED

MaixPy 是由 Sipeed 基于 K210 的一套 MicroPython 移植版本，在 MicroPython 能控制基础硬件属性的基础上，额外增加了图形处理（OpenMV）、机器学习模型加载、视频处理等功能

MaixPy & K210



https://github.com/kendryte/nncase

KPU

KPU is a general-purpose neural network processor, which can do convolutional neural network calculation at low power consumption, for example obtain the size, coordinates and types of detected objects or detect and classify faces and objects.

MaixPy & K210


```
import KPU as kpu
task = kpu.load(offset or file_path)
```

MaixPy & K210


```
import KPU as kpu
task = kpu.load(offset or file_path)
anchor = (1.889, 2.5245, 2.9465, 3.94056, 3.99987, 5.3658, 5.155437, 6.92275, 6.718375, 9.
kpu.init_yolo2(task, 0.5, 0.3, 5, anchor)
```

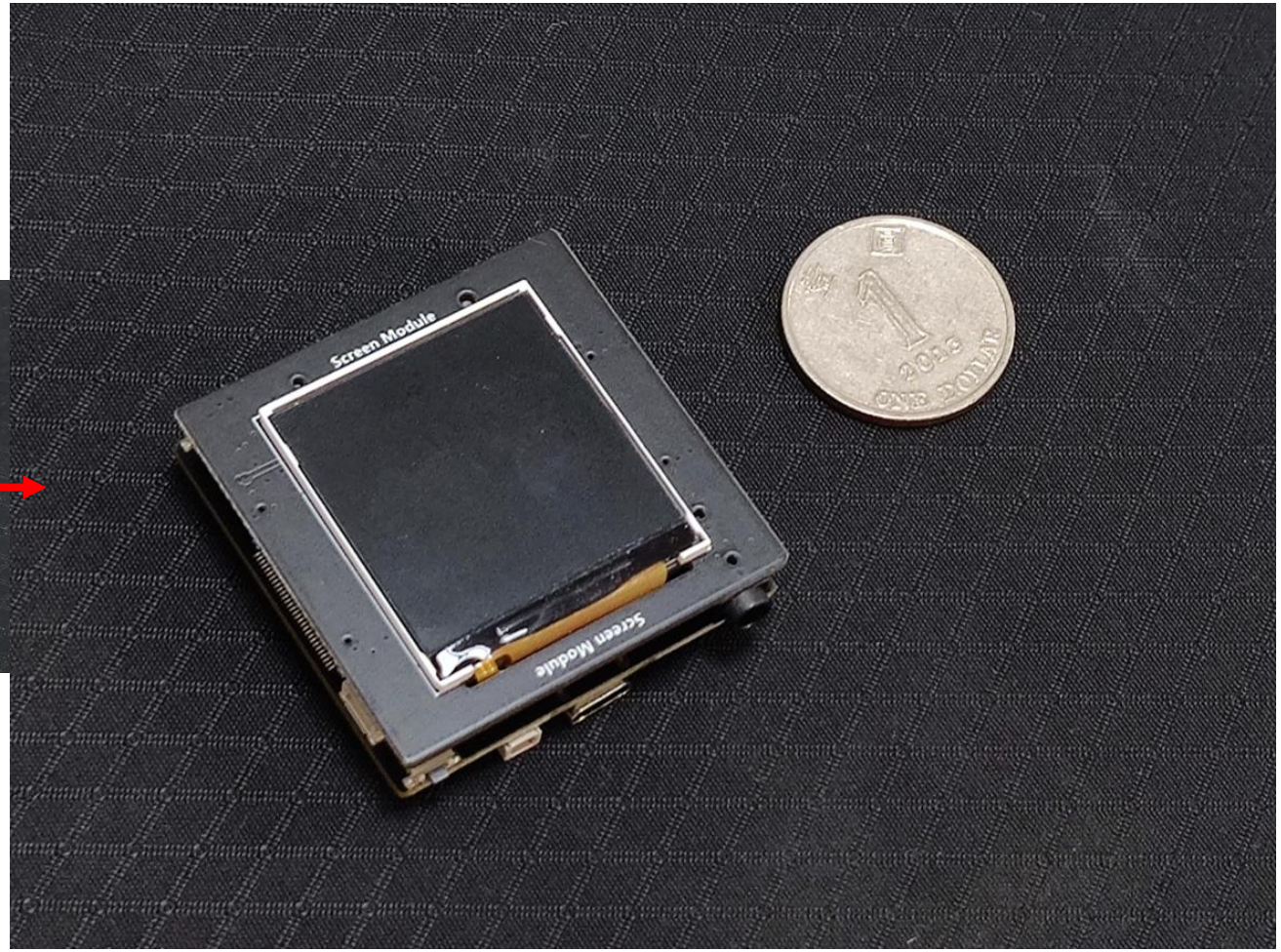
MaixPy & K210


```
32  while True:
33      img_facerecognition = sensor.snapshot()
34      code_facerecognition = kpu.run_yolo2(task_facerecognition, img_facerecognition)
35      img_facerecognition.ai_to_pix()
36      img_display = img_facerecognition.resize(224, 168)
37      img_display.draw_circle((int((224 / 2))),(int((168 / 2))), 2, color=(255, 255, 255), thickness=1, fill=True)
38      if code_facerecognition:
39          for i in code_facerecognition:
40              img_display.draw_rectangle(,, , , color=(255,255,255), thickness=2, fill=False)
41      _img_display_x, _img_display_y = 8, 36
42      lcd.display(img_display, oft=(_img_display_x,_img_display_y))
```

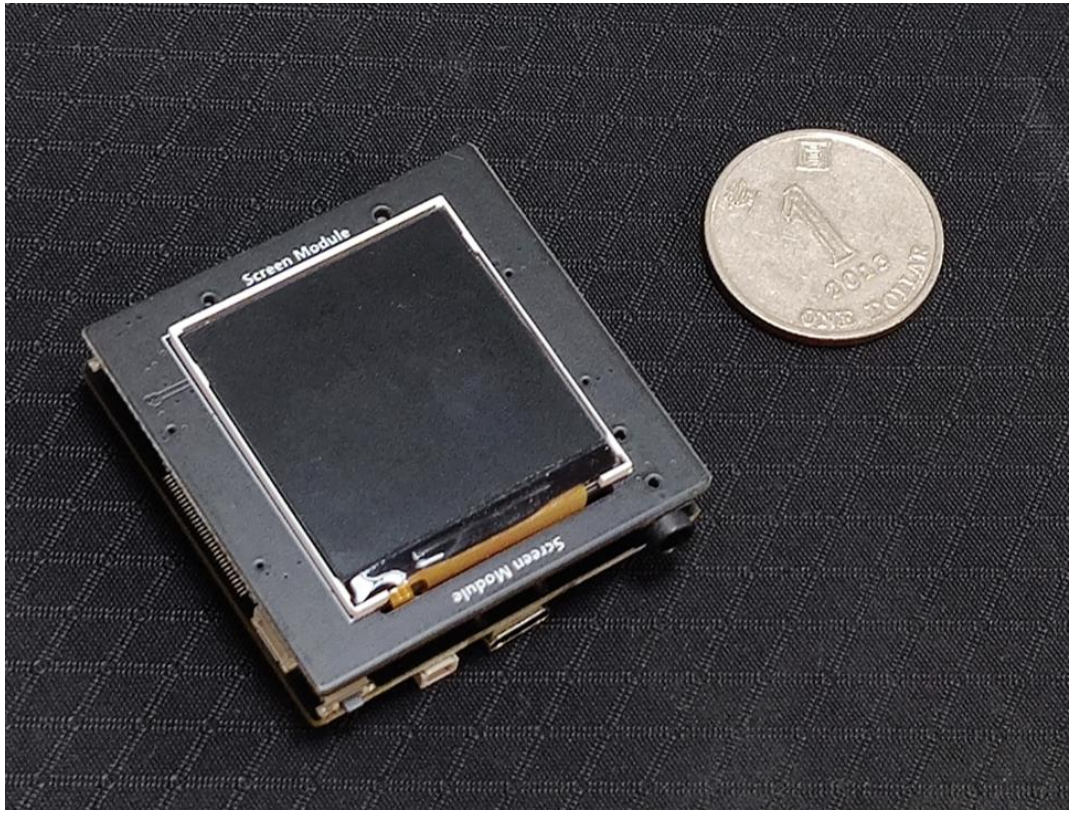
MaixPy & K210

我们集成了一款

仅比 1 元硬币稍大一些的 AI 教育模块



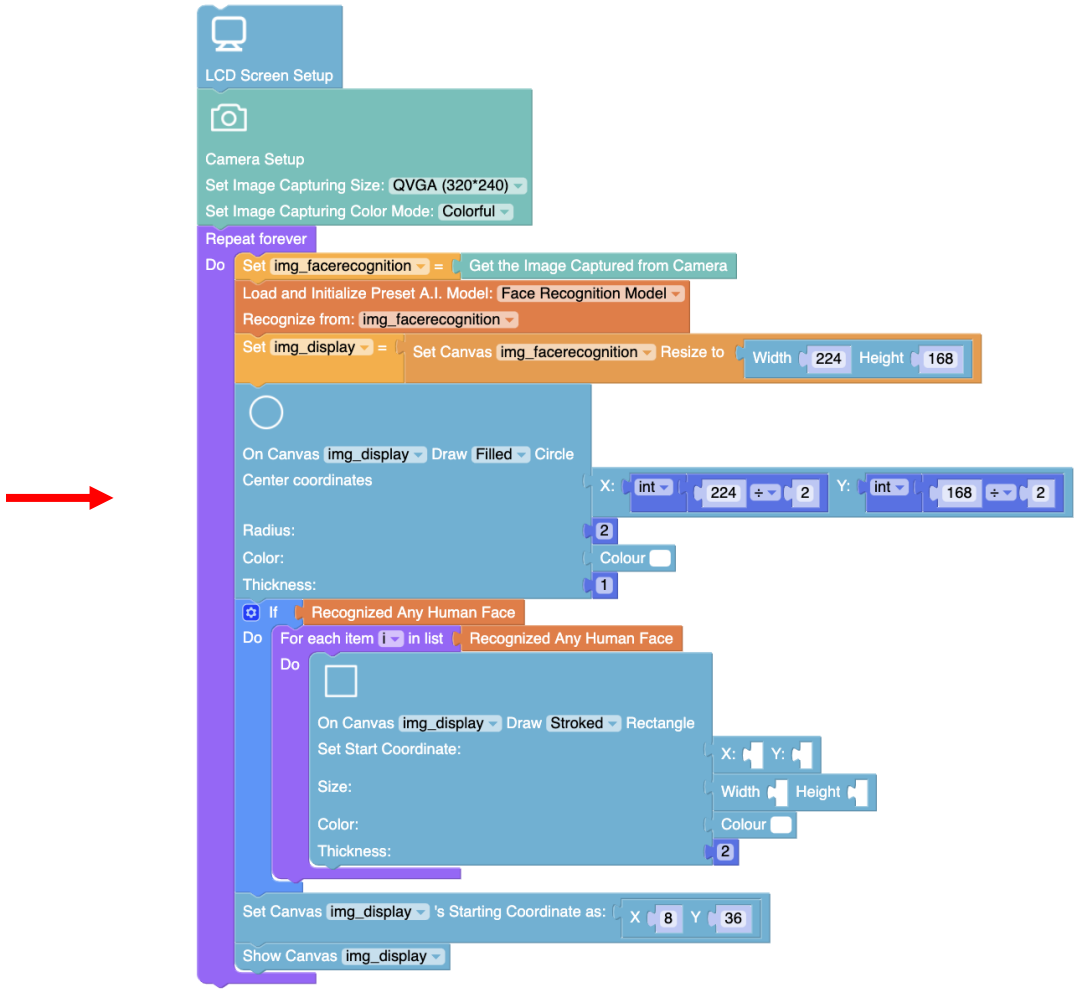
MaixPy & K210



- 计算机**视觉**：
  - **图**形操作：**图**像属性、尺寸控制等；
  - **圆**形**识**别、矩形**识**别、直**线识**别；
  - 二**维码识**别、条形**码识**别、AprilCode **识**别；
  - **颜**色**识**别、关**键**点**识**别
- 人**脸识**别、物体**识**别、人**脸**辨**识**、**语**音**识**别
- 运行自己**训练**的模型
  - YOLO, Keras, TensorFlow…；
  - ONNX, PaddlePaddle…

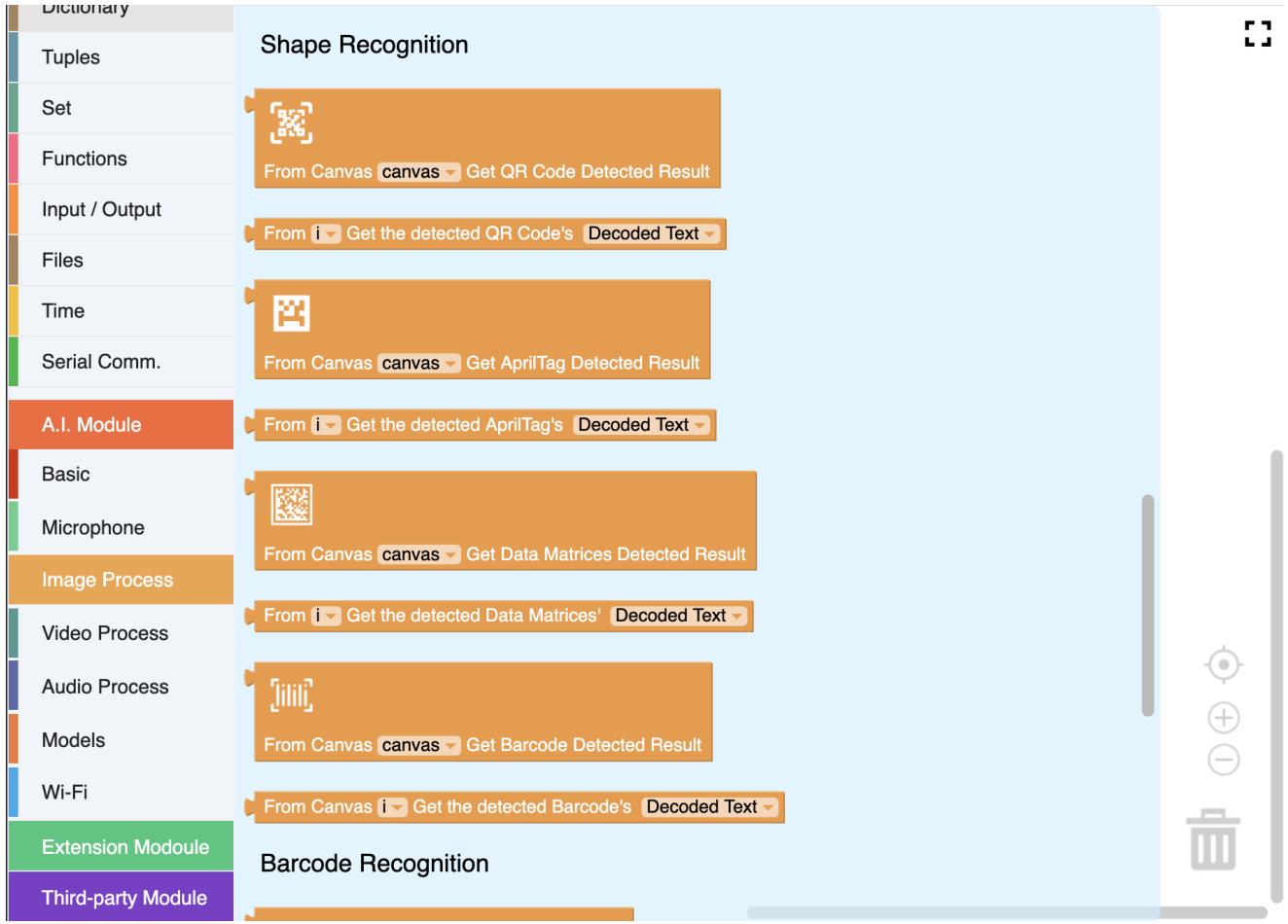
MaixPy & K210


```python
import lcd
import image
import sensor
import KPU as kpu

try:from cocorobo import display_cjk_string
except:pass

def lcd_draw_string(canvas, x, y, text, color=(255,255,255), font_size=1, scale=1, mono_space=False, auto_wrap=True):
    try:
        display_cjk_string(canvas, x, y, text, font_size=font_size, color=color)
        return canvas
    except: return canvas.draw_string(x, y, text, color=color, scale=scale, mono_space=mono_space)

_img_display_x, _img_display_y = 0, 0

task_facerecognition = kpu.load("/sd/preset/models/preset/face-recognition.kmodel")
anchor_face = (1.889, 2.5245, 2.9465, 3.94056, 3.99987, 5.3658, 5.155437, 6.92275, 6.718375, 9.01025)
a = kpu.init_yolo2(task_facerecognition, 0.5, 0.3, 5, anchor_face)

lcd.init(type=2,freq=15000000,width=240,height=240,color=(0,0,0))
lcd.rotation(1)
lcd.clear(lcd.BLACK)
sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.set_vflip(0)
sensor.set_hmirror(0)
sensor.skip_frames(30)
sensor.run(1)

while True:
    img_facerecognition = sensor.snapshot()
    code_facerecognition = kpu.run_yolo2(task_facerecognition, img_facerecognition)
    img_facerecognition.ai_to_pix()
    img_display = img_facerecognition.resize(224, 168)
    img_display.draw_circle((int((224 / 2))),(int((168 / 2))), 2, color=(255, 255, 255), thickness=1, fill=True)
    if code_facerecognition:
        for i in code_facerecognition:
            img_display.draw_rectangle(,, , , color=(255,255,255), thickness=2, fill=False)
    _img_display_x, _img_display_y = 8, 36
    lcd.display(img_display, oft=(_img_display_x,_img_display_y))
```



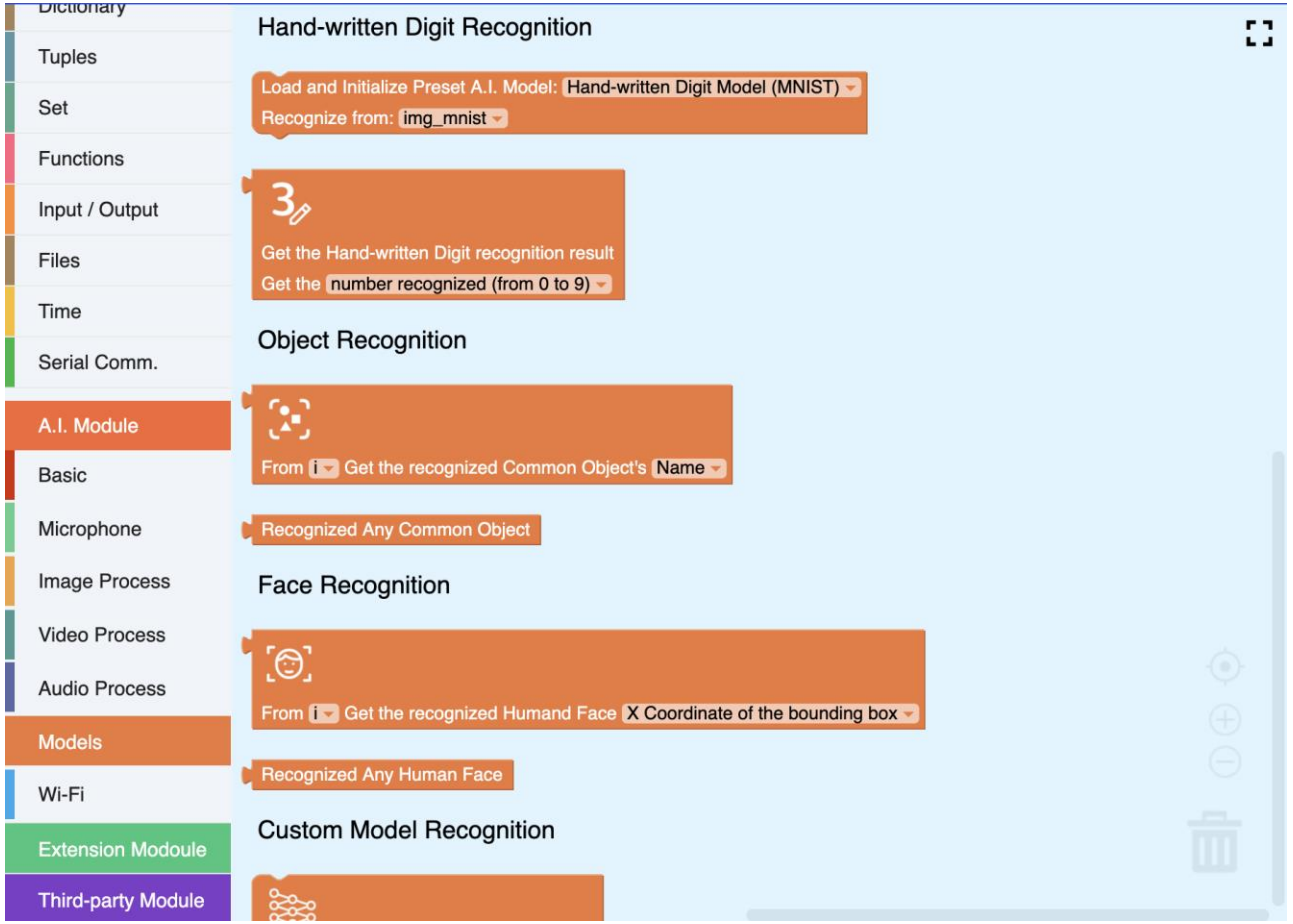
计算机视觉

- 图形操作：图像属性、尺寸控制等；
- 圆形识别、矩形识别、直线识别；
- 二维码识别、条形码识别；
- 颜色识别、关键点识别

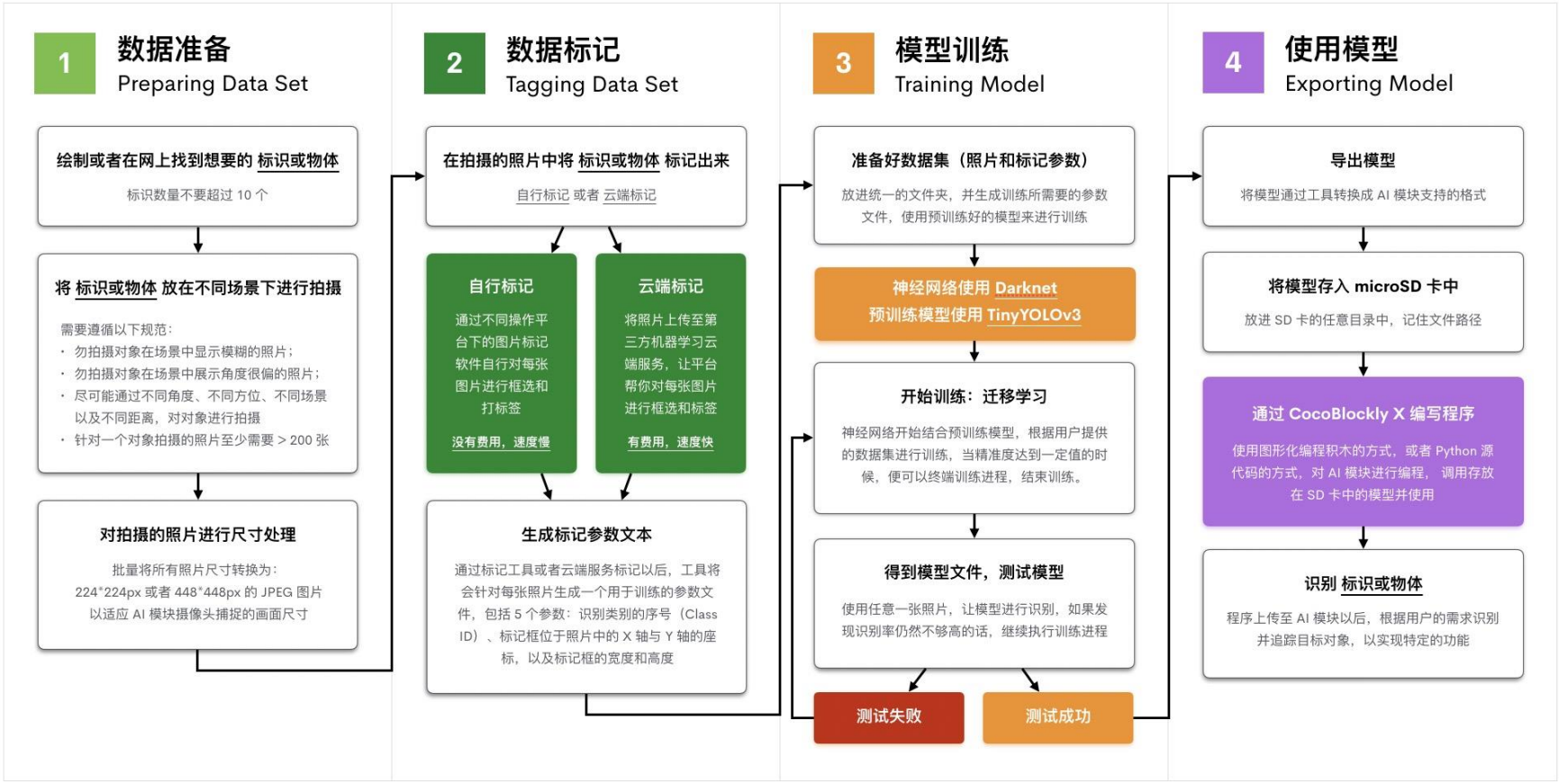


边缘计算（离线模型推理）

- 人脸识别／辨识、物体识别、语音识别
- 运行自己训练的模型
  - YOLO, Keras, TensorFlow…；
  - ONNX, PaddlePaddle…
- 离线分类器
  - 离线训练自定义物体识别模型



CocoRobo AI Training Center 模型训练全过程



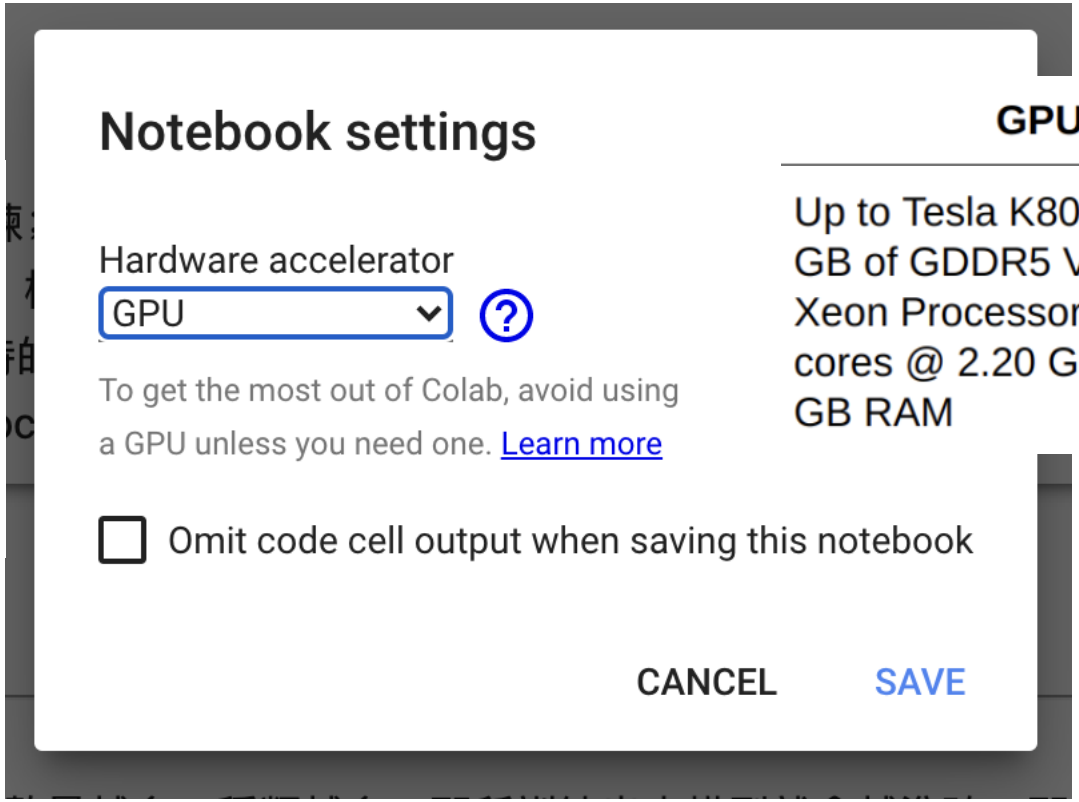
小学体验，中学深入

Agenda

- Python + Blockly = **积木式编程**
- Python + Embedded System = **电子编程教学**
- Python + Model Training = **人工智能教学**

Google Colab

colab



Google Colab

CocoRobo_人工智慧學習：機器學習全過程_一_從數據採集到使用模型.ipynb

File Edit View Insert Runtime Tools Help Last edited on March 23

Comment Share

+ Code + Text Connect Editing

概覽

本教程將會使用 Google Colab，來訓練一個可以識別客製化物件的模型，並能夠在 CocoRobo 二代 A.I. 模組中使用，整個過程分爲以下 6 個步驟：

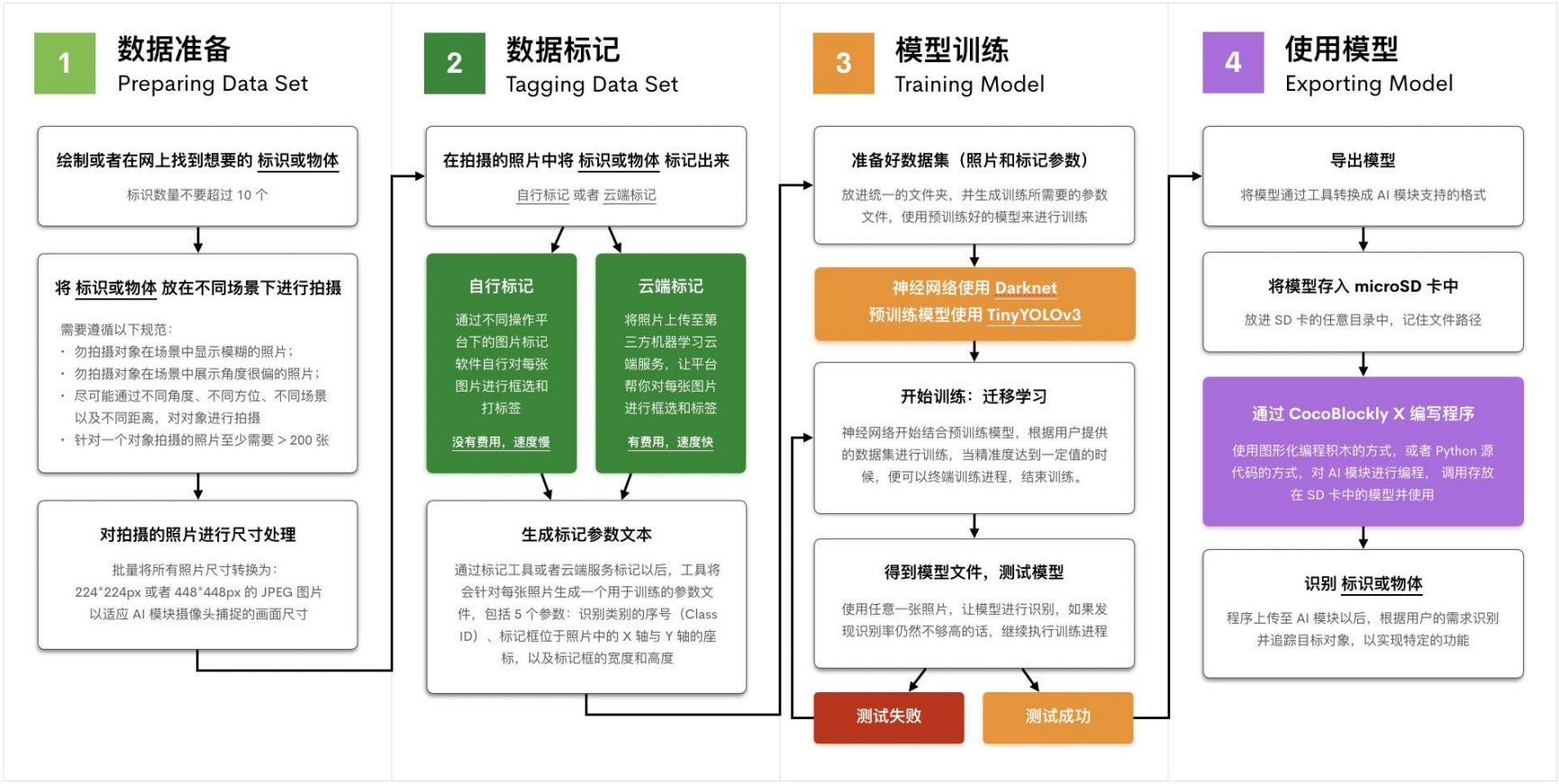
- **採集數據**：將要識別的物體進行拍攝；
- **標記數據**：將拍攝的照片進行數據標記；
- **訓練模型**：將標記好的照片放入訓練工具中進行遷移訓練；
- **驗證模型**：用訓練好的模型來識別隨機抽取的一張照片，檢驗識別率；
- **導出模型**：將訓練的模型轉換成 CocoRobo A.I. 模組支持的模型格式；
- **使用模型**：將轉換好的模型存入 microSD 卡中，透過 CocoBlockly X 來進行模型識別的程式設計

一、採集數據

衆所周知，機器學習模型最重要的部分就是數據集，數據集的數量越多、種類越多，那所訓練出來模型就會越準確。那麼對於基於圖像的模型訓練，我們的數據就是各式各樣的圖片文件，這些圖片中要包含你要識別的物體，你要通過不同的角度、不同的距離、不同的背景去拍攝他們。

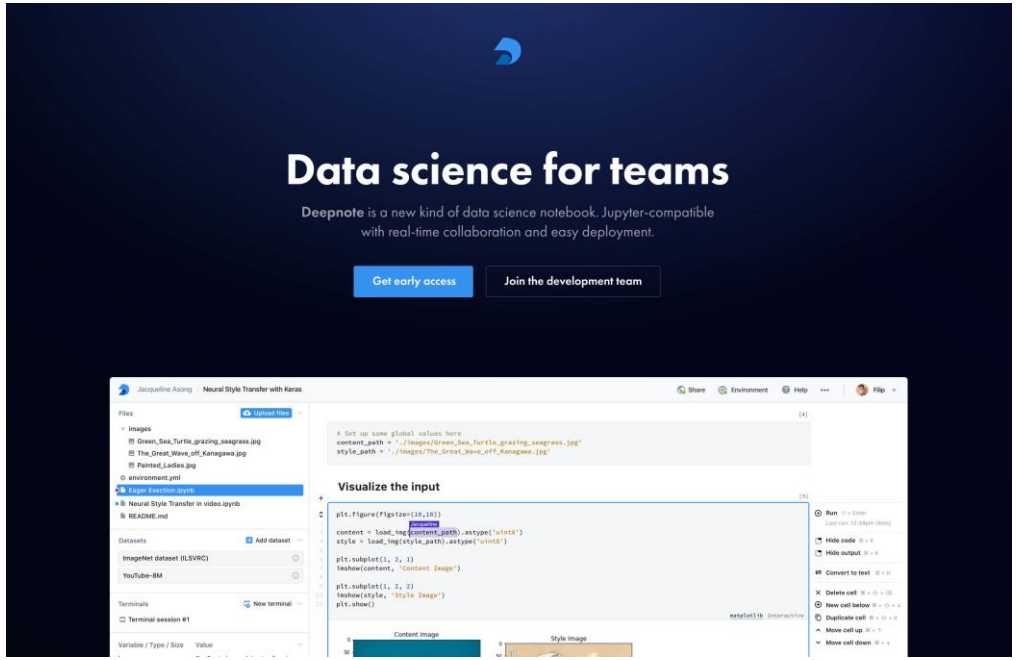
在這篇教程中，我們將識別以下 4 張卡片中的物件：

CocoRobo AI Training Center 模型训练全过程



小学体验，中学深入

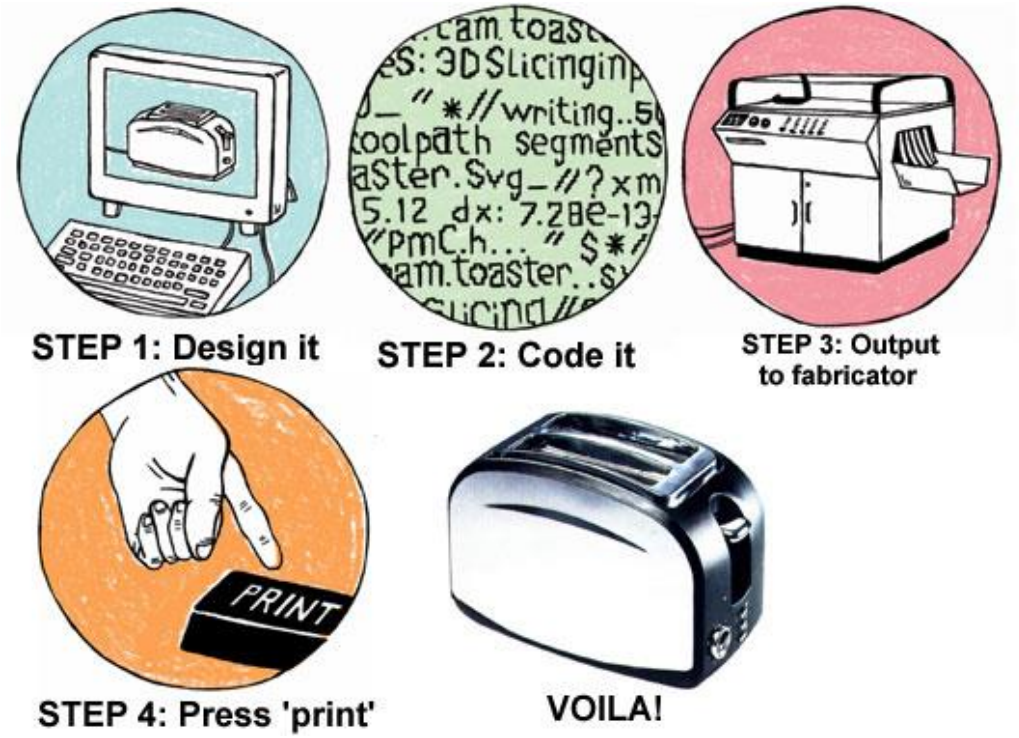
Google Colab



Agenda

Make Almost Anything with Python!

Make with Python



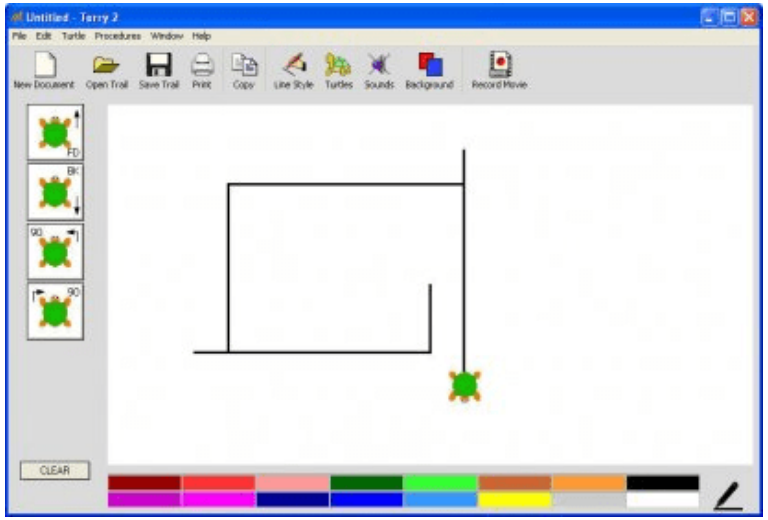
http://archive.boston.com/news/globe/ideas/articles/2005/01/30/how_to_make_almost_anything/

Make Almost Anything with Python!

Python 不仅可以很方便的拿来写软件，也可以很方便地拿来控制硬件，

从物联网控制，到边缘计算，这些你想实现的 Idea，

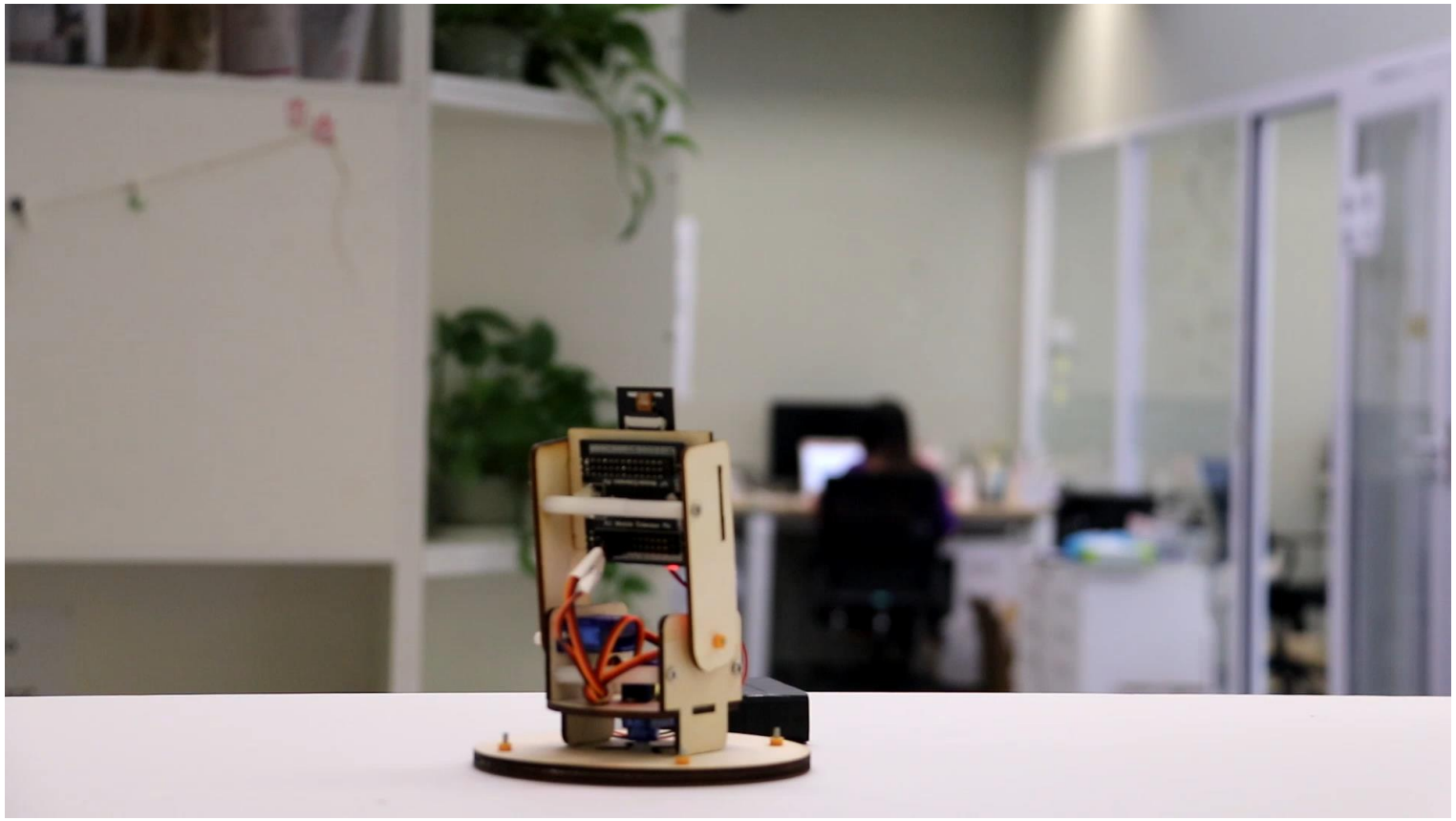
都可以以很低的门槛学习，并最终创作出来



编程学习工具都应该秉承 Logo 的理念：易学性高、可扩展性高

Make with Python: Some Cases

人脸识别云台



Make with Python: Some Cases

垃圾识别自动分类

PyCon China 2020

Python For Good

PyConChina 2020 | PYTHON 中国开发者大会 2020

Thank You!

 @technommy

 @tommyjtl

 tjtl.io

 tommy@cocorobo.cc