

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

- 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

*: 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。

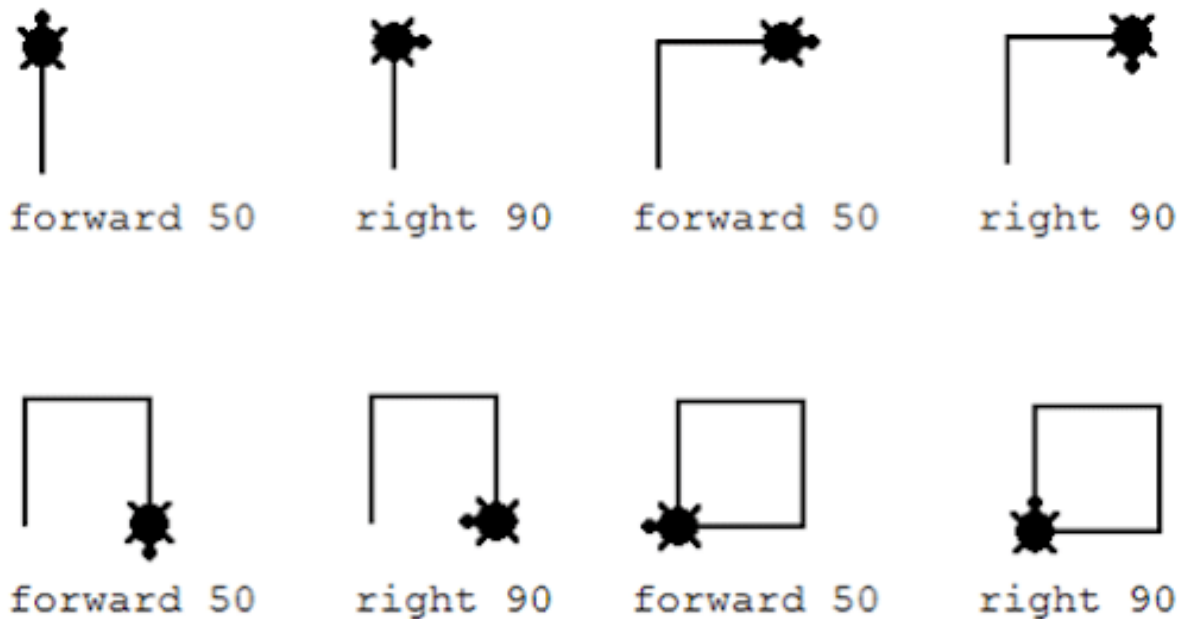
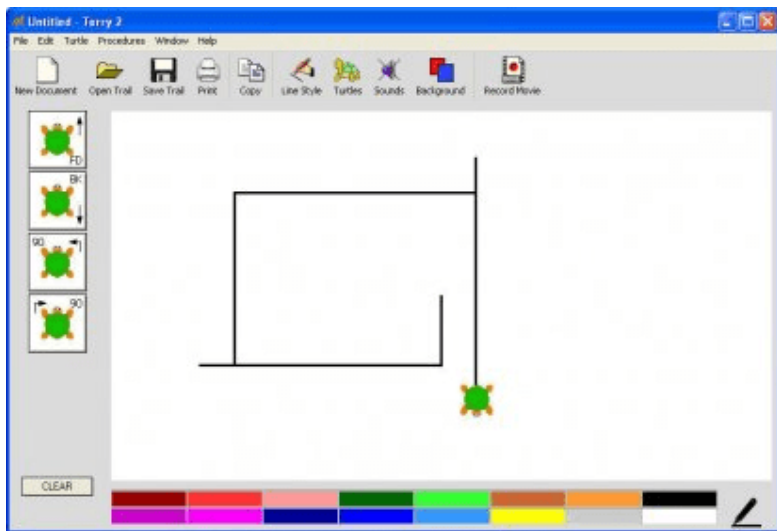
K-12 教育里的 Python + AI 实践

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？

PC Logo



© 2000 Logo Foundation

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

[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](#)



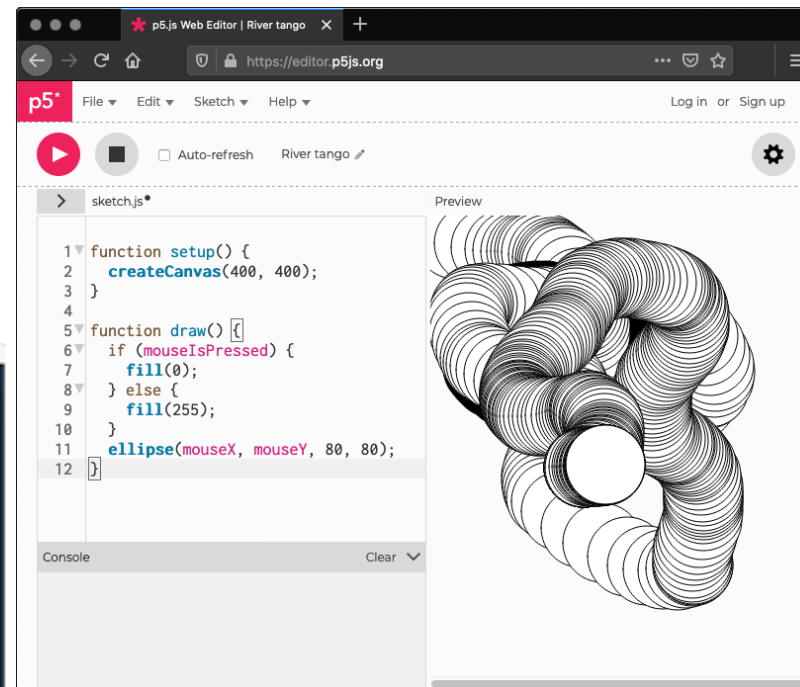
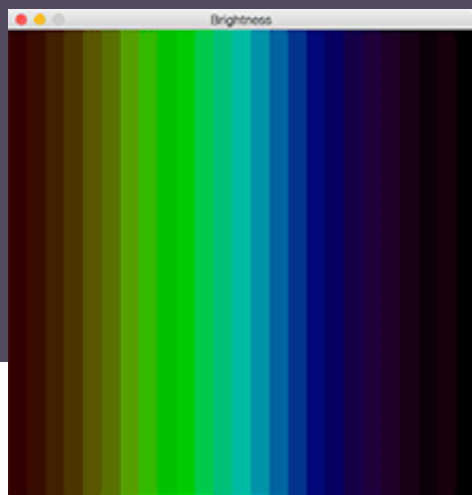
[Megan Galbraith](#)



[Simon Greenwold](#)

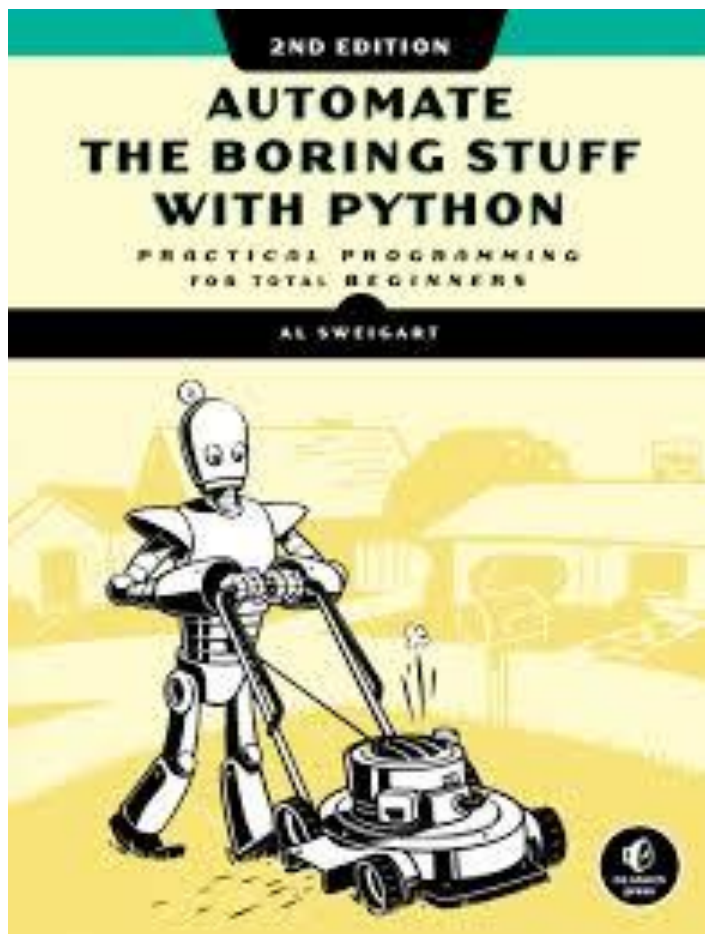


[Justin Manor](#)



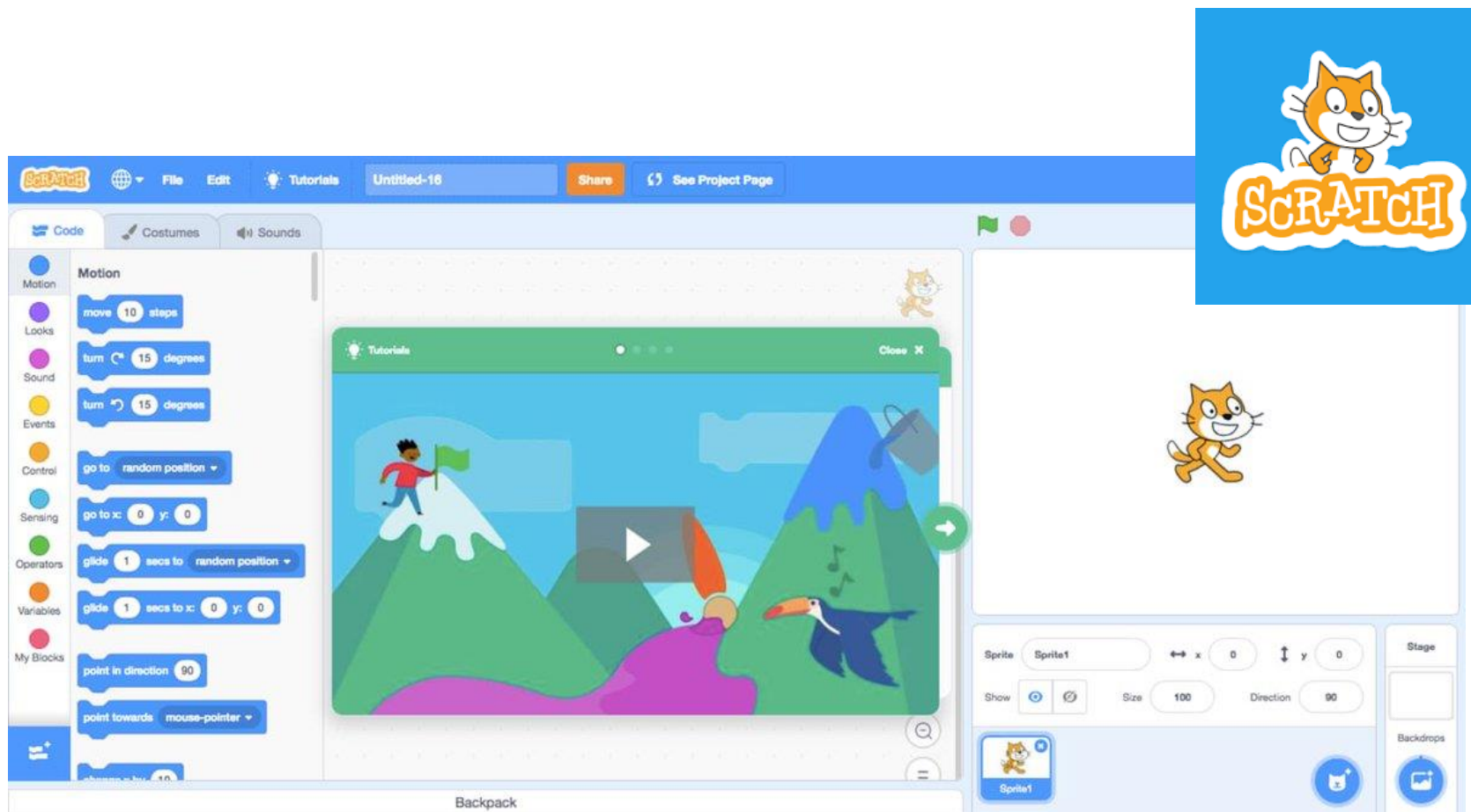
Message Area

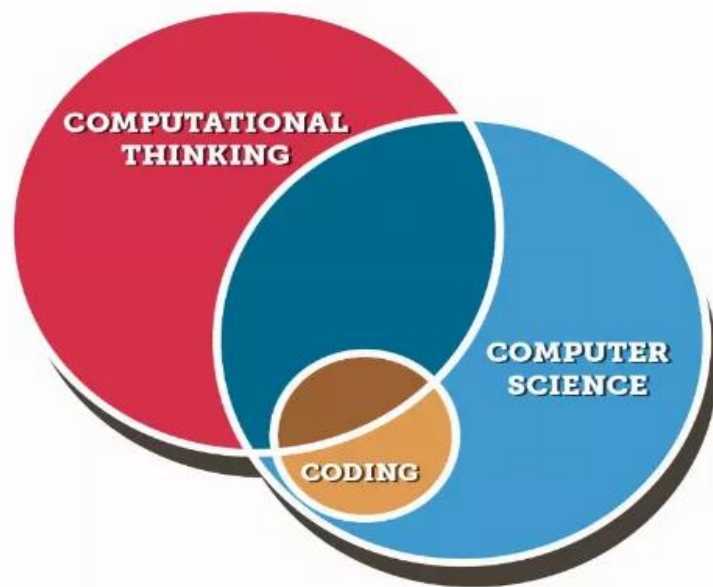
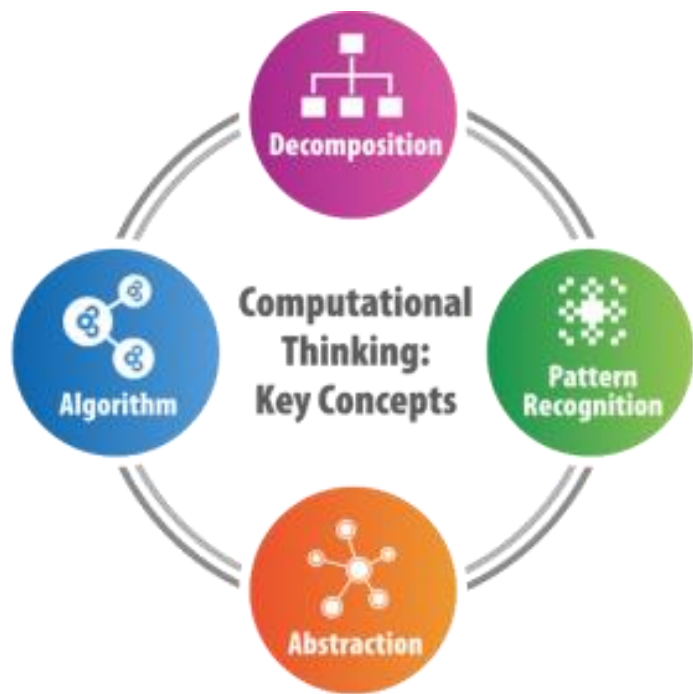
Console/Errors



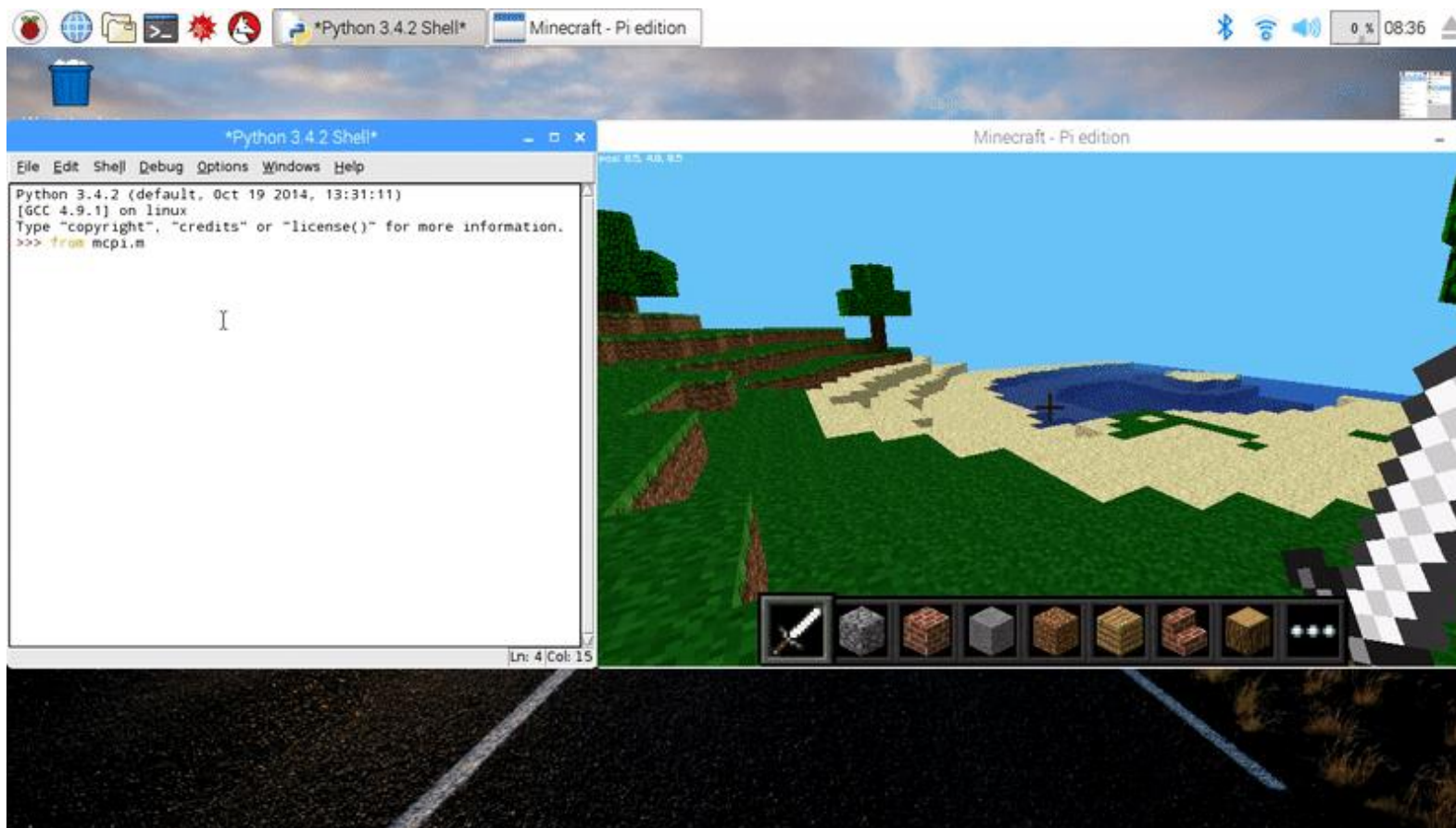
培养学生的逻辑思维能力 提供一种新的思维工具去解决事情

适用于画家、音乐家、建筑师、市场营销等各行各业





MineCraft 脚本编辑器



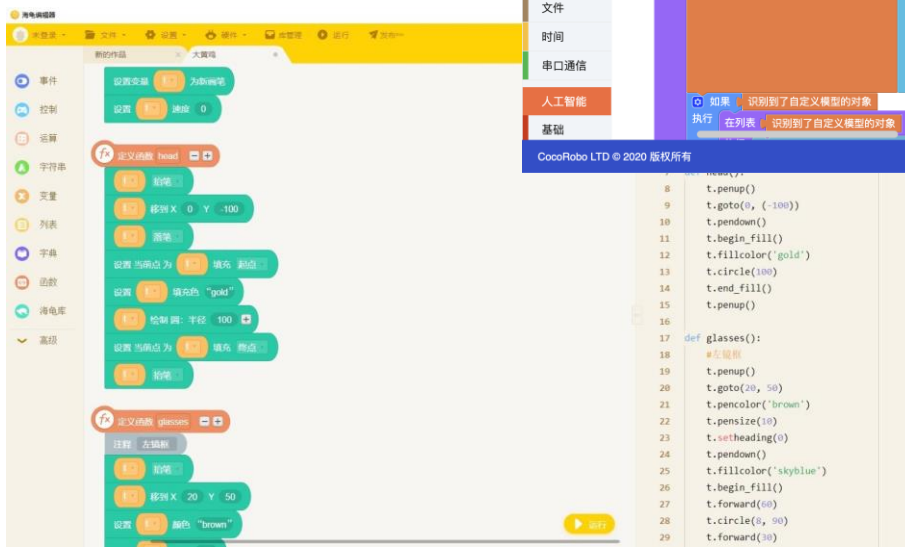
教些什么：游戏化学习



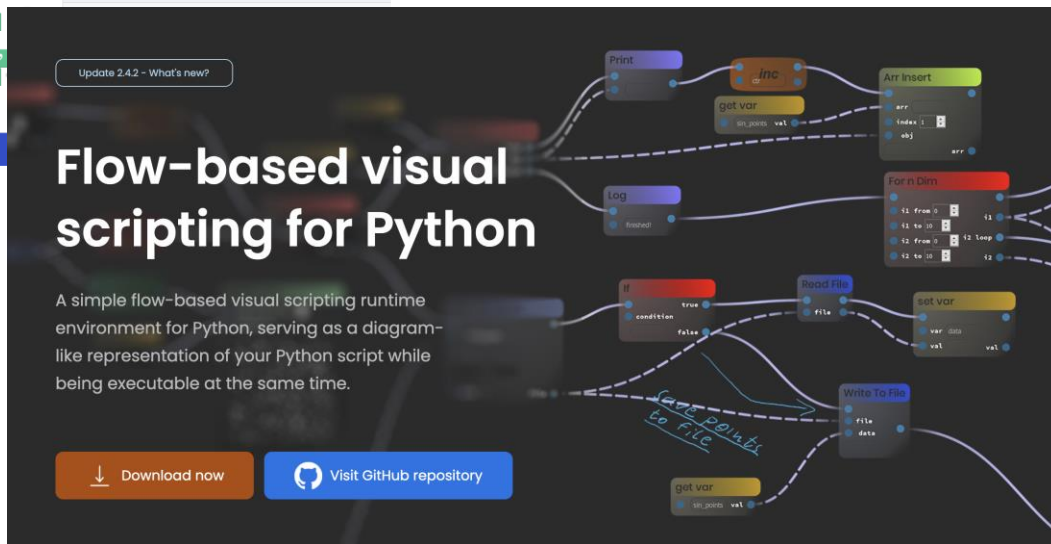
教些什么：图形化编程

CocoBlockly X

海龟编辑器



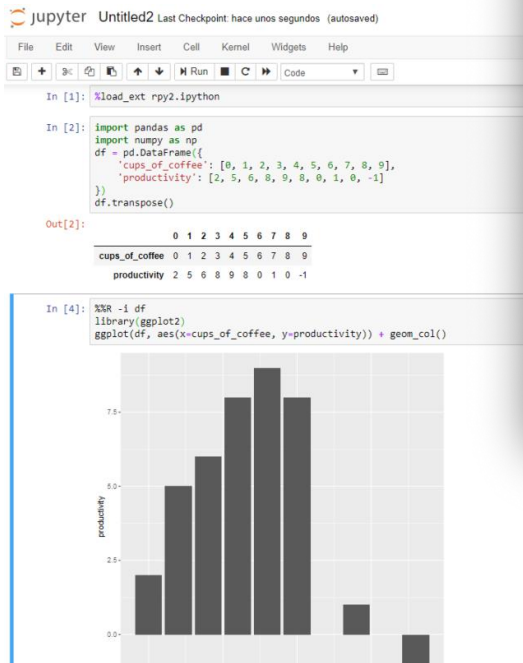
Ryven



教些什么：源码式编程

Python Official Build

Jupyter Notebook



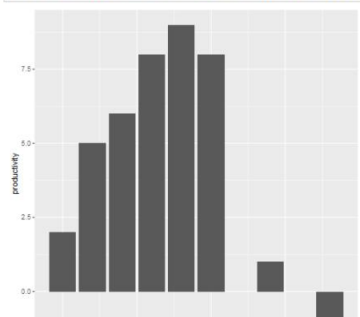
The screenshot shows a Jupyter Notebook interface with the following code and output:

```
In [1]: %load_ext rpy2.ipynthon
```

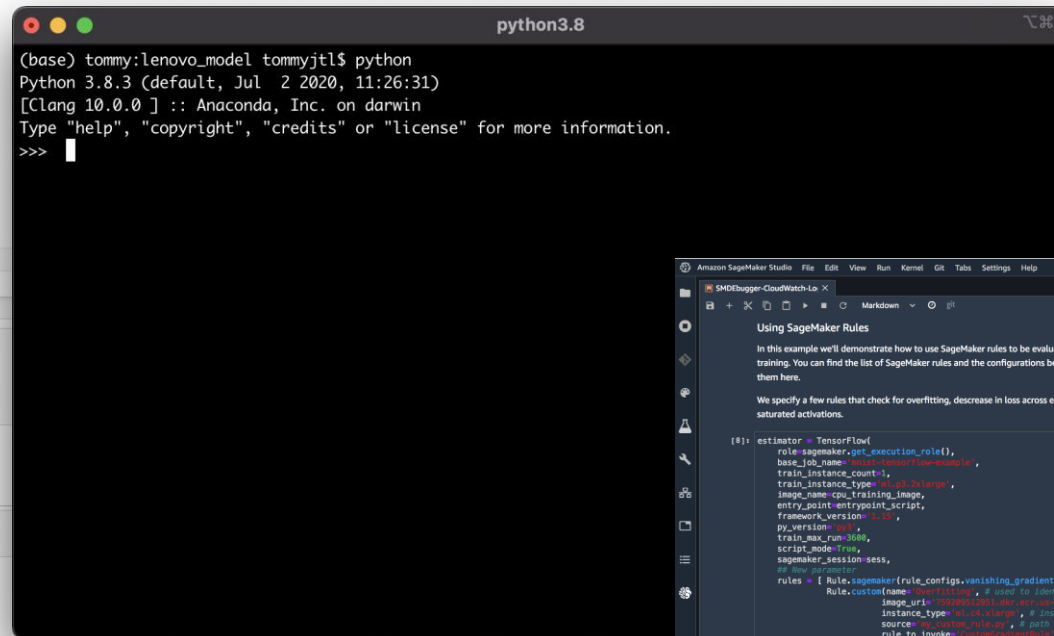
```
In [2]: import pandas as pd
import numpy as np
df = pd.DataFrame({
    'cups_of_coffee': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
    'productivity': [2, 5, 6, 8, 9, 8, 0, 1, 0, -1]
})
df.transpose()
```

	0	1	2	3	4	5	6	7	8	9
cups_of_coffee	0	1	2	3	4	5	6	7	8	9
productivity	2	5	6	8	9	8	0	1	0	-1

```
In [4]: %R -i df
library(ggplot2)
ggplot(df, aes(x=cups_of_coffee, y=productivity)) + geom_col()
```

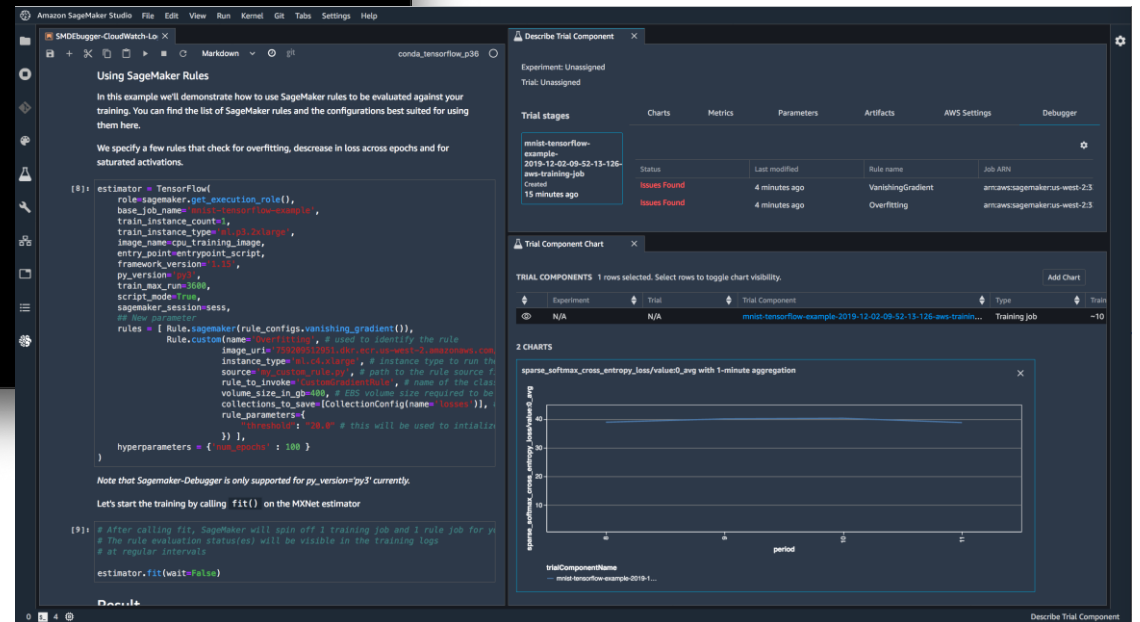


The bar chart displays productivity on the y-axis (ranging from -1 to 9) against the number of cups of coffee on the x-axis (ranging from 0 to 9). The bars represent the productivity values: 2, 5, 6, 8, 9, 8, 0, 1, 0, -1.



```
python3.8
(base) tommy:lenovo_model tommyjtl$ python
Python 3.8.3 (default, Jul 2 2020, 11:26:31)
[Clang 10.0.0] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

AWS SageMaker Studio



The screenshot shows the AWS SageMaker Studio interface with the following code and output:

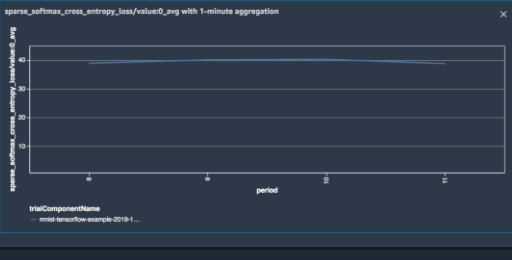
```
[8]: estimator = TensorFlow(
    role=sagemaker.get_execution_role(),
    base_job_name='ml-tensorflow-example',
    train_instance_count=1,
    train_instance_type='ml.t2.xlarge',
    image_name='cpu_training_image',
    entry_point='entrypoint_script',
    framework_version='mxnet',
    py_version='py3',
    train_max_run=3600,
    script_mode=True,
    sagemaker_session=sess,
    # Now parameter
    rules = [ Rule.sagemaker(rule_configs.vanishing_gradient()),
              Rule.custom(name='overfitting', # used to identify the rule
                           image_url='75249512951.dkr.ecr.us-west-2.amazonaws.com',
                           instance_type='ml.t2.xlarge', # instance type to run the
                           source='custom_rule.py', # path to the rule source
                           rule_to_invoke='CustomGradientRule', # name of the class
                           volume_size_gb=100, # EBS volume size required to be
                           collections_to_save=[CollectionConfig(name='losses')]),
              ],
    hyperparameters = { 'num_epochs': 100 }
)
```

Note that SageMaker-Debugger is only supported for py_version='py3' currently.
Let's start the training by calling fit() on the MXNet estimator

```
[9]: # After calling fit, SageMaker will spin off 1 training job and 1 rule job for you
# The rule evaluation status(es) will be visible in the training logs
# at regular intervals
estimator.fit(wait=False)
```

The interface also shows a table of trial stages and a chart of sparse_softmax_cross_entropy_loss/value0_avg with 1-minute aggregation.

Experiment	Trial	Trial Component	Type	Train
N/A	N/A	ml-tensorflow-example-2019-12-02-09-52-15-126-aws-train...	Training job	-10



The chart shows the loss value over time, with the y-axis labeled 'sparse_softmax_cross_entropy_loss/value0_avg' and the x-axis labeled 'period'.

K-12 教育里的 Python + AI 实践

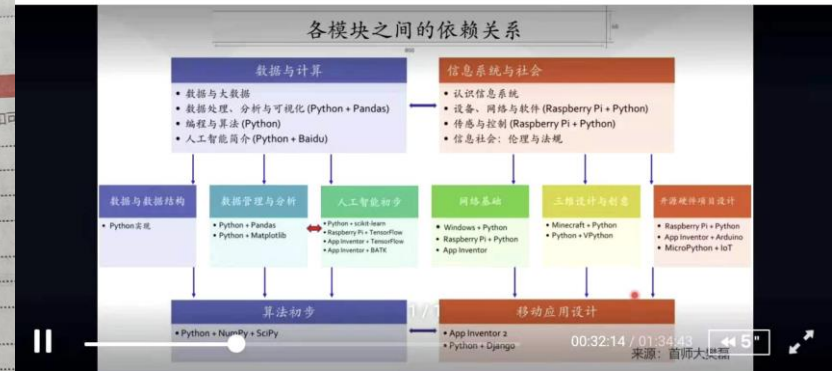
1. 教些什么？
2. 怎么去教？
3. 用什么教？

怎么教？(Python & AI)



2.2 数字化学习与创新	29
2.2.1 数字化工具与资源及其优势	29
2.2.2 体验数字化创新学习过程	33
第三章 算法基础	39
项目范例 设计从A市到B市耗时最少的旅行路线方案	40
3.1 体验计算机解决问题的过程	43
3.1.1 人工解决问题的过程	43
3.1.2 计算机解决问题的过程	46
3.2 算法及其描述	48
3.2.1 算法	48
3.2.2 算法的描述	49
3.3 计算机程序与程序设计语言	53
3.3.1 计算机程序	54
3.3.2 计算机程序设计语言	55
第四章 程序设计基础	63
项目范例 设计购买纪念品的最佳方案	64
4.1 程序设计语言的基础知识	66
4.1.1 Python的常量和变量	67
4.1.2 Python的数据类型	68
4.1.3 Python的运算符与表达式	69
4.1.4 Python的函数	71
4.2 运用顺序结构描述问题求解过程	73
4.2.1 数据的输入与输出	73
4.2.2 顺序结构的应用	74
4.3 运用选择结构描述问题求解过程	76
4.3.1 if语句的应用	80
4.3.2 if...elif...else语句的应用	85
4.4 运用循环结构描述问题求解过程	85
4.4.1 for循环的应用	88
4.4.2 while循环的应用	88
4.4.3 循环嵌套的应用	88
4.4.4 循环控制的应用	88
第五章 数据处理和可视化表达	99
项目范例 网络购物平台客户行为数据分析和可视化	99
5.1 认识大数据	100
5.1.1 大数据	100
5.1.2 大数据的特征	100
5.1.3 大数据对日常生活的影响	100
5.2 数据的采集	100
5.2.1 数据采集的方法和工具	100
5.2.2 数据的存储和保护	100
5.3 数据的分析	100
5.3.1 特征探索	113
5.3.2 关联分析	115
5.3.3 聚类分析	115
5.3.4 数据分类	116
5.4 数据的可视化表达	118
5.4.1 数据可视化表达的方式	119
5.4.2 数据可视化表达的工具	120

× 【课程设计】基于RoboMaster EP 的创客课程设计



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)

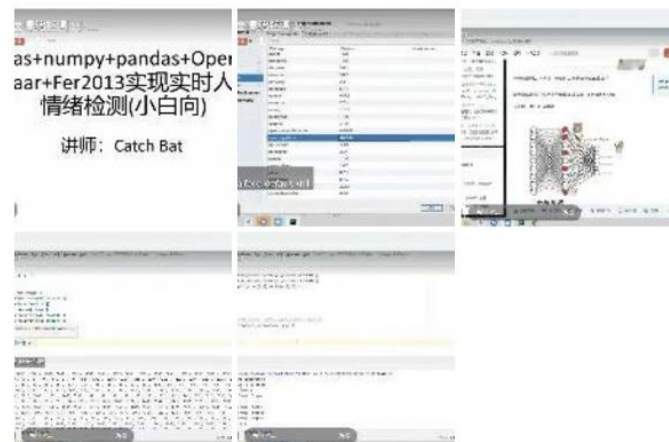
13岁的初中学生可以特别6地用tensorflow搞个小ai，未来少儿编程(api调用师)会像语文数学一样成为基础必修课



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👍



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？

- 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

- 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

The screenshot shows the 'Blockly Games: Maze' interface. The browser address bar displays `https://blockly-games.appspot.com/maze?lang=en&level=10&skin=0`. The game title 'Blockly Games: Maze' is followed by a progress indicator of 10 circles and a language dropdown set to 'English'. The main area features a maze with a yellow path and a character. A notification at the bottom left states 'You have 4 blocks left.' and a 'Reset' button is at the bottom center. The code editor on the right contains the following script:

```
repeat until [ ]  
do  
  if path [to the left]   
  do [turn left]   
  if path [ahead]   
  do [move forward]   
  else [turn right]
```

Block Factory Block Exporter Workspace Factory

Block Library Save "block_type" Delete "block_type" Preview: LTR Clear Library Import Block Library Download Block Library

Input
Field
Type
Colour

name block_type
inputs
automatic inputs
no connections
tooltip
help url
colour hue: 230°

Block Definition: JSON

```
{  
  "type": "block_type",  
  "message0": "",  
  "colour": 230,  
  "tooltip": "",  
  "helpUrl": ""  
}
```

Generator stub: Python

```
Blockly.Python['block_type'] = function(block) {  
  // TODO: Assemble Python into code variable.  
  var code = '...\n';  
  return code;  
};
```

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


`random.randint(1, 100)`

Block Factory Block Exporter Workspace Factory

Block Library Save "random_int" Delete "random_int" Preview: LTR Clear Library Import Block Library Download Block Library

Input Field Type Colour

name random_int

inputs

- value input from
- fields left text Random Integer from
- type Number
- value input to
- fields left text to
- type Number

inline inputs

- left output
- tooltip
- help url
- output type any
- colour hue: 230°

Preview:

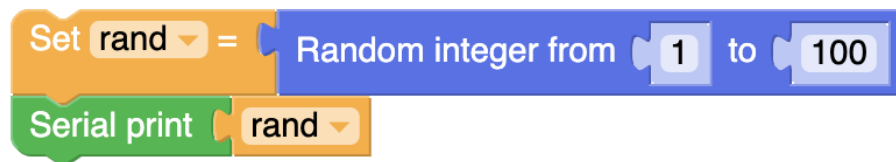
Random Integer from to

Block Definition: JSON

```
{
  "type": "random_int",
  "message0": "Random Integer from %1 to %2",
  "args0": [
    {
      "type": "input_value",
      "name": "from",
      "check": "Number"
    },
    {
      "type": "input_value"
```

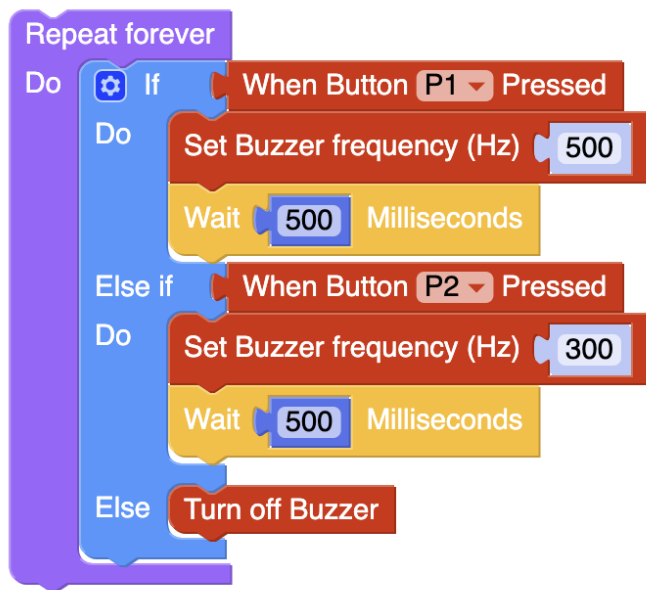
Generator stub: Python

```
Blockly.Python['random_int'] = function(block) {
  var value_from = Blockly.Python.valueToCode(block, 'from', Blockly.Python.ORDER_ATOMIC);
  var value_to = Blockly.Python.valueToCode(block, 'to', Blockly.Python.ORDER_ATOMIC);
  // TODO: Assemble Python into code variable.
  var code = '...';
  // TODO: Change ORDER_NONE to the correct strength.
  return [code, Blockly.Python.ORDER_NONE];
};
```



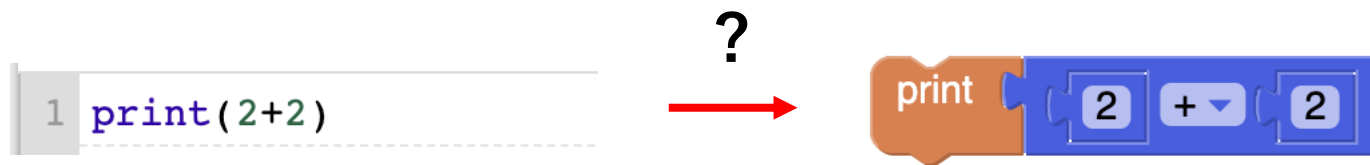
```
1 import random
2
3
4
5 rand = random.randint(1, 100)
6 print(rand)
7
```

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
```



能不能把代码转成积木？



能不能把代码转成积木？

Google Blockly

The image shows the Google Blockly web interface. On the left is a library of blocks with the following categories: Logic, Loops, Math, Variables, Text, Lists, Functions, Output, Turtles, Dictionaries, Classes, Tuples, and Data - Weather. Below the library is an "ADD LIBRARY" button. The main workspace is currently empty. On the right side, there are two panels: "Python Source Code" at the top, which is empty and has a line number "1" and a copy icon; and "Output Area" below it, which is also empty. At the bottom right of the interface are three buttons: "RUN", "CLEAR", and "RESET".

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)

纯前端
(Skulpt, Pydiode)

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

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.

JavaScript		Python	Example
String	↔	str	"Hello, Pyodide"
Uint8ClampedArray	↔	bytes	"\xff\x7"
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

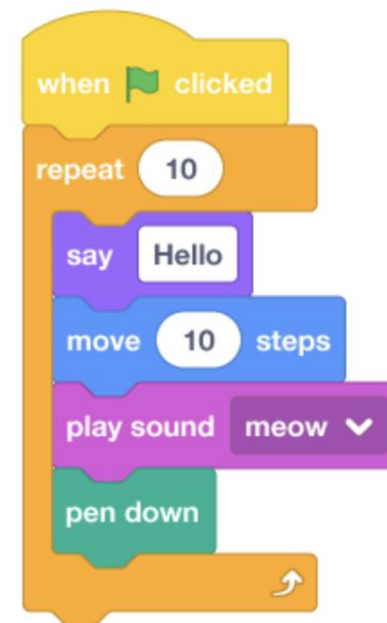
```
import pygame theApp = App()
```

```
pygame.display.set_mode
```

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



```
3 import pygame
4 from pygame.locals import *
5
6 class App:
7     def __init__(self):
8         self._running = True
9         self._display_surf = None
10        self.size = self.weight, self.height = 640, 400
11
12    def on_init(self):
13        pygame.init()
14        self._display_surf = pygame.display.set_mode(self.size)
15        self._running = True
16
17    def on_event(self, event):
18        if event.type == pygame.QUIT:
19            self._running = False
20
21    def on_loop(self):
22        pass
23
24    def on_render(self):
25        pass
26
27    def on_cleanup(self):
28        pygame.quit()
29
30    def on_execute(self):
31        if self.on_init() == False:
32            self._running = False
33
34        while self._running:
35            for event in pygame.event.get():
36                self.on_event(event)
37                self.on_loop()
38                self.on_render()
39                self.on_cleanup()
40
41 if __name__ == "__main__":
42     theApp = App()
43     theApp.on_execute()
```



```
import pygame theApp = App()
```

```
pygame.display.set_mode
```

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

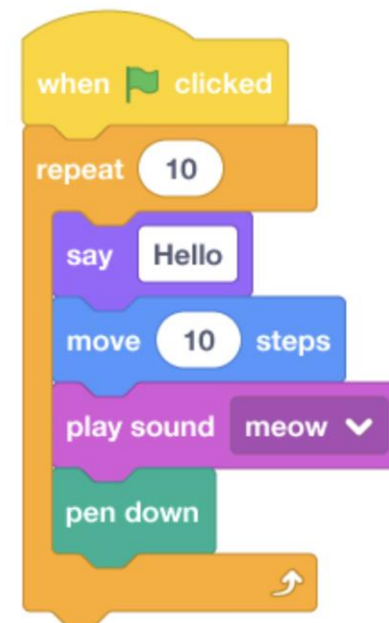


```
import pygame
from pygame.locals import *

class App:
    def __init__(self):
        self._running = True
        self._display_surf = None
        self.size = *pygame.display.get_surface().get_size(), 640, 480
        self._display_surf = pygame.display.set_mode(self.size, pygame.DOUBLEBUF)
        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()
```

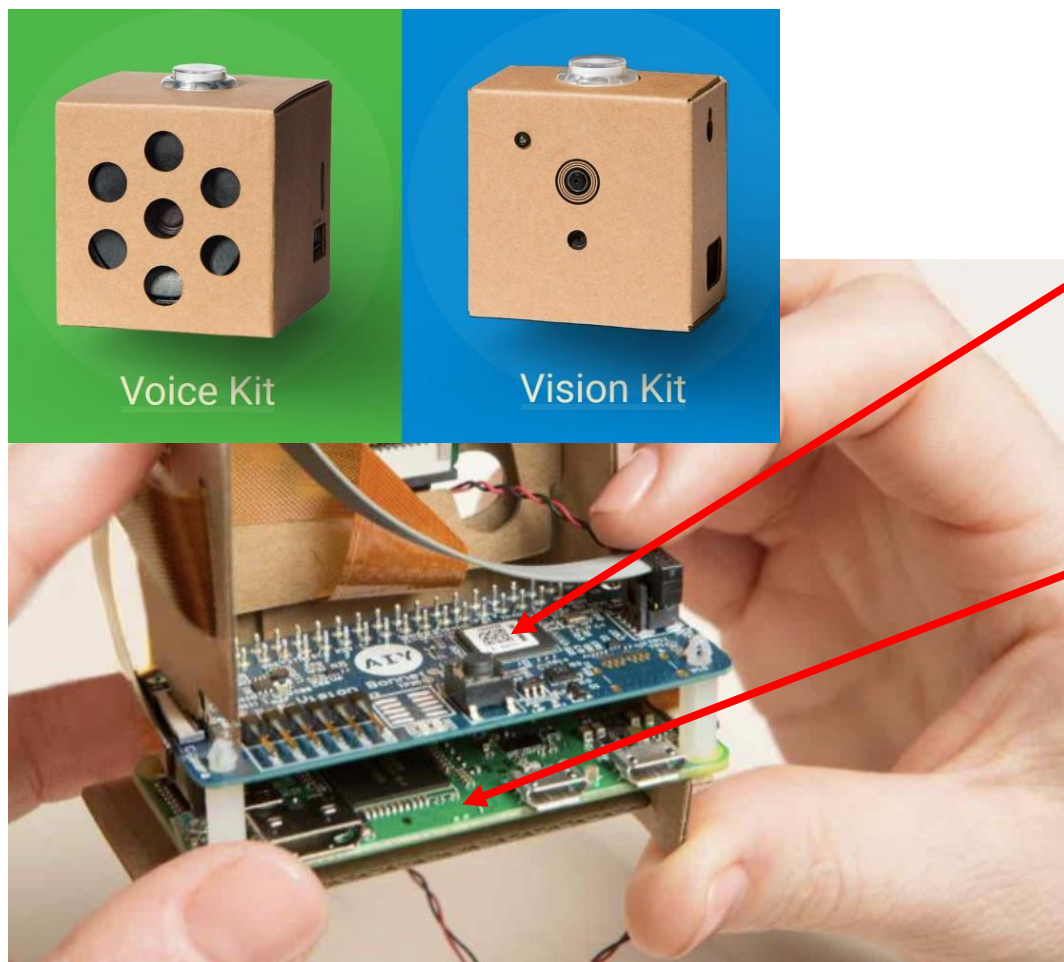
不适合刚入门
的用户学习



Google Blockly



基础的 Python 可以积木化，
Python + A.I. 能否也积木化？




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)

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

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

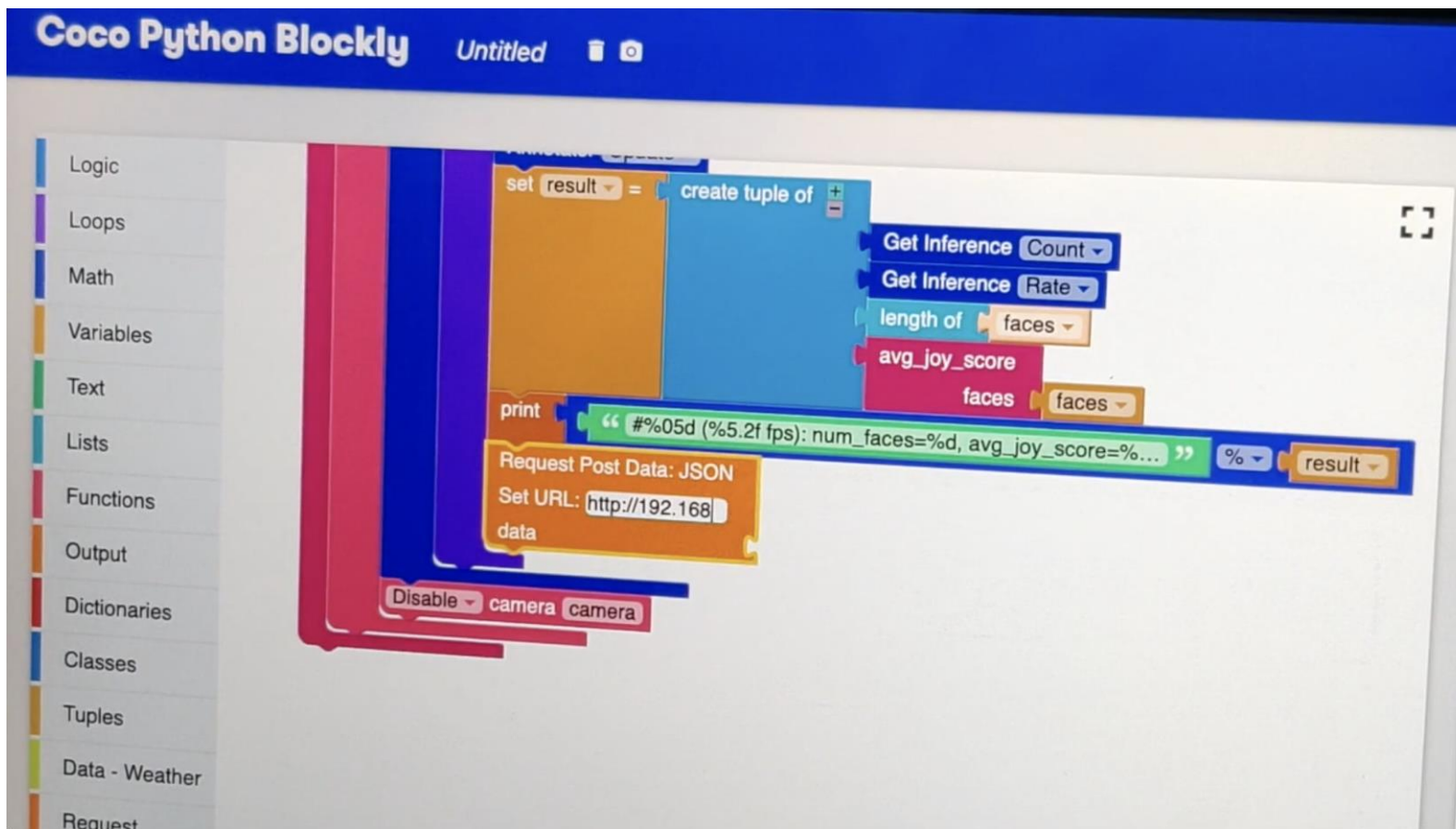
..

📁 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

```
1  import argparse
2
3  from picamera import PiCamera
4
5  from ai.y.vision.inference import CameraInference
6  from ai.y.vision.models import face_detection
7  from ai.y.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)))
```



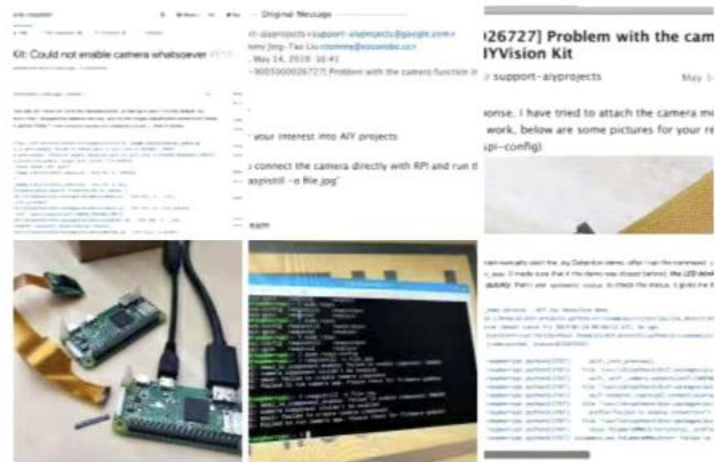


一切都看上去不错，但是…



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。



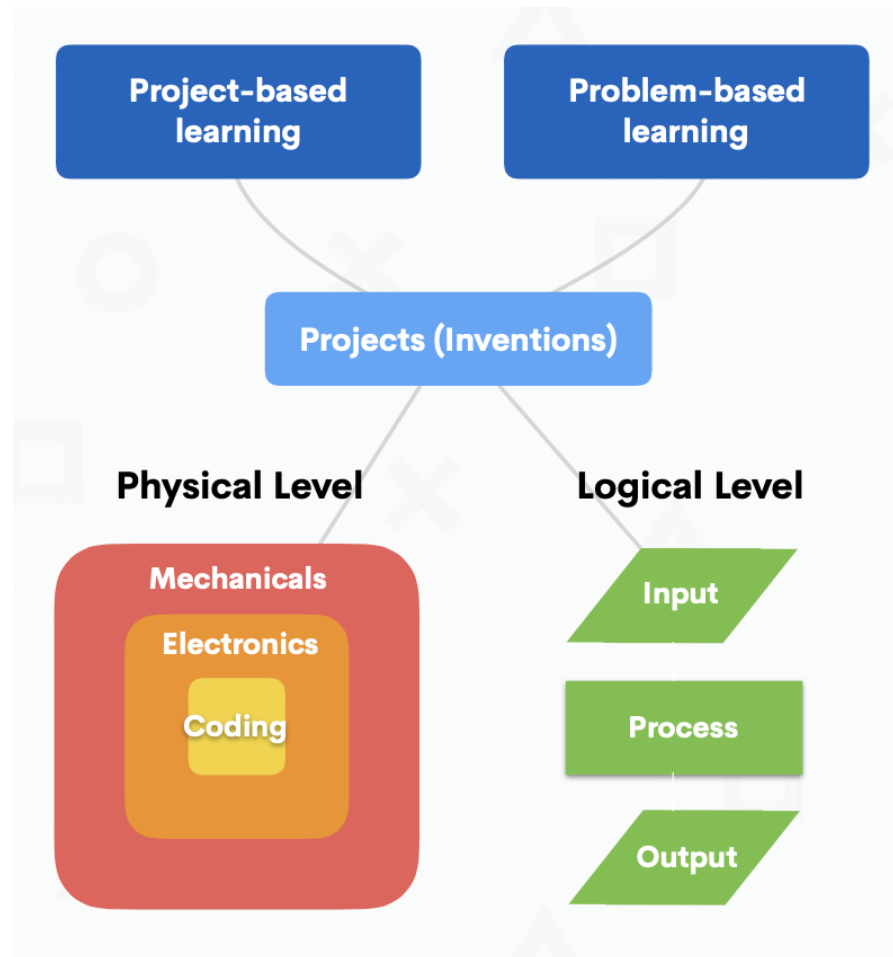
2019年5月14日 20:23 删除

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

- 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



MicroPython : 嵌入式硬件版本 Python

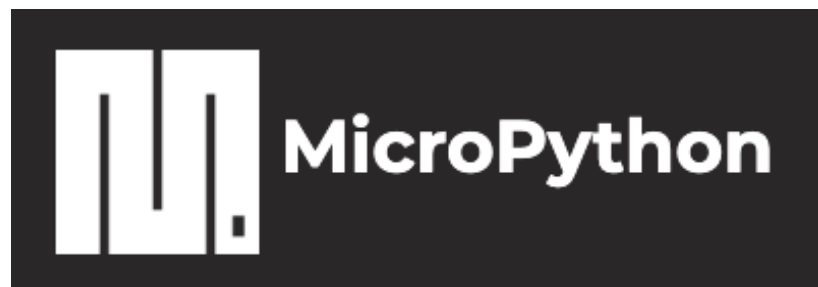


Completely free, open source software

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.

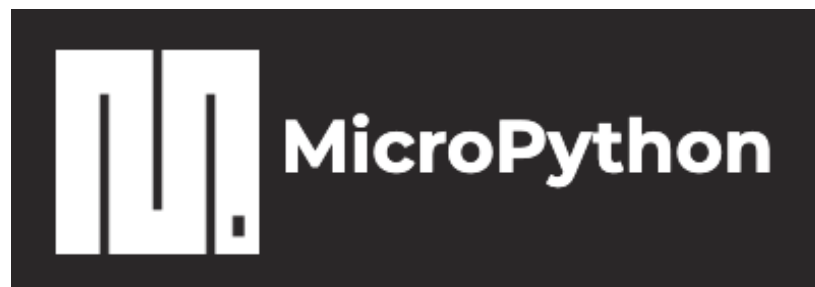
MicroPython is packed full of advanced features such as an interactive prompt, arbitrary precision integers, closures, list comprehension, generators, exception handling and more. Yet it is compact enough to fit and run within just 256k of code space and 16k of RAM.

在精简的嵌入式硬件上运行 Python



STM32, SAMD21, ARM, ESP8266, ESP32...

在精简的嵌入式硬件上运行 Python

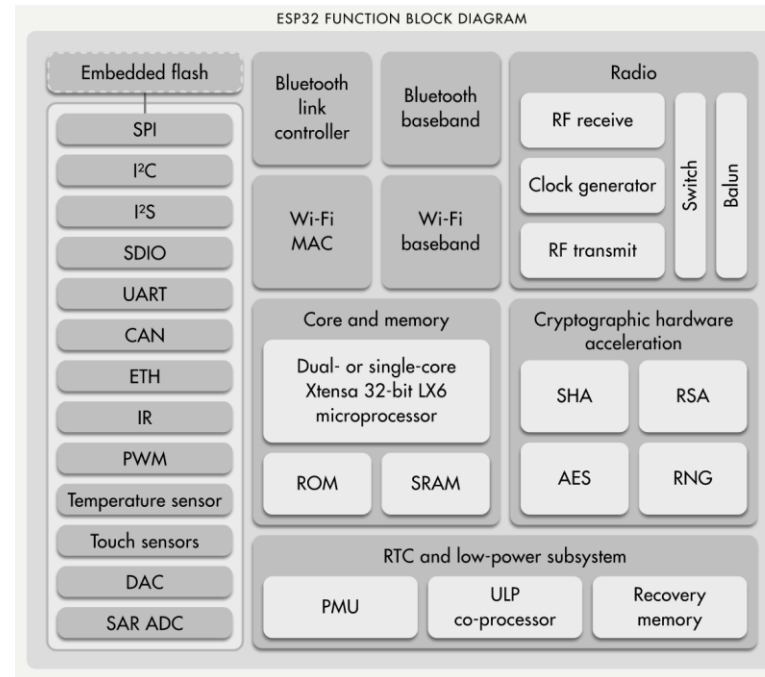


STM32, SAMD21, ARM, ESP8266, ESP32...



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>



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

MicroPython is written in C99, and is highly customizable.

The screenshot shows the GitHub repository for MicroPython. At the top, it displays the repository name 'micropython / micropython' and various statistics: 750 watches, 11.5k stars, and 3.8k forks. Below this, there are navigation tabs for Code, Issues (827), Pull requests (348), Actions, Projects, Wiki, Security, and Insights. The main content area shows a file tree with folders like .github, docs, drivers, examples, extmod, lib, logo, mpy-cross, ports, py, tests, and tools. Each folder is accompanied by a recent commit message and its timestamp. On the right side, there is an 'About' section describing MicroPython as a lean and efficient Python implementation for microcontrollers and constrained systems. It also includes links to the project website, tags for 'micropython', 'python', 'embedded', and 'microcontroller', a 'Readme' link, and a 'MIT License' link. Below the 'About' section, there is a 'Releases' section showing 46 releases, with the latest release being 'New uasyncio module, cod...' on Sep 2. At the bottom right, there is a 'Sponsor this project' button.

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

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))
```

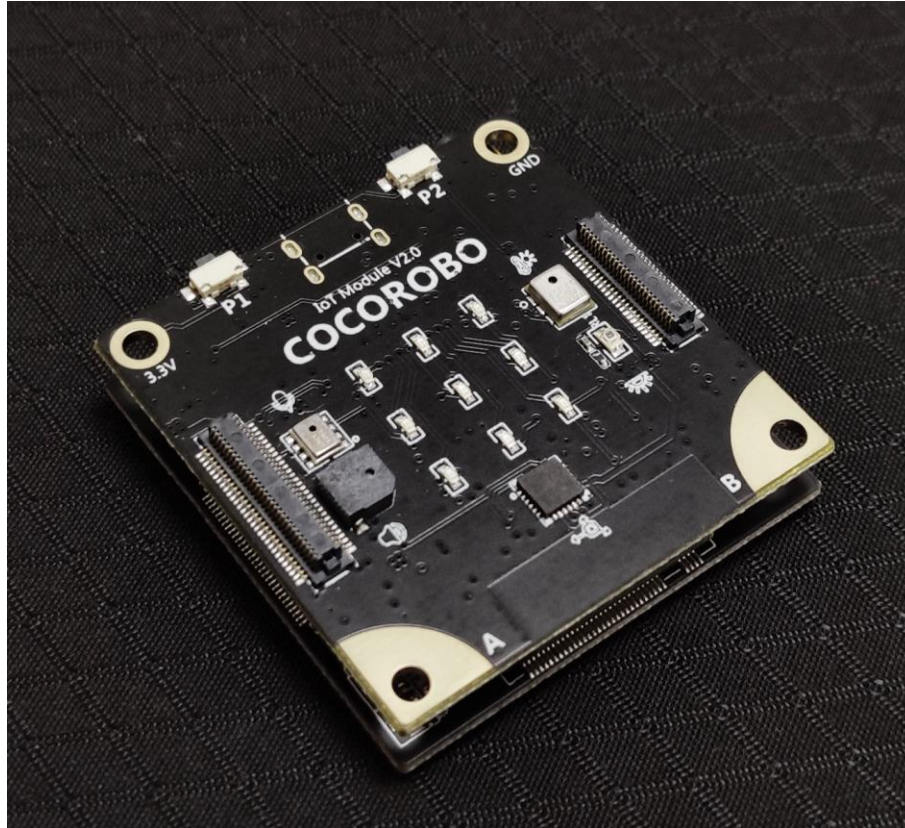
The period is in milliseconds.

Virtual timers are not currently supported on this port.



The screenshot shows a block-based programming interface with a sidebar on the left and a main workspace on the right. The sidebar lists various modules, and the main workspace shows blocks for different categories.

Module	Block
Logic	Button
Loops	When Button P1 Pressed
Math	When Button P1 Released
Variables	Read the Status of Button P1
Text	
Lists	Touch
Dictionary	When Touch Area A is Touched
Tuples	When Touch Area A is Not Touched
Set	Get Touch Area A's Capacitive Variable
Functions	
Input / Output	LED lamp
Files	Set LED # 1 Light Up
Time	Set LED # 1 Light Off
Serial Comm.	Light Up the LED Located in (Column 1, Row 1)
IoT Module	Light Off the LED Located in (Column 1, Row 1)
Basics	
Wi-Fi	Buzzer
Internet	



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

MicroPython



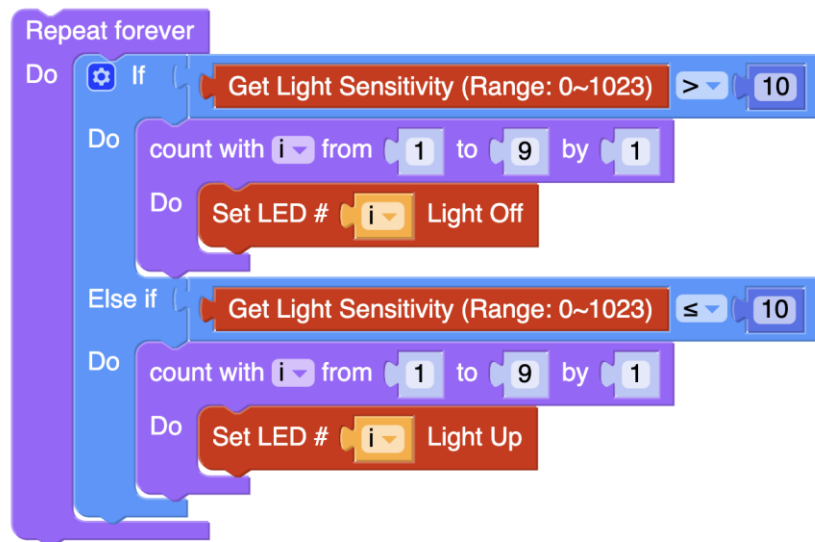
浏览器内上传程序

The screenshot displays the CocoBlockly X Beta web interface. The top navigation bar includes the logo, a project name '未命名项目', a '保存' (Save) button, and various utility icons. A left sidebar lists programming blocks such as '逻辑' (Logic), '循环' (Loop), '数学运算' (Math), '变量' (Variable), '文字' (Text), '列表' (List), '字典' (Dictionary), '元组' (Tuple), '集合' (Set), '函数' (Function), '输入/输出' (Input/Output), '文件' (File), '时间' (Time), '串口通信' (Serial Communication), '人工智能' (AI), and '扩展功能' (Extended Function). The main workspace is currently empty. On the right, a 'Python 源代码' (Python Source Code) editor is open, showing a '串口交互窗' (Serial Interaction Window) with a toggle switch and icons for copy, download, and refresh. Below the editor, the '设备' (Device) section is set to '有线上传' (Wired Upload) with a checked checkbox. A message prompts the user to upload code to the board, and a dropdown menu shows 'COM15'. At the bottom of this section are buttons for '运行' (Run), '上传' (Upload), '更换启动模式' (Change Boot Mode), and '重启设备' (Restart Device). The footer contains 'CocoRobo LTD © 2020 Copyright' and 'CocoBlockly X Beta'.

来看几个简单的示例

- Loops
- Math
- Variables
- Text
- Lists
- Dictionary
- Tuples
- Set
- Functions
- Input / Output
- Files
- Time
- Serial Comm.
- IoT Module**
- Basics
- Wi-Fi
- Internet
- IoT Service

来看个简单的示例：光照感应灯



```
Python Source Code Serial Interaction
32 elif (state == False):
33     if pin == 1: _light_1.value(0)
34     if pin == 2: _light_2.value(0)
35     if pin == 3: _light_3.value(0)
36     if pin == 4: _light_4.value(0)
37     if pin == 5: _light_5.value(0)
38     if pin == 6: _light_6.value(0)
39     if pin == 7: _light_7.value(0)
40     if pin == 8: _light_8.value(0)
41     if pin == 9: _light_9.value(0)
42
43 while True:
44     if (_PHOTONRESISTOR.read()) > 10:
45         for i in range(1, 10):
46             _led_light(False,i)
47     elif (_PHOTONRESISTOR.read()) <= 10:
48         for i in range(1, 10):
49             _led_light(True,i)
50
```

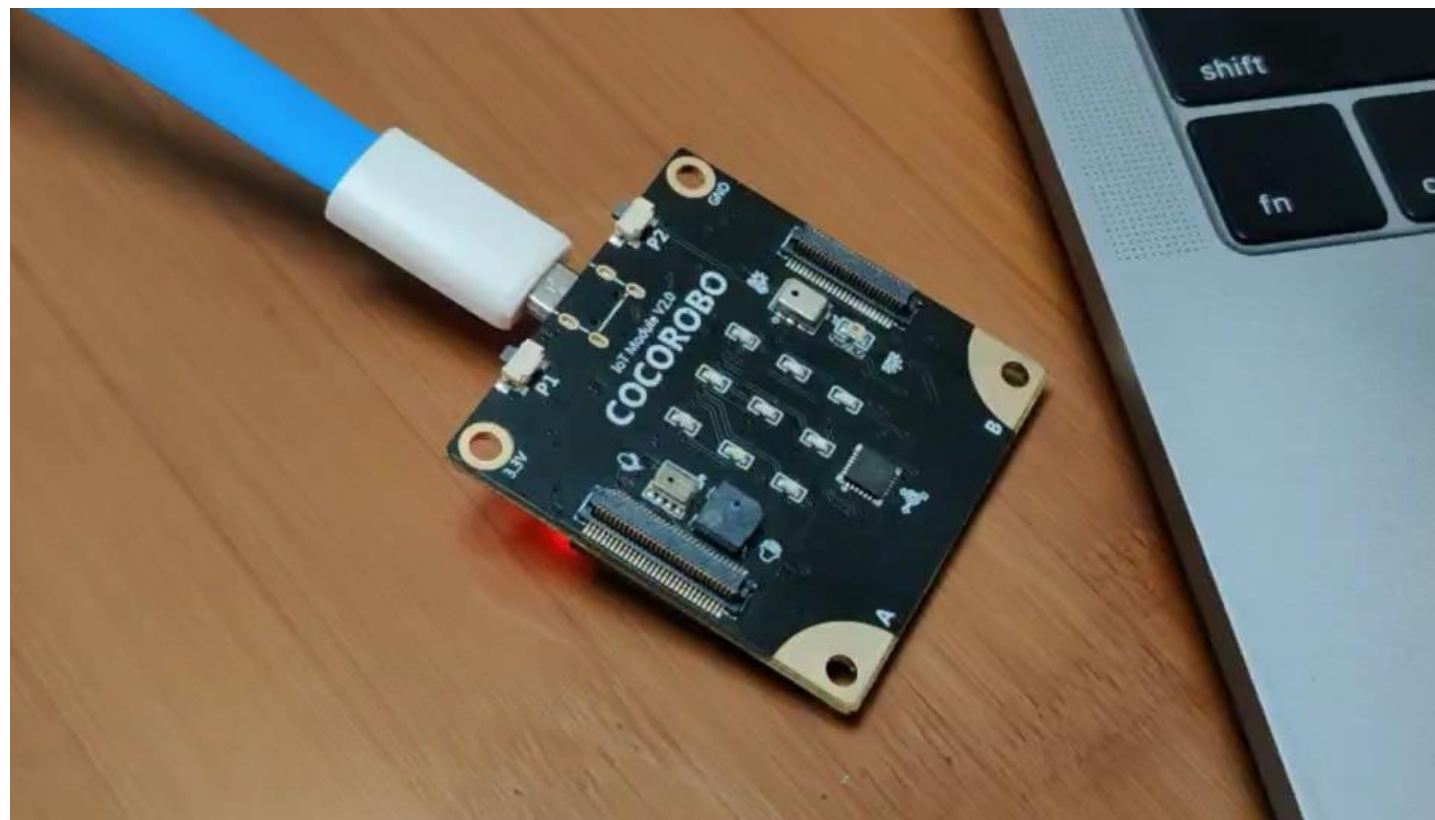
Device USB Upload

Directly upload your code to the electronic modules.

No ports detected

RUN UPLOAD RESET DEVICE

MicroPython



- Loops
- Math
- Variables
- Text
- Lists
- Dictionary
- Tuples
- Set
- Functions
- Input / Output
- Files
- Time
- Serial Comm.
- IoT Module
- Basics
- Wi-Fi
- Internet
- IoT Service

屏幕图案绘制



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

Draw Line

Python Source Code

```
1 import machine, time
2 import st7789
3
4
5
6 spi = machine.SPI(2, baudrate=20000000, polarity=1,
7 tft = st7789.ST7789(spi, 240, 240, reset=machine.Pin
8 tft.init()
9
10 while True:
11     for i in range(0, 242, 5):
12         tft.line(0,i, i,240, st7789.color565(51,102,25
13     for i in range(0, 242, 5):
14         tft.line(i,240, 240,(int((241 - i))), st7789.c
15     for i in range(0, 242, 5):
16         tft.line(240,(int((241 - i))), (int((241 - i)
17     for i in range(0, 242, 5):
18         tft.line((int((241 - i))),0, 0,i, st7789.color
19
```

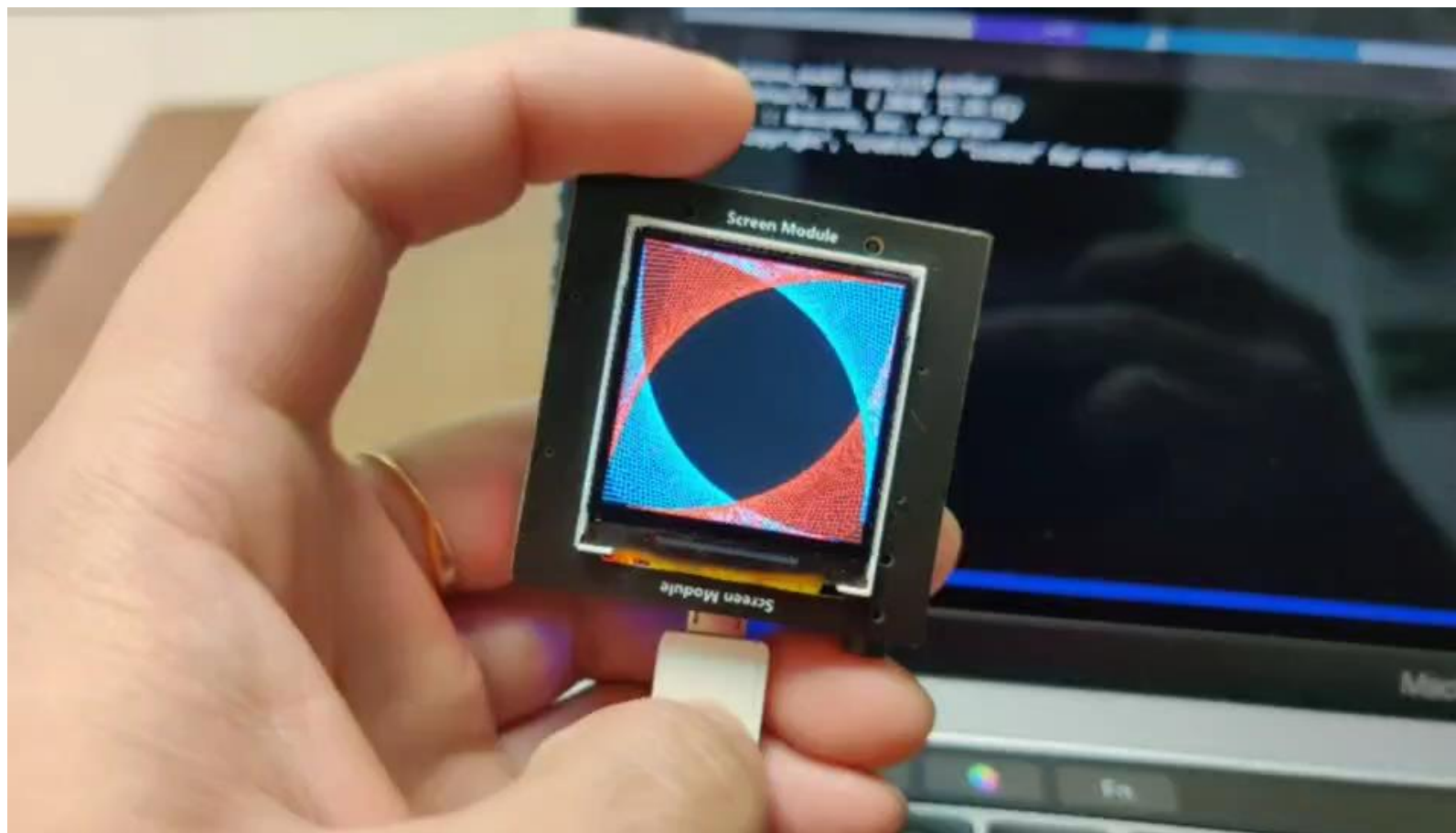
Device USB Upload

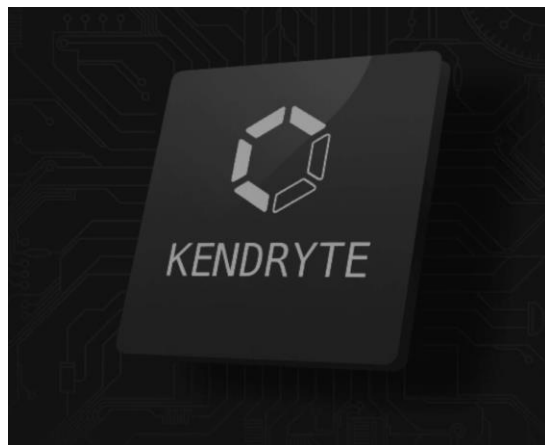
Connecting Module(You are connecting the module. If you do not respond for a long time, please reconnect to USB)

/dev/tty.usbserial-1450

RUN UPLOAD RESET DEVICE

MicroPython



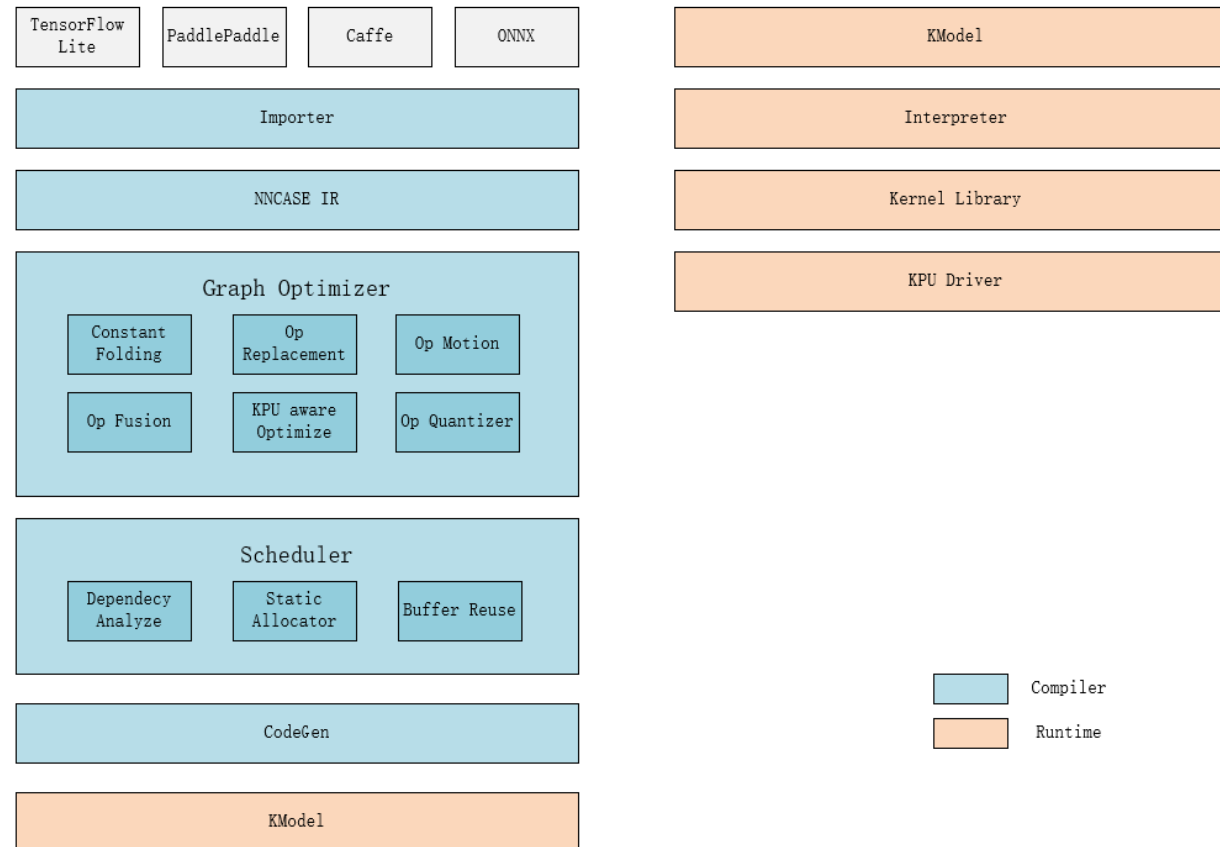


K210 是一款 RISC-V 双核 64 位 CPU，可以仅凭低功耗进行边缘计算，处理机器学习模型（功耗 1W，算力 1TOPS）。



MaixPy 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)
```

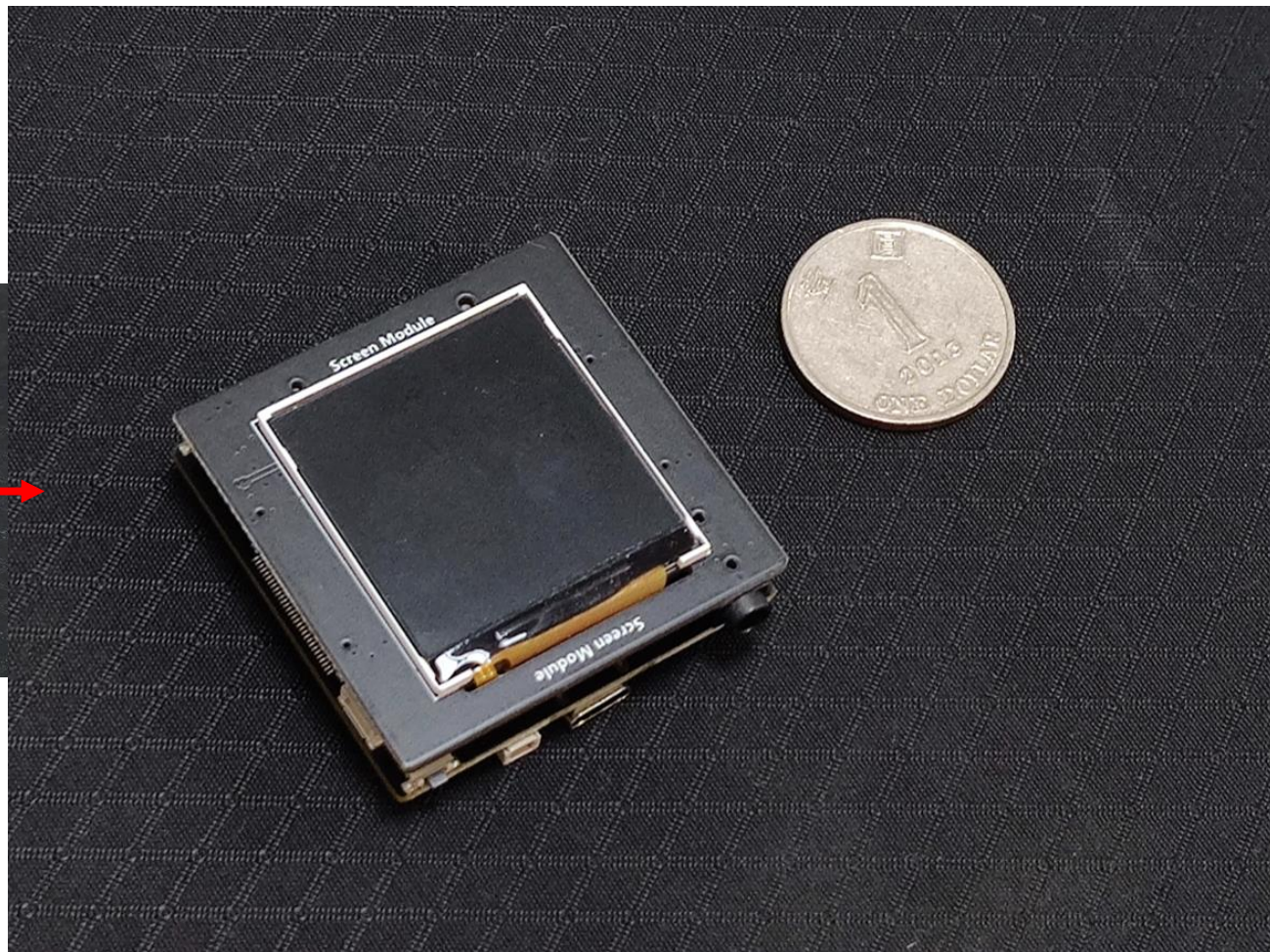
MaixPy & K210

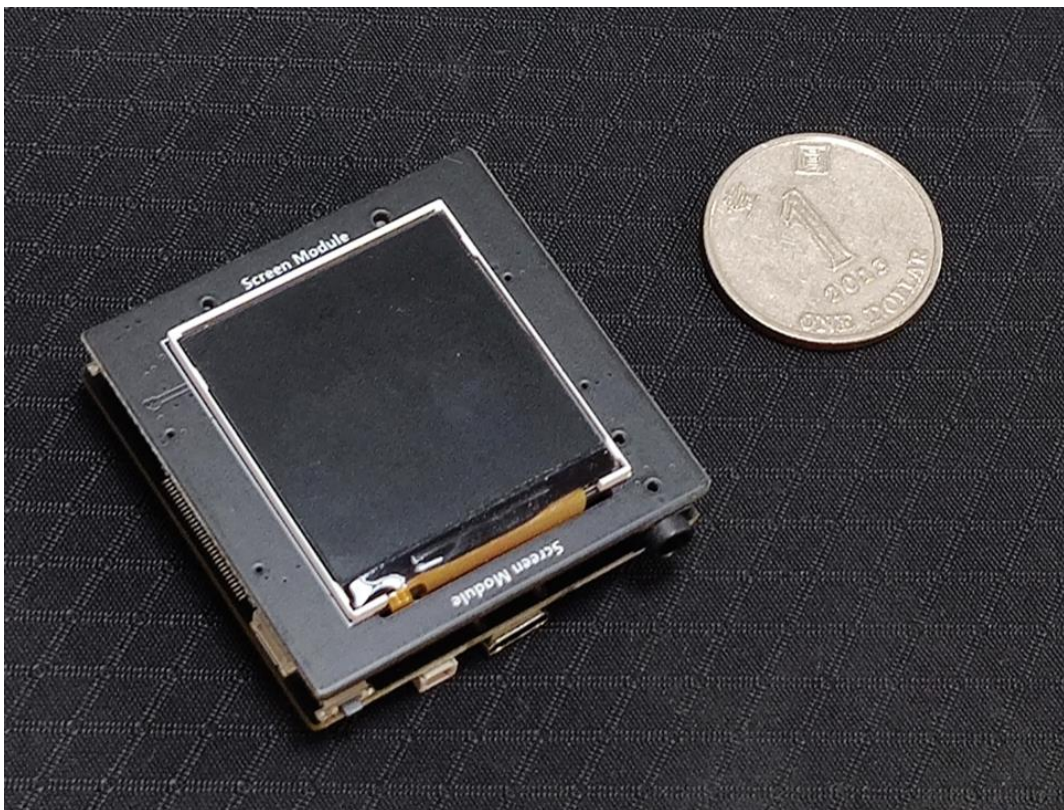
```
import KPU as kpu
task = kpu.load(offset or file_path)
anchor = (1.889, 2.5245, 2.9465, 3.94056, 3.99987, 5.3658, 5.155437, 6.92275, 6.718375, 9.
kpu.init_yolo2(task, 0.5, 0.3, 5, anchor)
```

```
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

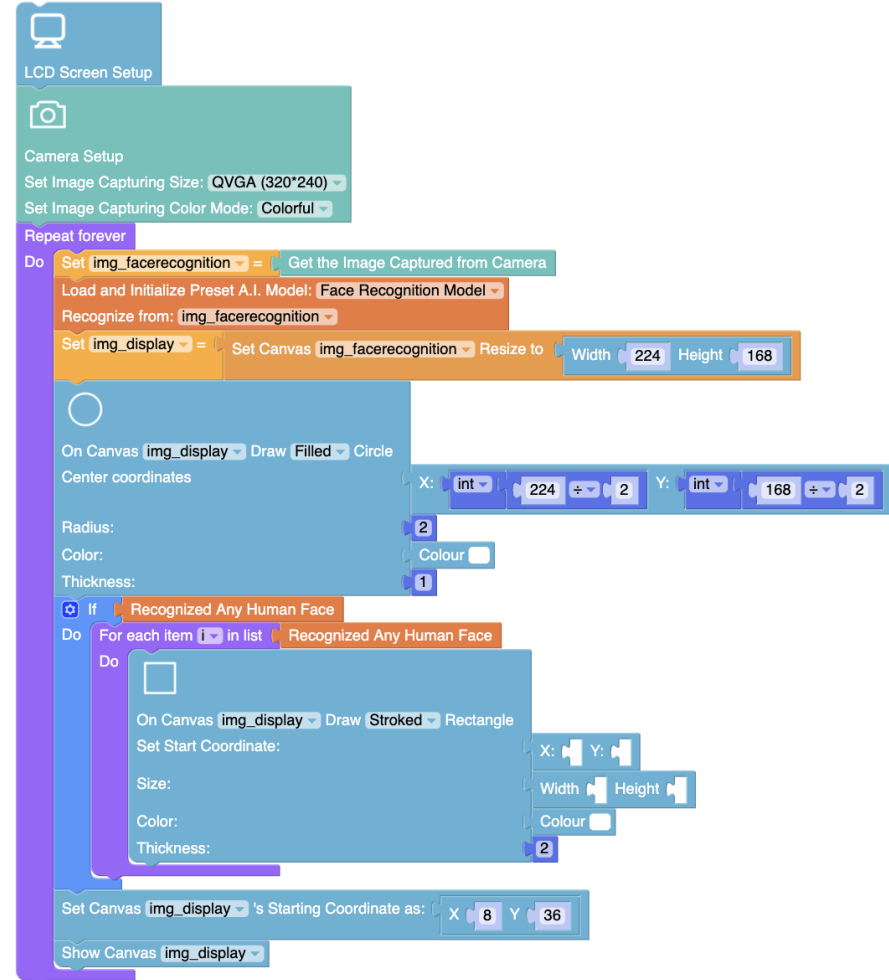




- 计算机视觉：
 - 图形操作：图像属性、尺寸控制等；
 - 圆形识别、矩形识别、直线识别；
 - 二维码识别、条形码识别、AprilCode 识别；
 - 颜色识别、关键点识别
- 人脸识别、物体识别、人脸辨识、语音识别
- 运行自己训练的模型
 - YOLO, Keras, TensorFlow…；
 - ONNX, PaddlePaddle…

MaixPy & K210

```
1 import lcd
2 import image
3 import sensor
4 import KPU as kpu
5
6 try: from cocorobo import display_cjk_string
7 except: pass
8
9 def lcd_draw_string(canvas, x, y, text, color=(255,255,255), font_size=1, scale=1, mono_space=False, auto_wrap=True):
10     try:
11         display_cjk_string(canvas, x, y, text, font_size=font_size, color=color)
12         return canvas
13     except: return canvas.draw_string(x, y, text, color=color, scale=scale, mono_space=mono_space)
14
15 _img_display_x, _img_display_y = 0, 0
16
17 task_facerecognition = kpu.load("/sd/preset/models/preset/face-recognition.kmodel")
18 anchor_face = (1.889, 2.5245, 2.9465, 3.94056, 3.99987, 5.3658, 5.155437, 6.92275, 6.718375, 9.01025)
19 a = kpu.init_yolo2(task_facerecognition, 0.5, 0.3, 5, anchor_face)
20
21 lcd.init(type=2, freq=15000000, width=240, height=240, color=(0,0,0))
22 lcd.rotation(1)
23 lcd.clear(lcd.BLACK)
24 sensor.reset()
25 sensor.set_pixformat(sensor.RGB565)
26 sensor.set_framesize(sensor.QVGA)
27 sensor.set_vflip(0)
28 sensor.set_hmirror(0)
29 sensor.skip_frames(30)
30 sensor.run(1)
31
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(i, color=(255,255,255), thickness=2, fill=False)
41     _img_display_x, _img_display_y = 8, 36
42     lcd.display(img_display, ofx=_img_display_x, ofy=_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

The screenshot displays the MaixPy K210 interface. On the left is a vertical menu with categories: Dictionary, Tuples, Set, Functions, Input / Output, Files, Time, Serial Comm., A.I. Module (highlighted in orange), Basic, Microphone, Image Process, Video Process, Audio Process, Models, Wi-Fi, Extension Modoule, and Third-party Module. The main area shows the 'Shape Recognition' section with several options:

- From Canvas **canvas** Get QR Code Detected Result
- From **i** Get the detected QR Code's **Decoded Text**
- From Canvas **canvas** Get AprilTag Detected Result
- From **i** Get the detected AprilTag's **Decoded Text**
- From Canvas **canvas** Get Data Matrices Detected Result
- From **i** Get the detected Data Matrices' **Decoded Text**
- From Canvas **canvas** Get Barcode Detected Result
- From Canvas **i** Get the detected Barcode's **Decoded Text**

Below the 'Shape Recognition' section is the 'Barcode Recognition' section. The interface also includes a search icon, a zoom control (with +, -, and target icons), and a trash icon on the right side.

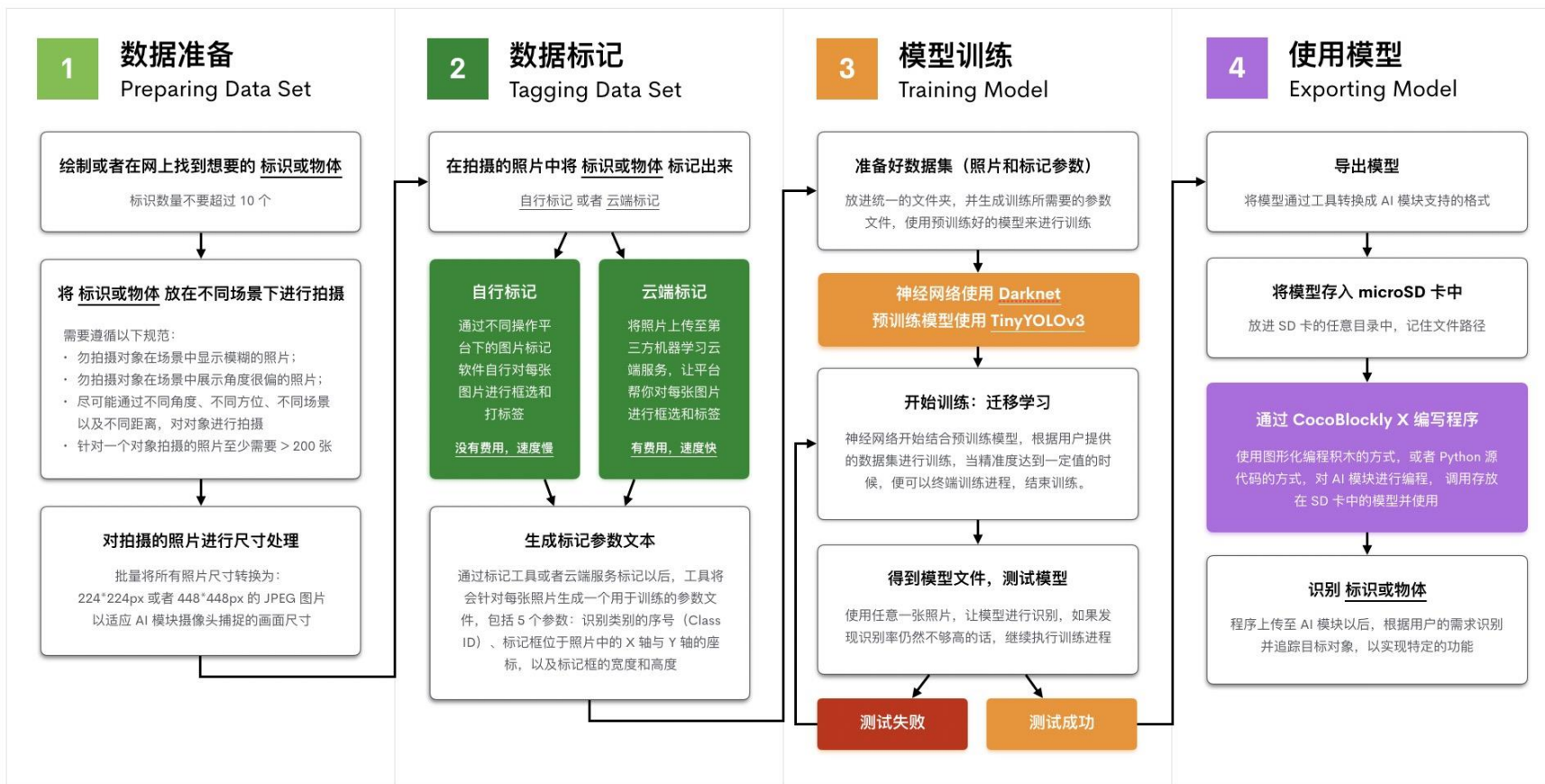
边缘计算（离线模型推理）

- 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

The screenshot displays the MaixPy K210 development environment interface. On the left is a sidebar menu with categories: Dictionary, Tuples, Set, Functions, Input / Output, Files, Time, Serial Comm., A.I. Module (highlighted), Basic, Microphone, Image Process, Video Process, Audio Process, Models, Wi-Fi, Extension Modoule, and Third-party Module. The main workspace shows four AI modules:

- Hand-written Digit Recognition:** Includes a block to 'Load and Initialize Preset A.I. Model: Hand-written Digit Model (MNIST)' and a 'Recognize from: img_mnist' block. Below is an output block showing the digit '3' and a block to 'Get the Hand-written Digit recognition result' with a dropdown for 'number recognized (from 0 to 9)'.
- Object Recognition:** Includes a block to 'Get the recognized Common Object's Name' and an output block 'Recognized Any Common Object'.
- Face Recognition:** Includes a block to 'Get the recognized Humand Face X Coordinate of the bounding box' and an output block 'Recognized Any Human Face'.
- Custom Model Recognition:** Shows a partially visible block at the bottom.

CocoRobo AI Training Center 模型训练全过程



小学体验, 中学深入

- 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

The logo for Google Colab, featuring the word "colab" in a lowercase, rounded, orange font.

Notebook settings

Hardware accelerator
GPU ?

To get the most out of Colab, avoid using a GPU unless you need one. [Learn more](#)

Omit code cell output when saving this notebook

CANCEL SAVE

GPU

Up to Tesla K80 with 12 GB of GDDR5 VRAM, Intel Xeon Processor with two cores @ 2.20 GHz and 13 GB RAM

 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 Settings Profile

+ 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，整個過程分為以下 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

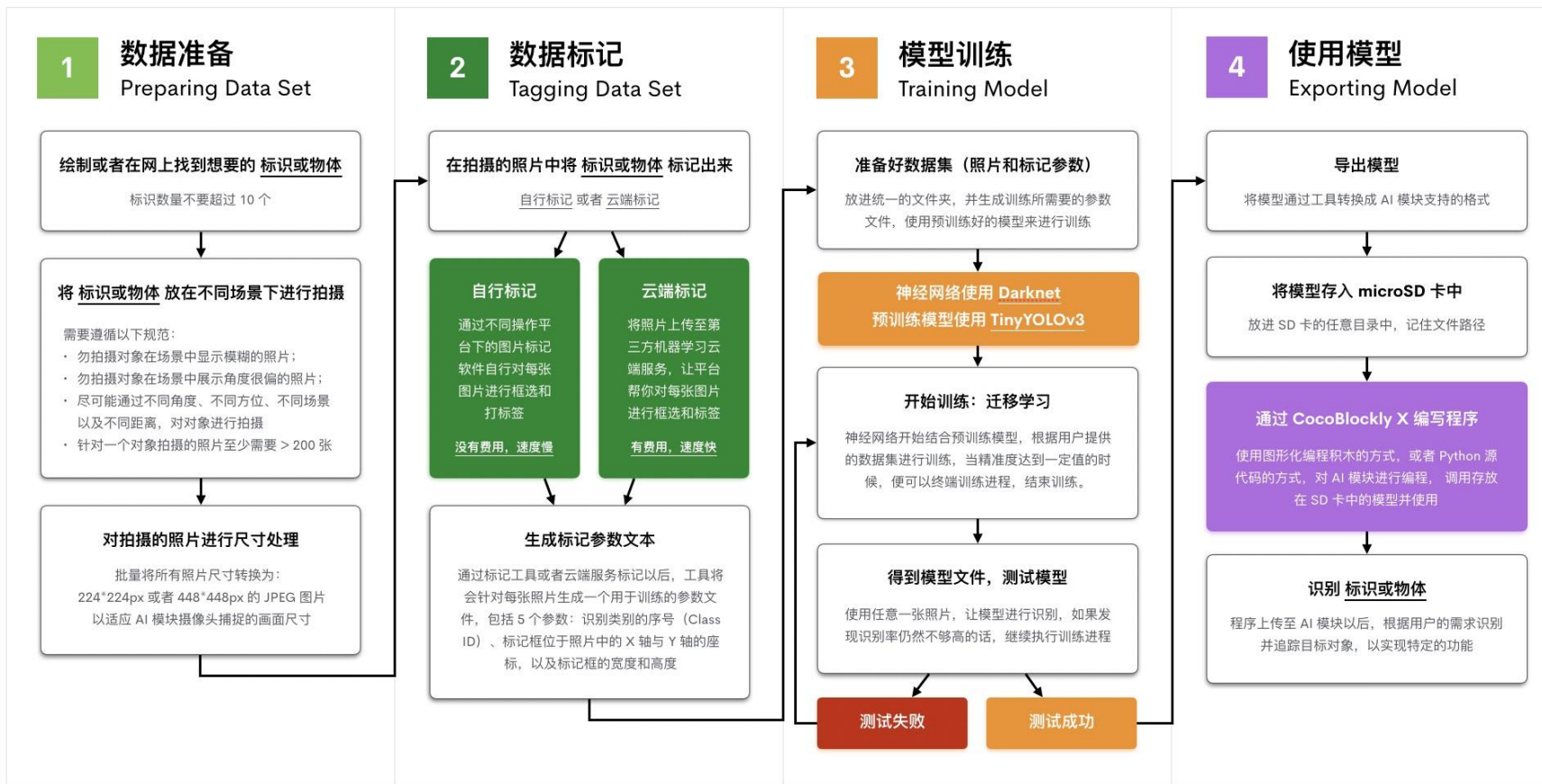
一、採集數據

眾所周知，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

Data science for teams

Deepnote is a new kind of data science notebook. Jupyter-compatible with real-time collaboration and easy deployment.

[Get early access](#) [Join the development team](#)

OpenBayes 建模系统
泛用型机器学习服务, 易学易用

BayesGear 算力容器
GPU 容器服务, 开箱即用

开箱即用
无需系统驱动、环境配置
常用框架、库和公开数据集, 自动建模系统一应俱全
让工程师专注实现业务逻辑

选择工具镜像

- TensorFlow v2.0
- Caffe2 v2
- OpenBayes v1.0.0
- PyTorch v1.2.0
- Apache MXNet v1.4.1
- 查看所有镜像

国土分类四种
Gear ID: 77882516 ● 正常运行 120 TFlops / 32 GB

法律文书自动摘要
Gear ID: 77881394 ● 资源闲置 100 TFlops / 8 GB

+ 新建算力容器

灵活配置
算力算法按需供给, 最低成本获得最高性能计算服务
无论小型实验还是企业级应用
弹性可扩展架构, 随时适应业务需求

Make Almost Anything with Python!

Make with Python



STEP 1: Design it



STEP 2: Code it



**STEP 3: Output
to fabricator**



STEP 4: Press 'print'

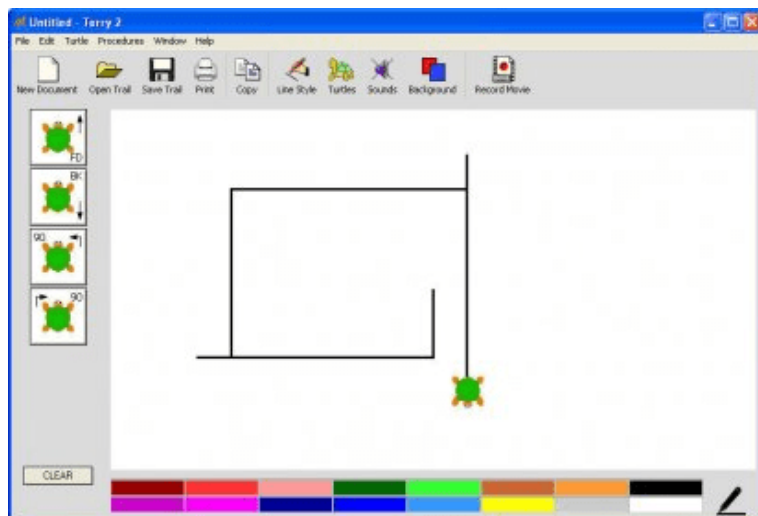


VOILA!

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

Make Almost Anything with Python!

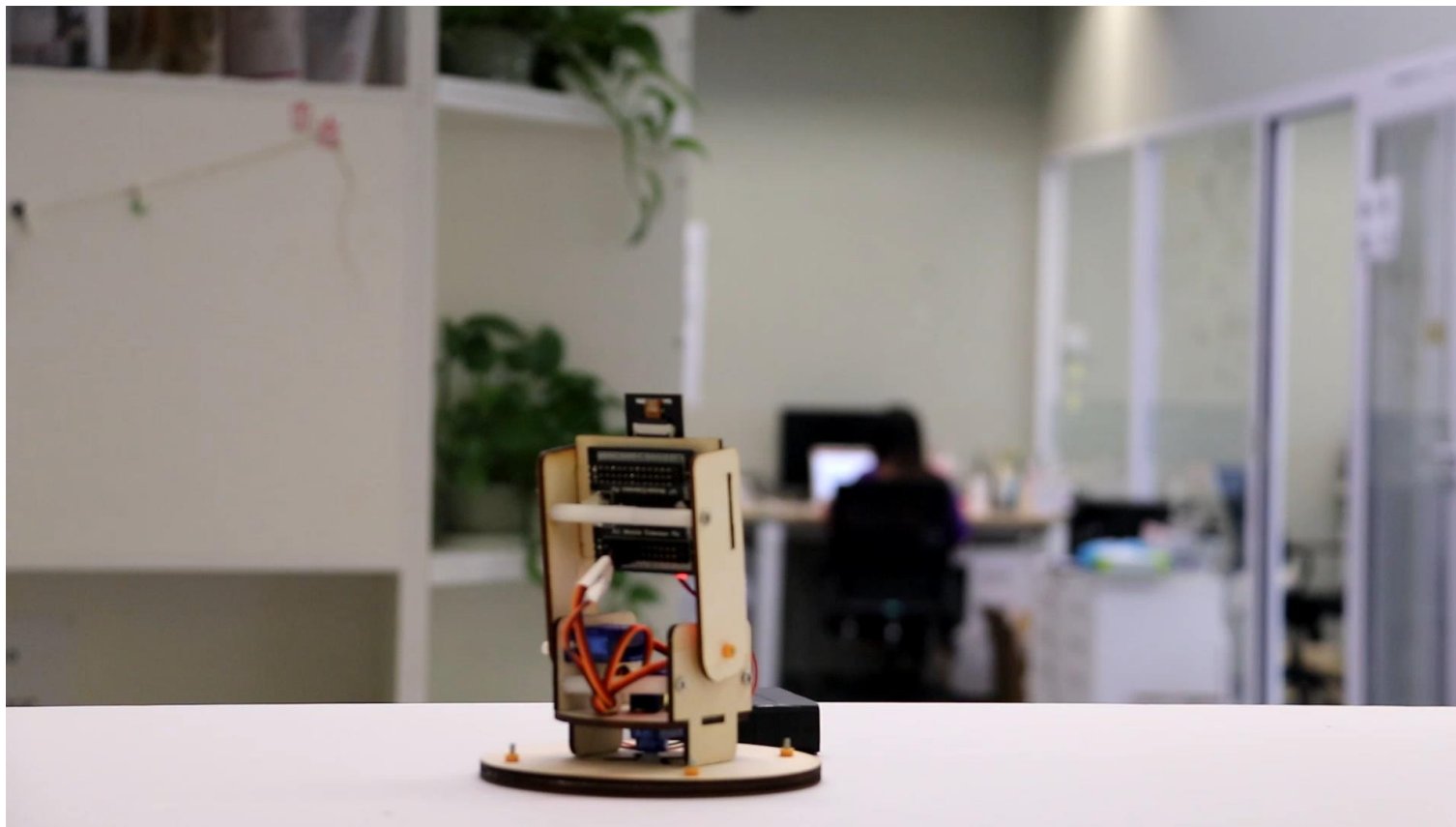
Python 不仅可以很方便的拿来写软件，也可以很方便地拿来控制硬件，
从物联网控制，到边缘计算，这些你想实现的 Idea，
都可以以很低的门槛学习，并最终创作出来



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



垃圾识别自动分类



Thank You!



@technommy



@tommyjtl



tjtl.io



tommy@cocorobo.cc