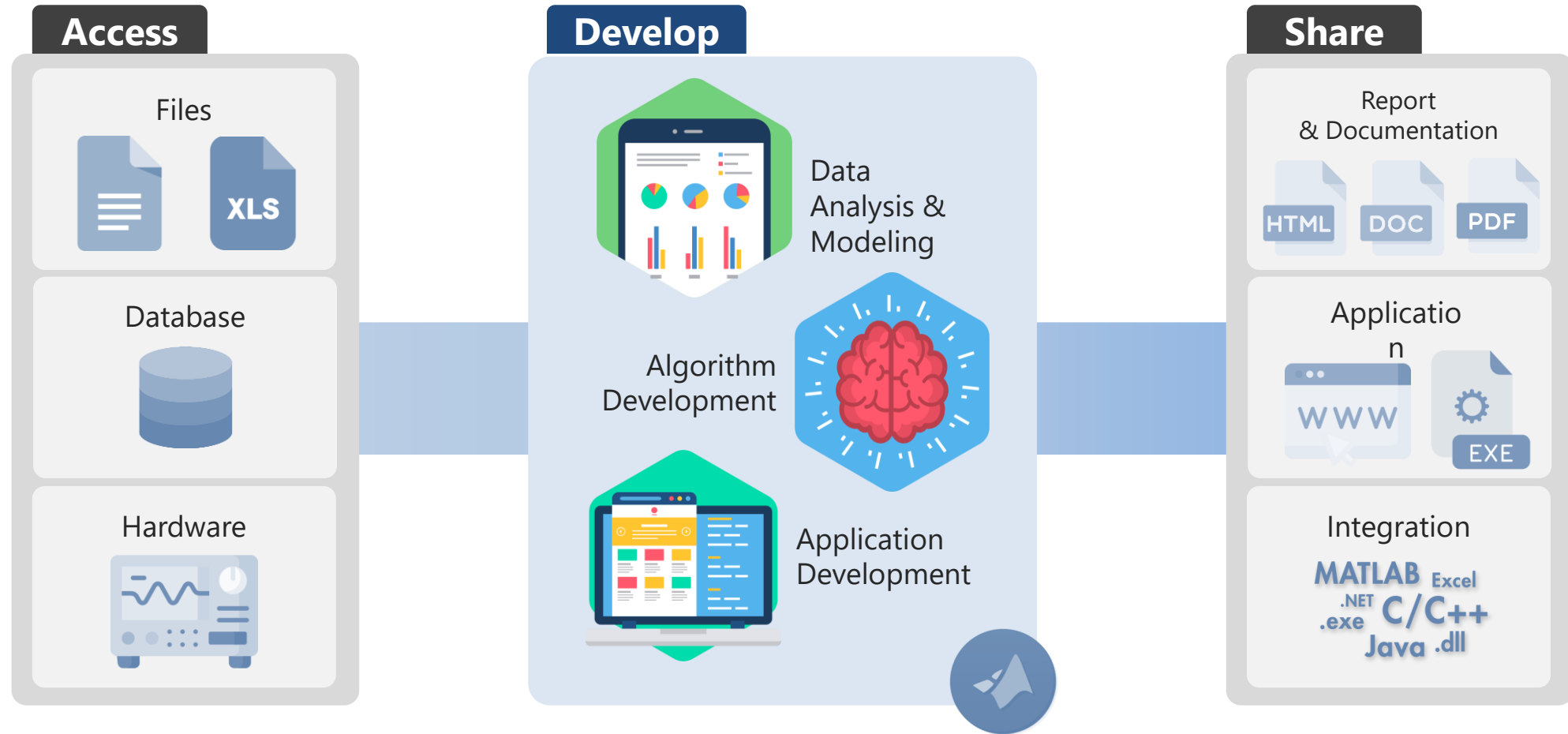


# 廣納百蟻 – MATLAB 和 Python 協作與架構擴展

Application Engineer  
Tim Yeh

# MATLAB 開發流程



# 把不同的積木拼湊起來？



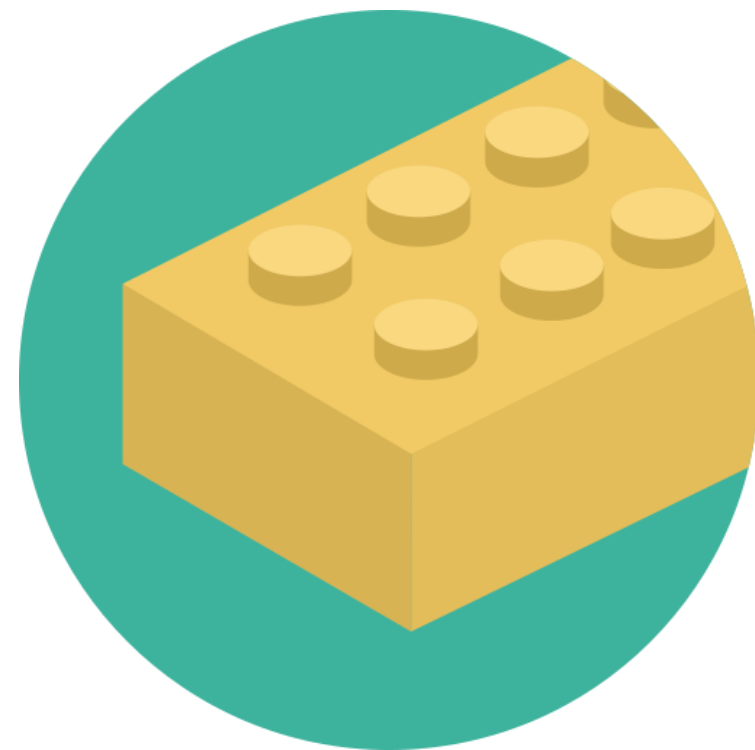
可以想像成不同的程式語言是不同牌子的積木。

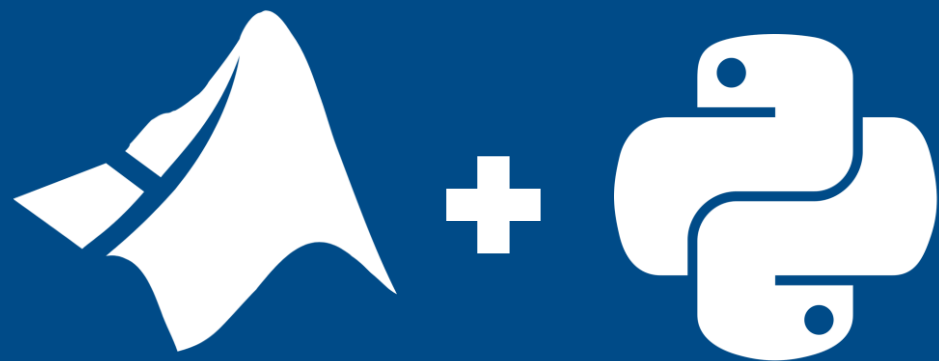


如果別的品牌有比較酷的零件，我能不能組在一起使用？



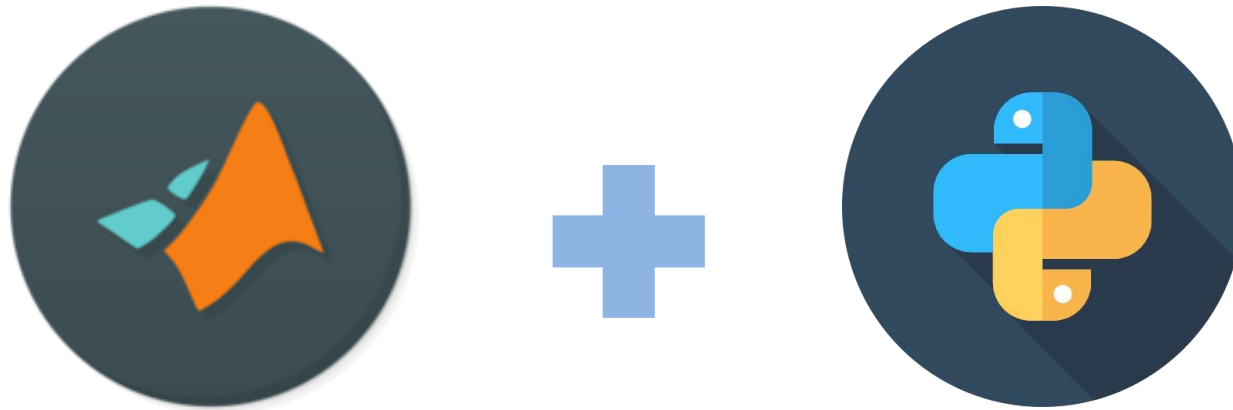
在開發流程的不同階段的抽換成不同的程式語言去實現？





**Intro**

# MATLAB 與 Python 協作



# 常見的問題



I would like to **integrate my colleagues' Python codes** in my MATLAB simulations models, is it possible?

Call Python code directly from MATLAB



Can I use MATLAB models in my Python code if I **don't have any MATLAB installed?**

Yes! You can build Python's libraries from your MATLAB code in one click



We need to put our whole process (MATLAB + Python) in production

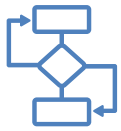
Deploy MATLAB models as micro services and integrate with Python code



# 與Python協作的需求



Need to **integrate** MATLAB code from a colleague



**Facilitate** development by using a simplified MATLAB workflow



Need **functionality** only available in MATLAB (e.g. Simulink)

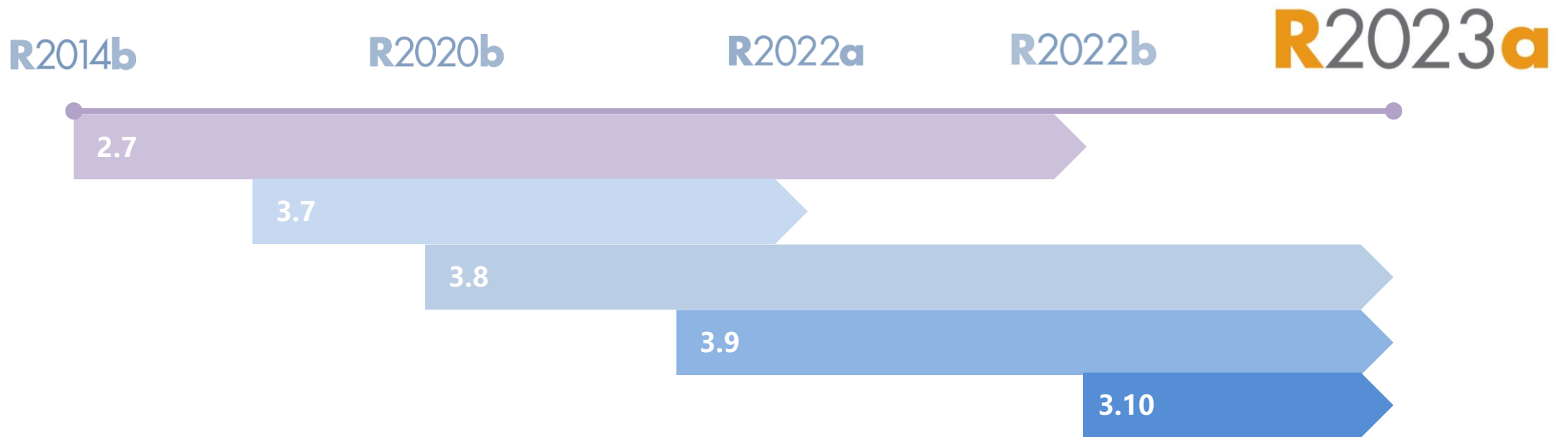


Want to **validate** conclusions by running equivalent MATLAB code



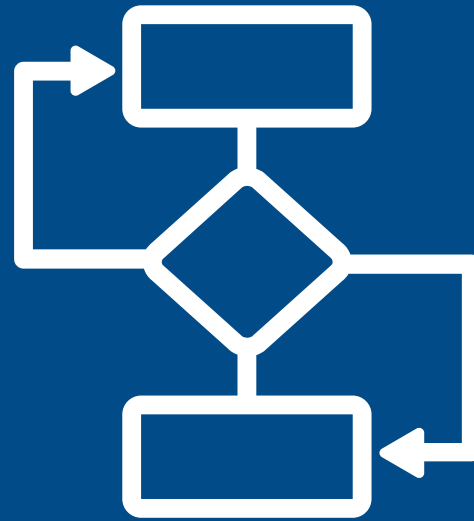
Leverage the work from the MATLAB **community**

# 版本支援



- R2014b+ required
- Python 2.7, 3.7, 3.8, 3.9, 3.10 are supported ([supported list](#))
- Architecture of MATLAB and Python must be the same (64bit)

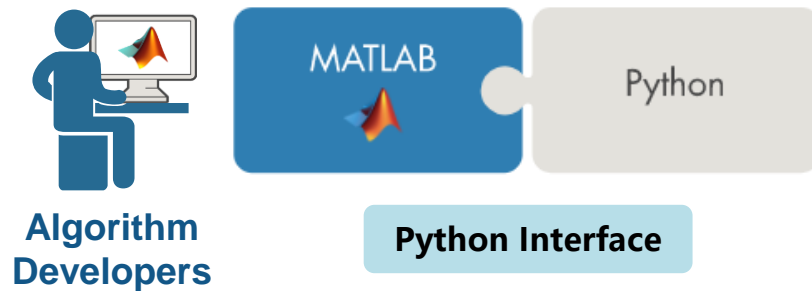




**Co-execution**

# MATLAB + Python

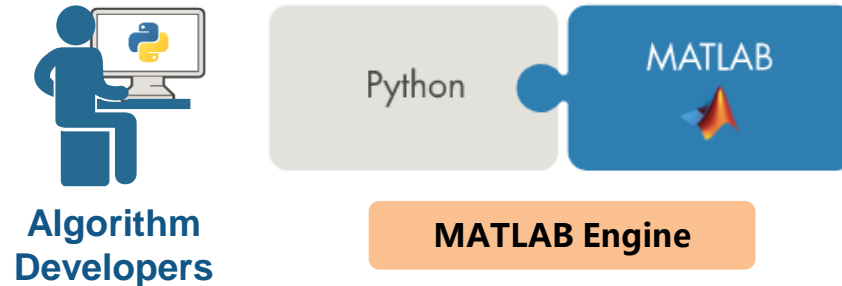
## Calling Python from MATLAB



Already working in MATLAB, and:

- Want to reuse existing Python code
- Need functionality that is only available in Python

## Calling MATLAB from Python



Already working in Python, and:

- Want to reuse existing MATLAB code
- Need functionality available in MATLAB
- Want to collaborate with MATLAB users



**Call Python from MATLAB**

# Call Python from MATLAB

## Set-up your Python environment



```
pyenv|
```

```
ans =
```

```
PythonEnvironment with properties:
```

```
Version: "3.8"
```

```
Executable: "C:\Users\Tim\anaconda3\envs\python38\pythonw.exe"
```

```
Library: "C:\Users\Tim\anaconda3\envs\python38\python38.dll"
```

```
Home: "C:\Users\Tim\anaconda3\envs\python38"
```

```
Status: NotLoaded
```

```
ExecutionMode: OutOfProcess
```



**pyenv**

**LIVE EDITOR**    **INSERT**    **VIEW**

New Open Save Compare Print Export FILE    Go To Find Bookmark NAVIGATE    Text **B I U M** CODE    Code Control Task Refactor    Run Section Run and Advance Run to End SECTION    Run Step Stop RUN

Call\_Python\_from\_MATLAB.mlx

## Call Python from MATLAB

### Check Installed Python Version

```
1 pyenv
```

### Call Python Built-in Function

```
2 % Create Python dictionary object  
3 pyDict = py.dict(X1=3,X2=4)  
4 key = py.list(pyDict.keys());  
5 values = py.list(pyDict.values());  
6  
7 % New MATLAB data type - "dictionary"  
8 matlabDict = dictionary(string(key),double(values))
```

Zoom: 100%    UTF-8    LF    script

24°C 多雲時晴    下午 04:43    2023/5/23

# 在 MATLAB 中調用 Python 的幾種方式



Call Python Library  
in MATLAB



Run Python Code  
Directly in MATLAB



Run Python File  
in MATLAB

# Call Python Library in MATLAB

» **PY.** **Numpy**  
**Pandas**

# Call Python Library in MATLAB

## Call Python Library

```
pyRow = int64(1); % double -> int64 (Integer in python)
pyCol = int64(10); % double -> int64 (Integer in python)
myRand = py.numpy.random.rand(pyRow,pyCol)
```

myRand =

Python `ndarray`:

0.6913    0.4230    0.7248    0.0366    0.1594    0.1084    0.0373    0.584

Use `details` function to view the properties of the Python object.

Use `double` function to convert to a MATLAB array.



# Run Python Code Directly in MATLAB

» **PYRUN** (   )

# Run Python Code Directly in MATLAB

## Run Python Code Directly in MATLAB

```
pyCode = [  
    "Grocery = {'Apple':100, 'Orange':30}";  
    "Items = Grocery.keys()";  
    "Prices = Grocery.values()"];  
[Items, Prices] = pyrun(pyCode, ["Items", "Prices"])
```

Items =

Python `dict_keys` with no properties.

```
dict_keys(['Apple', 'Orange'])
```

Prices =

Python `dict_values` with no properties.

```
dict_values([100, 30])
```



# Run Python Code Directly in MATLAB

» **PYRUNFILE** (   )

# Run Python File in MATLAB

## Run Python File in MATLAB

```
output = pyrunfile('pyAddup.py', 'output', a=1, b=10)|
```

```
output = 102
```



pyrunfile

```
1  ## Demo for MTK Day
2  # Simple add function show "pyrunfile" in MATLAB
3
4  def myAddUp(a,b):
5      a+=1
6      b*=10
7      return a+b
8
9  # Main
10 output = myAddUp(a,b)
```

# Integration with External Language & Models

**Caffe**  
MODELS

**Keras Importer**  
Importer for TensorFlow-Keras Models

**ONNX Converter**  
Export to ONNX model format



Caffe **K** Keras

 **Caffe2**  TensorFlow Lite

 PyTorch

 TensorFlow

# Demo: Import Keras Model

The screenshot displays the MATLAB Live Editor and Deep Network Designer interface. The Live Editor on the left shows the following code and output:

```
1 KerasModel = importTensorFlowNetwork('./digitsDAG...  
Importing the saved model...  
Translating the model, this may take a few minutes...  
Finished translation. Assembling network...  
Warning: Saved Keras networks do not include classes. Cla...  
output layer of the network. To specify classes, use the...  
Warning: Network issues detected.  
Caused by:  
Layer 'ClassificationLayer_activation_1': Empty Class...  
will be set to categorical(1:N), where N is the number...  
Import finished.
```

2 Configure model with DeepNetwork Designer  
deepNetworkDesigner

3 Data Path = 'C:\Program Files\MATLAB\R2022a\toolbo...  
4 imds = imageDatastore(DataPath, 'IncludeSubfolders...

The Deep Network Designer window on the right shows a neural network architecture with the following layers:

- input\_1 imageInputLayer
- conv2d\_1 convolution2dL... (left branch)
- conv2d\_2 convolution2dL... (right branch)
- conv2d\_1\_relu reluLayer (left branch)
- conv2d\_2\_relu reluLayer (right branch)
- max\_pooling2... maxPooling2dL... (left branch)
- max\_pooling2... maxPooling2dL... (right branch)
- flatten\_1 FlattenCStyleL... (left branch)
- flatten\_2 FlattenCStyleL... (right branch)

The Layer Library on the left lists various layer types such as softmaxLayer, sigmoidLayer, classificationLayer, regressionLayer, rpnSoftmaxLayer, rcnnBoxRegress..., rpnClassification..., pixelClassificatio..., and dicePixelClassifi... The Properties Overview panel on the right shows a warning icon and the text "Classification".

# Demo: Import Keras Model

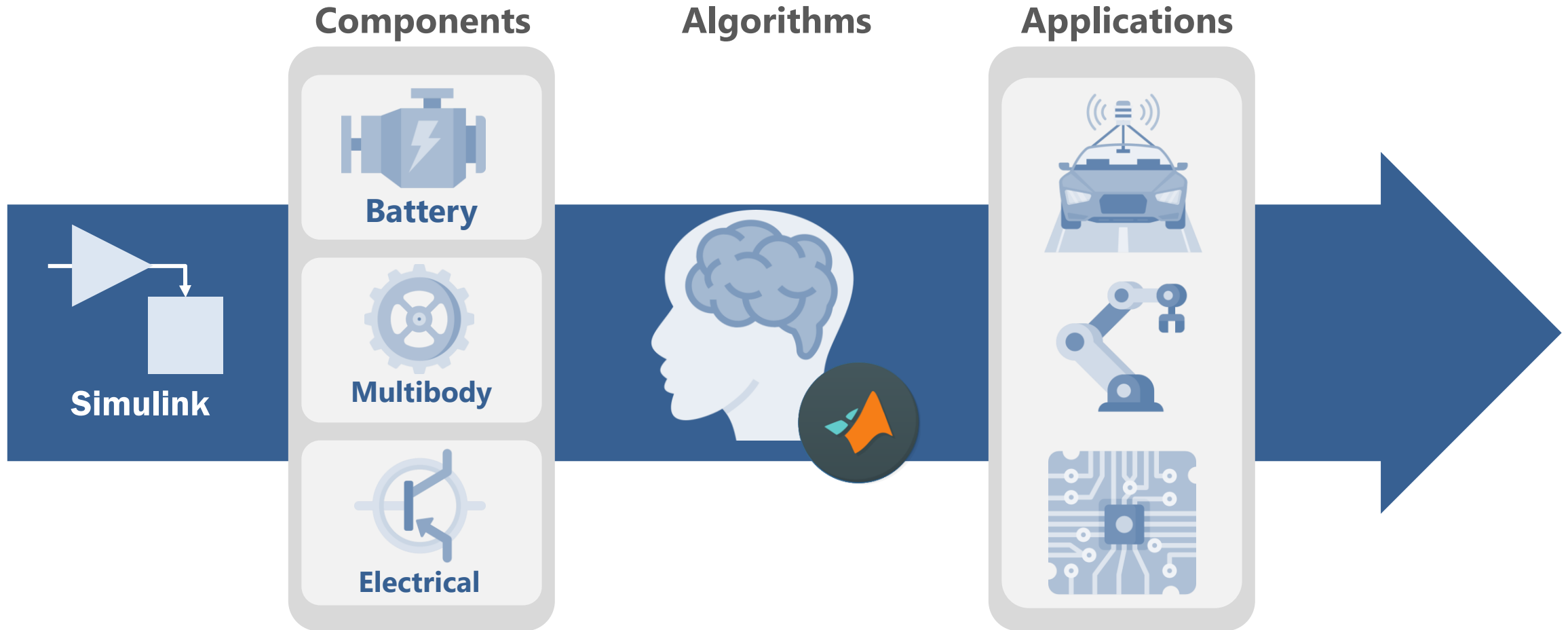
The screenshot displays the MATLAB Live Editor interface for a script titled 'ImportModelFromKerasmlx.mlx'. The editor is divided into sections for different stages of the workflow:

- Import Keras Model**: Line 1 contains the code `KerasModel = importTensorFlowNetwork('./digitsDAGnet');`
- Configure model with DeepNetwork Designer**: Line 2 contains the code `deepNetworkDesigner`.
- Data Preprocessing**: Lines 3-11 contain the following code:

```
DataPath = 'C:\Program Files\MATLAB\R2022a\toolbox\nnet\ndemos\ndatasets\DigitDataset\';  
imds = imageDatastore(DataPath,'IncludeSubfolders',true,'LabelSource','foldernames');  
[trainimds,testimds] = splitEachLabel(imds,0.7);  
options = trainingOptions('adam',...  
    'InitialLearnRate',0.01,...  
    'Verbose',0,...  
    'ExecutionEnvironment','gpu',...  
    'Plots','training-progress',...  
    'Shuffle',"every-epoch");
```
- Model Training**: This section is currently empty.

The bottom status bar shows 'Zoom: 110%', 'UTF-8', 'LF', and 'script'.

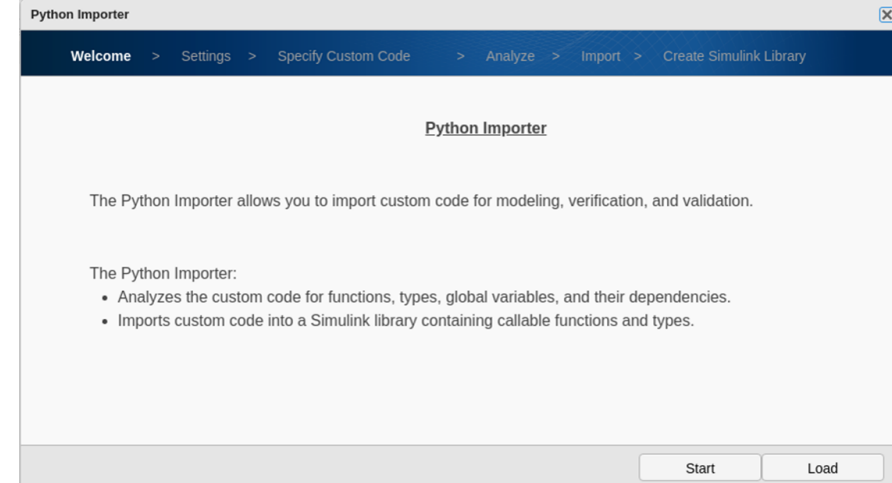
# Simulink with Python





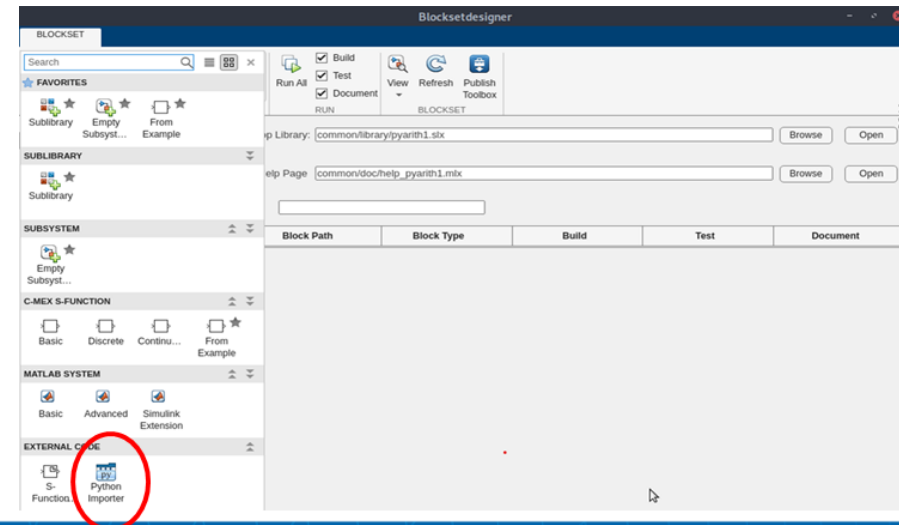
# Python Importer in Simulink

- Import a Python function into a Simulink model for simulation
  - Integrate a custom Python function into a Simulink model as library block
  - Integrate a package of Python functions with each python function corresponding to a library block
- Integrated with Blockset Designer



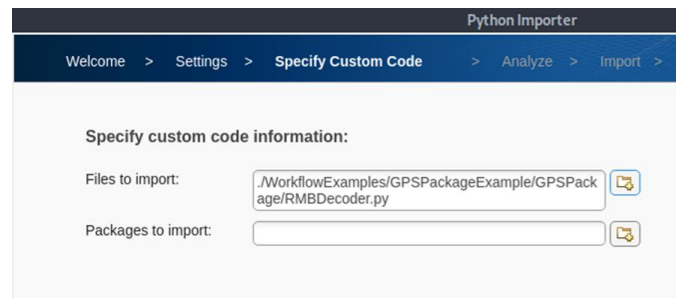
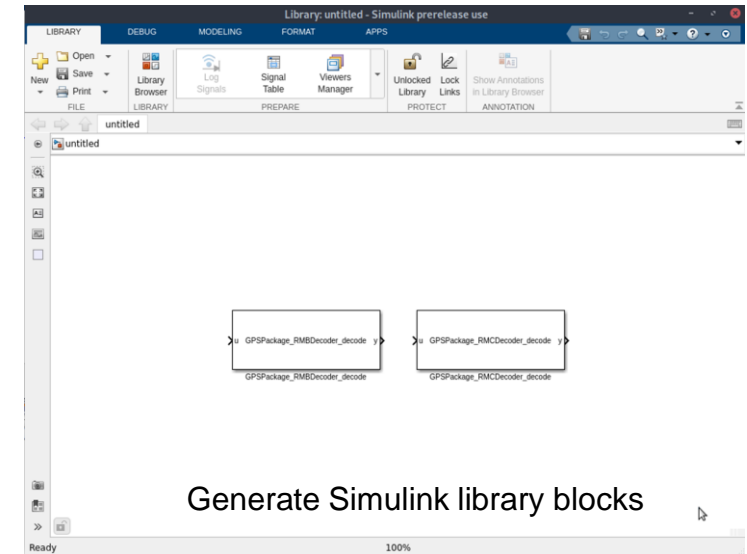
Launch GUI from MATLAB Command line

- `>> obj = Simulink.PythonImporter()`
- `>> obj.view()`

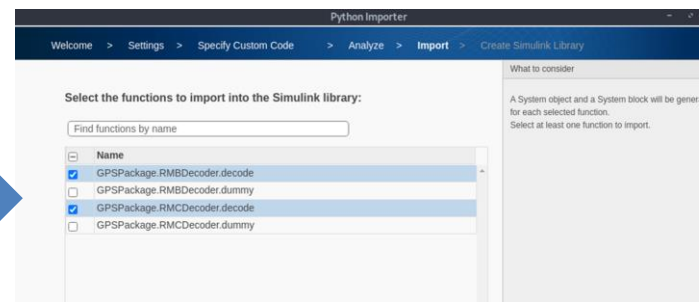


# Python Importer in Simulink

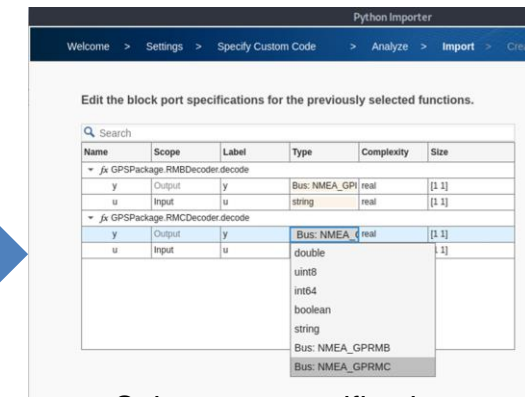
- Support Python lists, tuples, and dictionaries data types
- Support Python functions whose inputs could be Simulink signal inputs or block parameters



Select import file/package



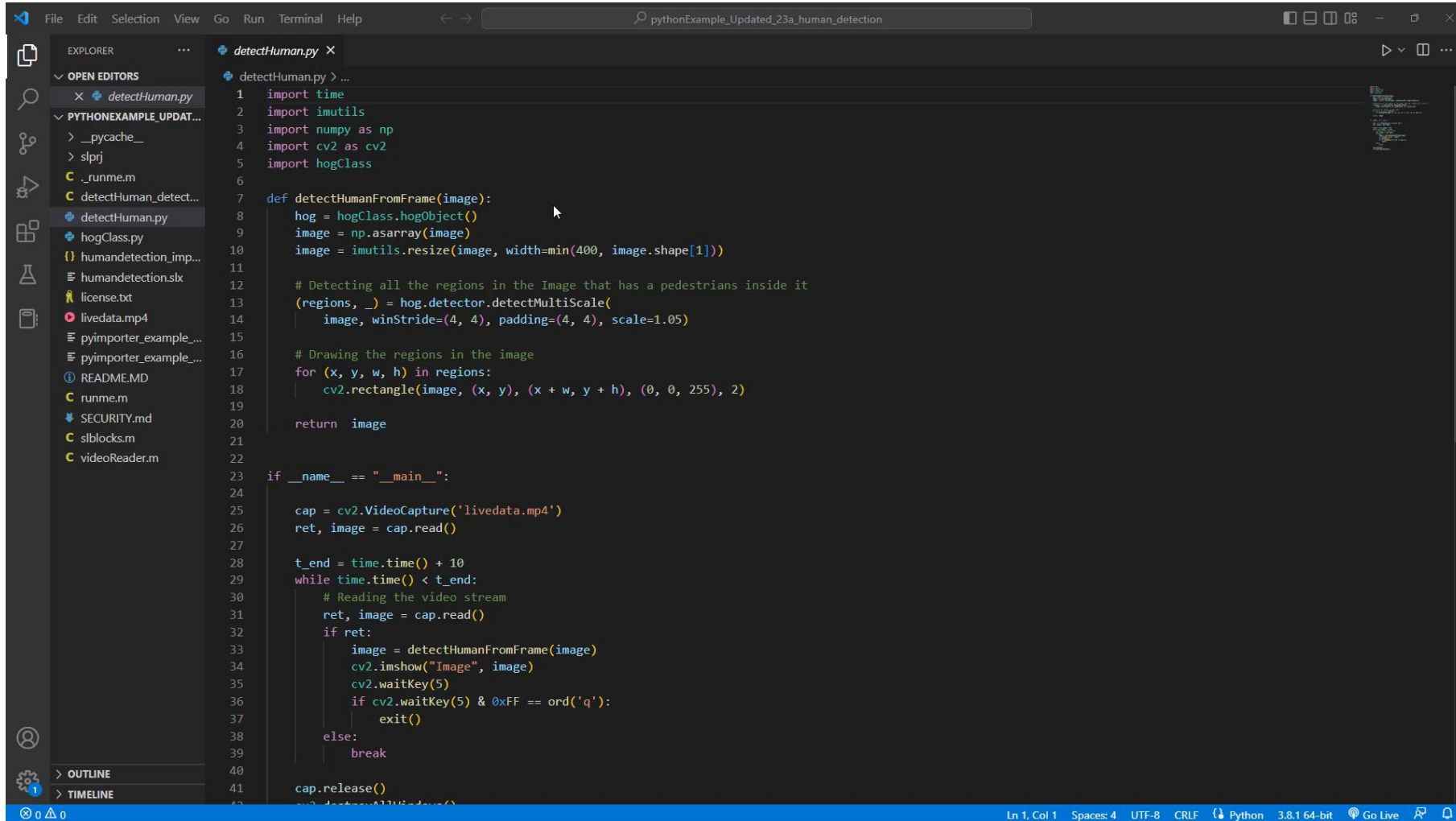
Select import functions



Select port specifications



# Python Importer in Simulink



The image shows a code editor window with a dark theme. The title bar reads "pythonExample\_Updated\_23a\_human\_detection". The Explorer panel on the left shows a project structure with files like "detectHuman.py", "hogClass.py", and "videoReader.m". The main editor area displays the following Python code:

```
1 import time
2 import imutils
3 import numpy as np
4 import cv2 as cv2
5 import hogClass
6
7 def detectHumanFromFrame(image):
8     hog = hogClass.hogObject()
9     image = np.asarray(image)
10    image = imutils.resize(image, width=min(400, image.shape[1]))
11
12    # Detecting all the regions in the Image that has a pedestrians inside it
13    (regions, _) = hog.detector.detectMultiScale(
14        image, winStride=(4, 4), padding=(4, 4), scale=1.05)
15
16    # Drawing the regions in the image
17    for (x, y, w, h) in regions:
18        cv2.rectangle(image, (x, y), (x + w, y + h), (0, 0, 255), 2)
19
20    return image
21
22
23 if __name__ == "__main__":
24
25    cap = cv2.VideoCapture('livedata.mp4')
26    ret, image = cap.read()
27
28    t_end = time.time() + 10
29    while time.time() < t_end:
30        # Reading the video stream
31        ret, image = cap.read()
32        if ret:
33            image = detectHumanFromFrame(image)
34            cv2.imshow("Image", image)
35            cv2.waitKey(5)
36            if cv2.waitKey(5) & 0xFF == ord('q'):
37                exit()
38        else:
39            break
40
41    cap.release()
```

The status bar at the bottom indicates "Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.8.1 64-bit Go Live".

MAI LAB R2023a - prerelease use

HOME PLOTS APPS EDITOR PUBLISH VIEW

New Open Save Compare Print Go To Find Refactor Analyze Insert Property Insert Method Input / Output System Block Run Section Run and Advance Run Step Stop

FILE NAVIGATE CODE ANALYZE SYSTEM OBJECT SECTION RUN

C:\Users\weiwuli\OneDrive - MathWorks\Desktop\23a\_TKF\pythonExample\_Updated\_23a\_human\_detection

Current Folder

- videoReader.m
- slblocks.m
- SECURITY.md
- runme.m
- README.MD
- pyimporter\_example\_23a.slxc
- pyimporter\_example\_23a.slxc.autosave
- pyimporter\_example\_23a.slxc
- livedata.mp4
- license.txt
- humandetection\_import\_autosave.json
- humandetection.slxc.bak
- humandetection.slxc
- hogClass.py
- detectHuman\_detectHumanFromFrame.m
- detectHuman.py
- .\_runme.m
- slprj
- \_\_pycache\_\_

Editor - C:\Users\weiwuli\OneDrive - MathWorks\Desktop\23a\_TKF\pythonExample\_Updated\_23a\_human\_detection\detectHuman\_detectHumanFromFrame.m

```

1 | classdef detectHuman_detectHumanFromFrame < matlab.System
2 | % Python def detectHumanFromFrame imported from python module detectHuman.py
3 | properties
4 | end
5 | methods(Access = protected)
6 | function validateInputsImpl(obj, varargin)
7 | if ~isempty(varargin{1})
8 |     validateattributes(varargin{1}, {'uint8'}, {'size',[480 640 3]});
9 | end
10 | end
11 | function setupImpl(obj)
12 |     py.importlib.import_module('detectHuman');
13 | end
14 | function [image] = stepImpl(obj,image)
15 |     image = uint8(py.detectHuman.detectHumanFromFrame(image));
16 | end
17 | function varargout = getOutputDataTypeImpl(obj)
18 |     varargout{1} = 'uint8';
19 | end
20 | function varargout = getOutputSizeImpl(obj)
21 |     varargout{1} = [300 400 3];
22 | end
23 | function varargout = isOutputComplexImpl(obj)
24 |     varargout{1} = false;
25 | end

```

Command Window

```

OutputFolder:
CustomCode: [1x1 Simulink.PythonImporter.CustomCode]
ParseInfo: [1x1 Simulink.PythonImporter.ParseInfo]
Options: [1x1 Simulink.PythonImporter.Options]

>> obj.view
>> obj.view
fx >>

```

pyimporter\_example\_23a.slxc (Simulink Model)

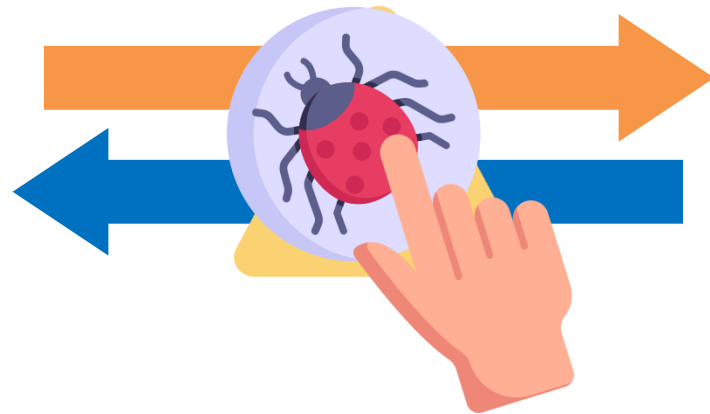
**Model version:**  
9.2

**Saved in Simulink version:**  
R2023a Prerelease

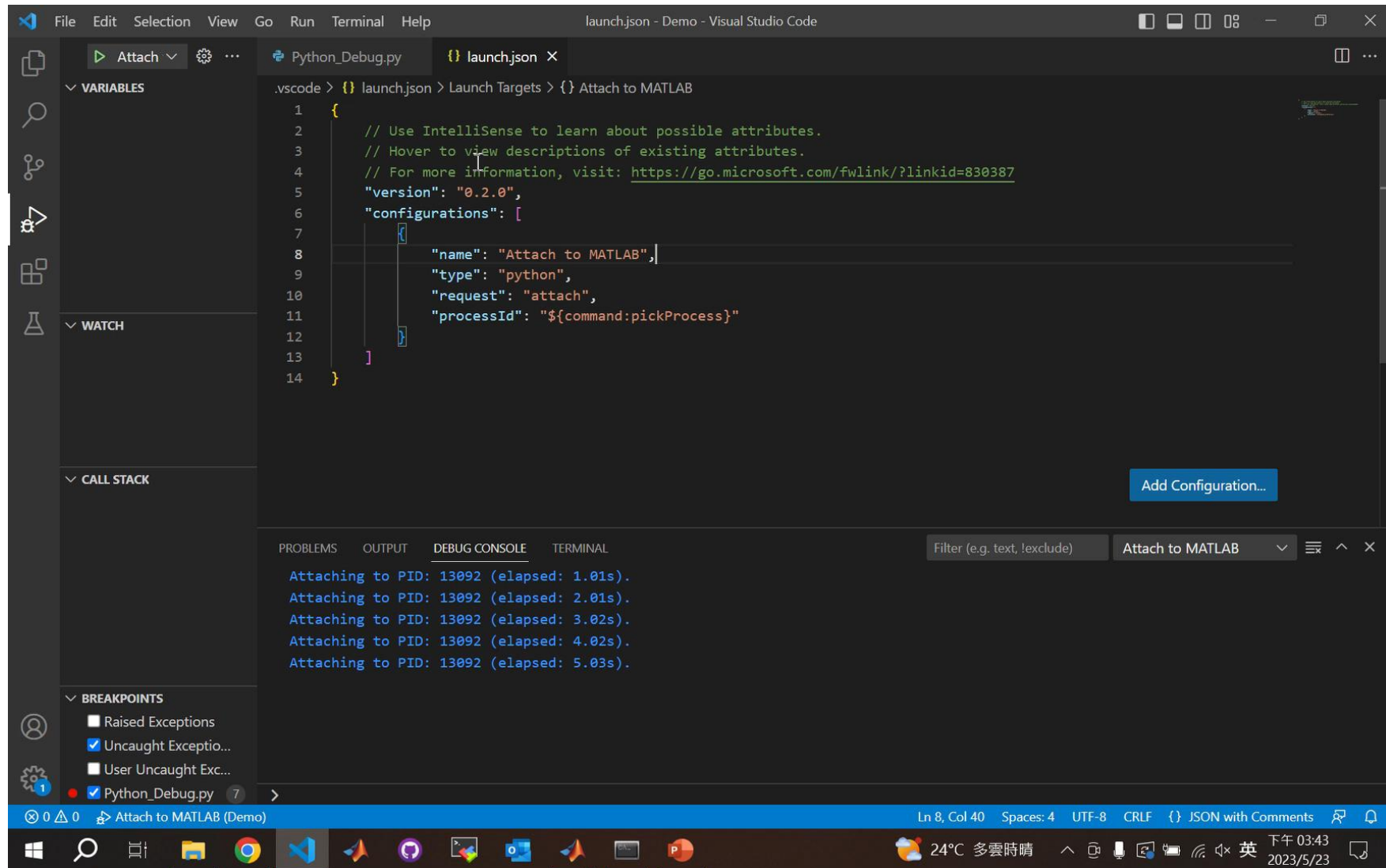
Preview:

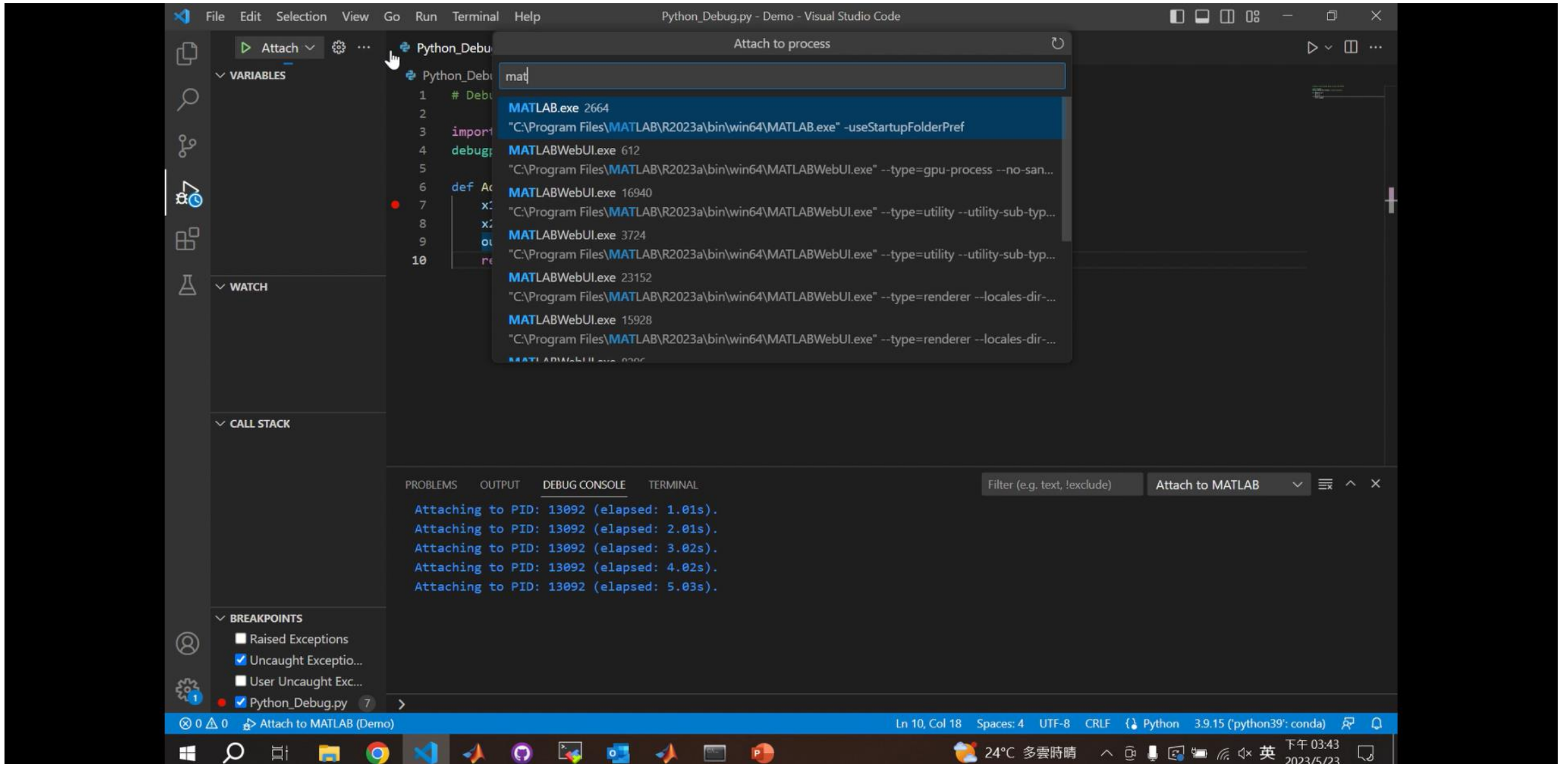
Zoom: 125% UTF-8 LF detectHuman\_detectHumanFromFrame Ln 1 Col 1

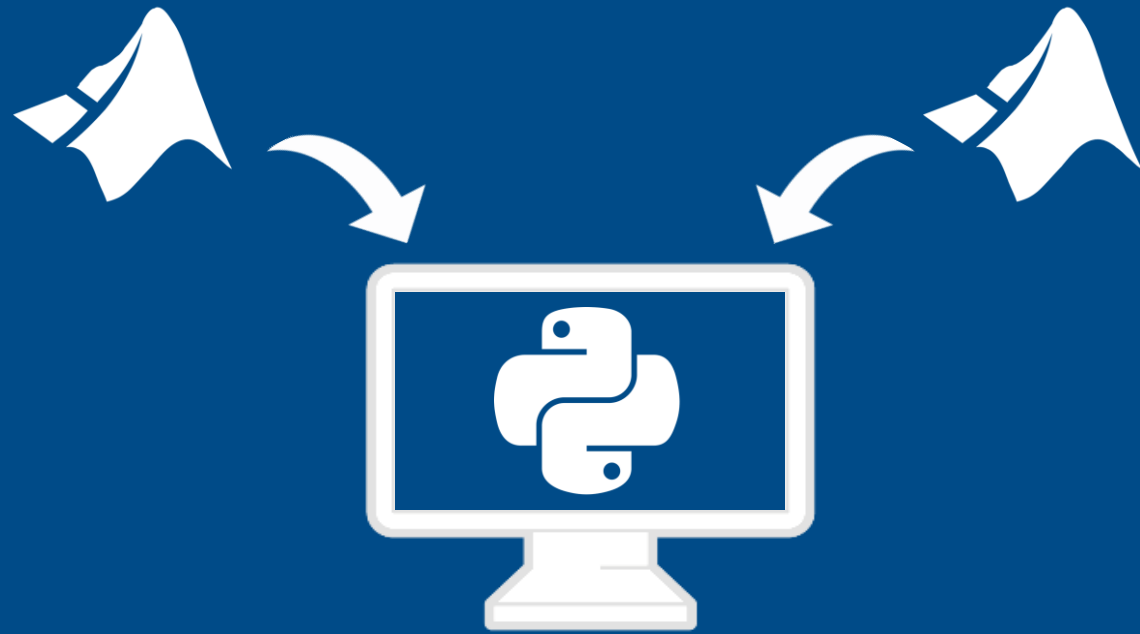
# Debug Python Code in MATLAB with VScode



# Debug Python Code in MATLAB with VScode





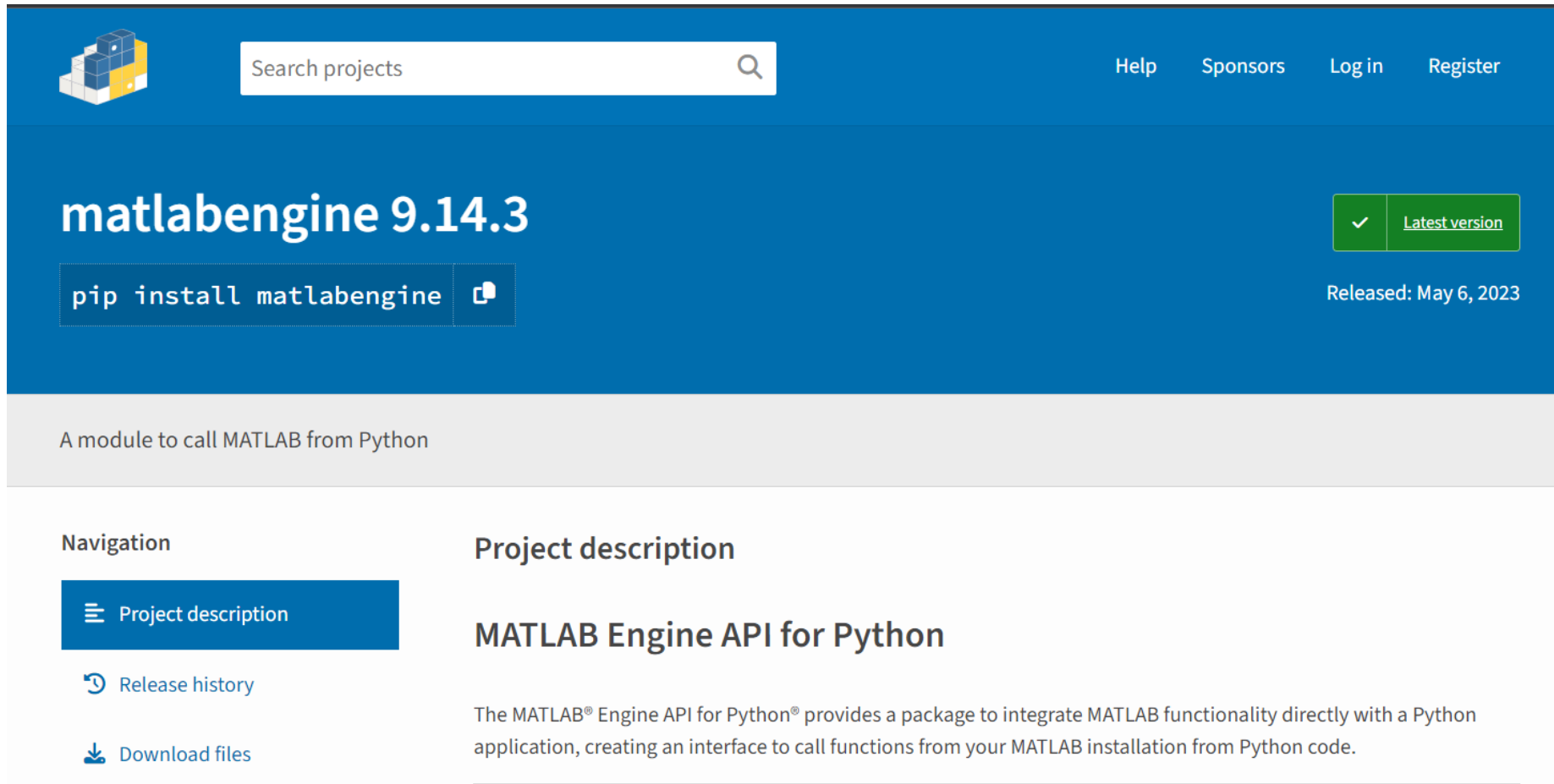


# Call MATLAB from Python



# Install MATLAB Engine API for Python – with PIP

## Set-up your Python environment



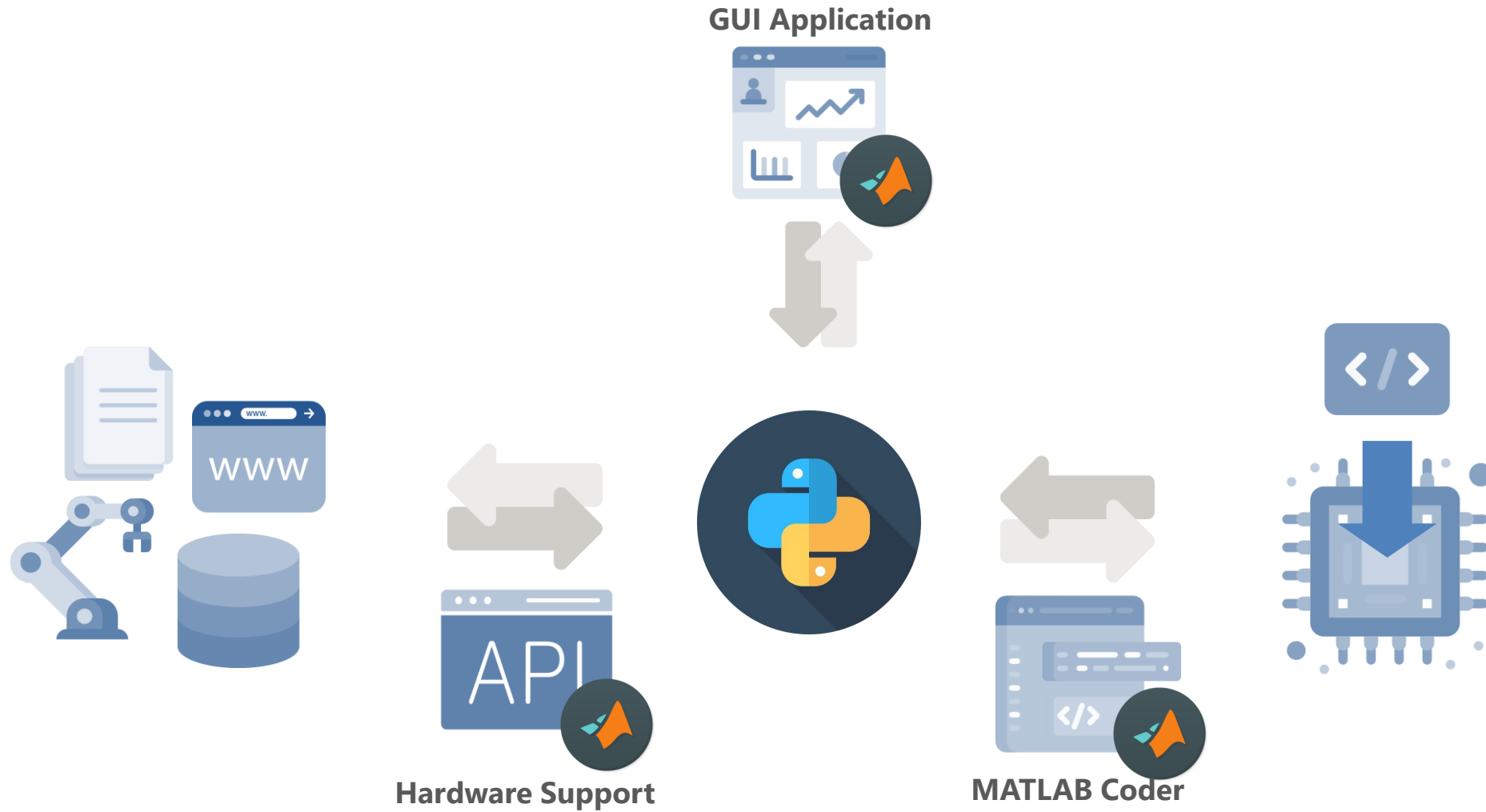
The screenshot shows the PyPI page for the 'matlabengine' package. At the top, there is a search bar and navigation links for 'Help', 'Sponsors', 'Log in', and 'Register'. The main header displays the package name 'matlabengine 9.14.3' in large white text on a blue background. To the right of the version number is a green checkmark icon and a 'Latest version' button. Below the version number is a code block containing the command 'pip install matlabengine' and a copy icon. To the right of the code block, it says 'Released: May 6, 2023'. Below the header is a grey bar with the text 'A module to call MATLAB from Python'. The page is divided into two columns. The left column is titled 'Navigation' and contains three items: 'Project description' (highlighted with a blue background), 'Release history', and 'Download files'. The right column is titled 'Project description' and contains the text 'MATLAB Engine API for Python' followed by a paragraph: 'The MATLAB® Engine API for Python® provides a package to integrate MATLAB functionality directly with a Python application, creating an interface to call functions from your MATLAB installation from Python code.'

# Call MATLAB from Python

**PIP Install** `matlabengine`

**Import** `matlabengine`

# 工具結合實例應用



# Demo: Python with MATLAB Image Labeler

在 Python 中使用 MATLAB Image Labeler

匯入所需函示庫

```
import matlab.engine as eng
import matplotlib.pyplot as plt
from matplotlib import style
import pandas as pd
import numpy as np
import torch
import torchvision
import torchvision.utils as utils
from PIL import Image
```

[1] ✓ 4.4s

建立與現有 MATLAB Session 連結

```
# 找到現有執行中的 MATLAB Session ID
# In MATLAB: matlab.engine.shareEngine
matlab_ID = eng.find_matlab()
print(matlab_ID)
```

The screenshot shows the MATLAB Image Labeler application. The main window displays a road scene with three cars highlighted by bounding boxes. The interface includes a menu bar with options like FILE, VIEW, LABEL, and EXPORT. On the left, there is a 'ROI Labels' panel with a 'Car' label selected. The main image area shows a first-person view from a car on a road with trees and a traffic light. Below the main image is a filmstrip of previous frames.

The screenshot shows the Visual Studio Code interface with a Python file named 'UsingAppInPython\_Image.ipynb'. The code is as follows:

```
import matlab.engine as eng
import matplotlib.pyplot as plt
from matplotlib import style
import sys
import pandas as pd
import numpy as np
import torch
import torchvision
import torchvision.utils as utils
from PIL import Image
```

Below the code, there is a section titled '匯入所需函示庫' (Import required libraries) and another section titled '建立與現有 MATLAB Session 連結' (Establish connection with existing MATLAB Session). The code in the second section is:

```
# 找到現有執行中的 MATLAB Session ID
# In MATLAB: matlab.engine.shareEngine
matlab_ID = eng.find_matlab()
print(matlab_ID)

# 選取欲建立連結的 MATLAB Session
sess = eng.connect_matlab(matlab_ID[0])
print(sess)
```

The interface also shows a file explorer on the left with files like 'BboxFromMATLAB.json', 'convertTorchBbox.m', and 'saveToJSON.m'. The bottom status bar indicates 'Ln 5, Col 1 CRLF' and the system tray shows the date and time as '下午 05:23 2023/5/23'.

# Demo: Python with MATLAB Signal Labeler

在 Python 中使用 MATLAB Signal Labeler

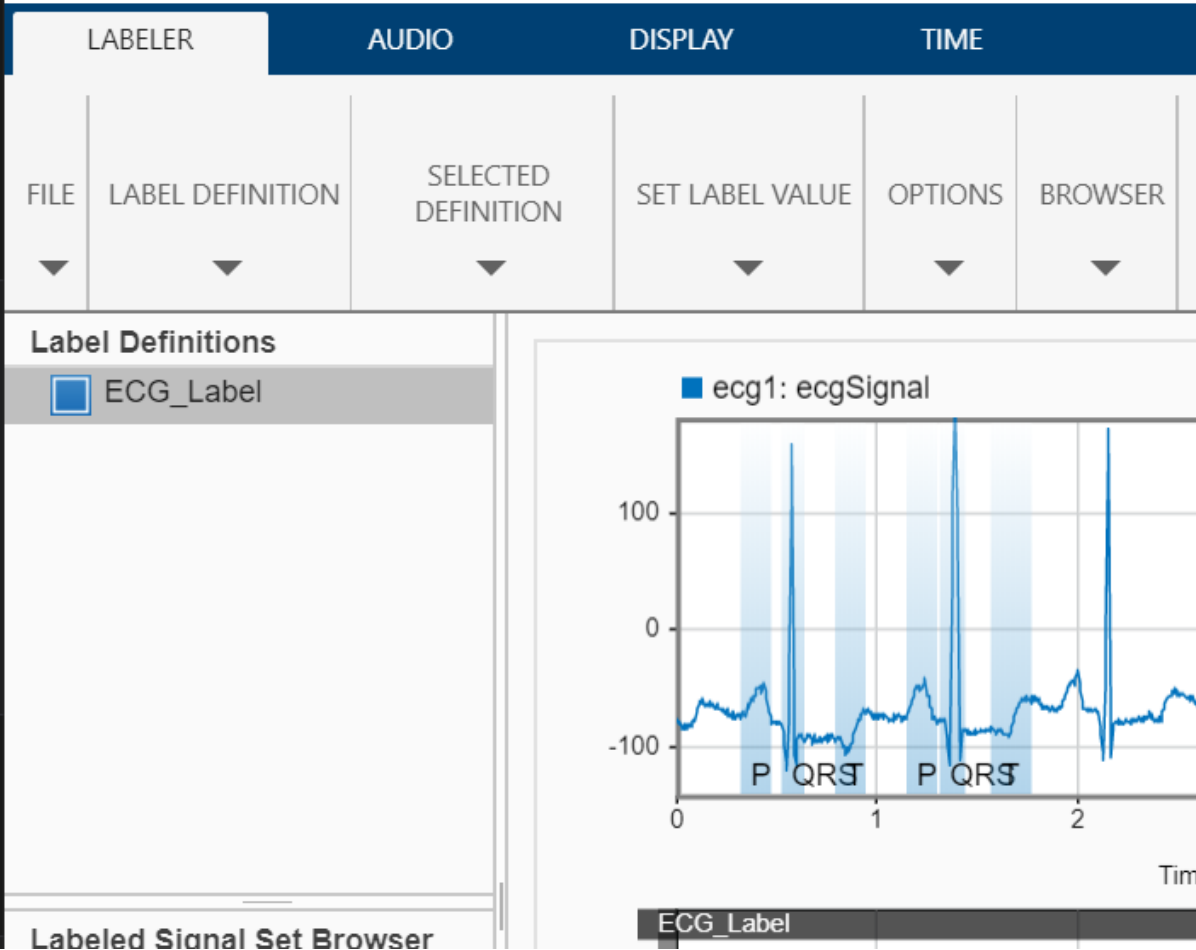
匯入所需函示庫

```
import matlab
import matlab.engine as eng
import matplotlib.pyplot as plt
from matplotlib import style
import pandas as pd
import numpy as np
import os
```

[1] ✓ 2.1s

建立與現有 MATLAB Session 連結

Signal Labeler





# Demo: Scikit-Learn + Experiment Manager

## Custom Training Experiment: Random Forest (Python)

Use this training function to define the training data, network architecture, training options, and training procedure used by the experiment. Experiment Manager saves the output of this function, so you can export it to the MATLAB workspace when the training is complete. For more information, see [Configure Custom Training Experiment](#).

### Input

- `params` is a structure with fields from the Experiment Manager hyperparameter table.
- `monitor` is an `experiments.Monitor` object that you can use to track the progress of the training, update information fields in the results table, record values of the metrics used by the training, and produce training plots.

### Output

- `output` is the output returned from the training function.

```
function output = Experiment1_training1(params,monitor)

% Initialize output
output.accuracy = [];
```

### Load and prep data

```
monitor.Status      = "Loading Training Data";
fname               = which("pumpData.parquet"); % need full path
df                  = py.pandas.read_parquet(fname);
y                   = pop(df, 'FaultType');
x                   = df;
s                   = uint8(1234);
traintestdata       = py.pumpy.split_data(x,y,0.2,s);
x_train             = traintestdata{1};
x_test              = traintestdata{2};
y_train             = traintestdata{3};
y_test              = traintestdata{4};
monitor.Progress    = 30;
```

### Define input parameters

Create a python dictionary to pass parameter values to the machine learning model (remember to convert types as needed).

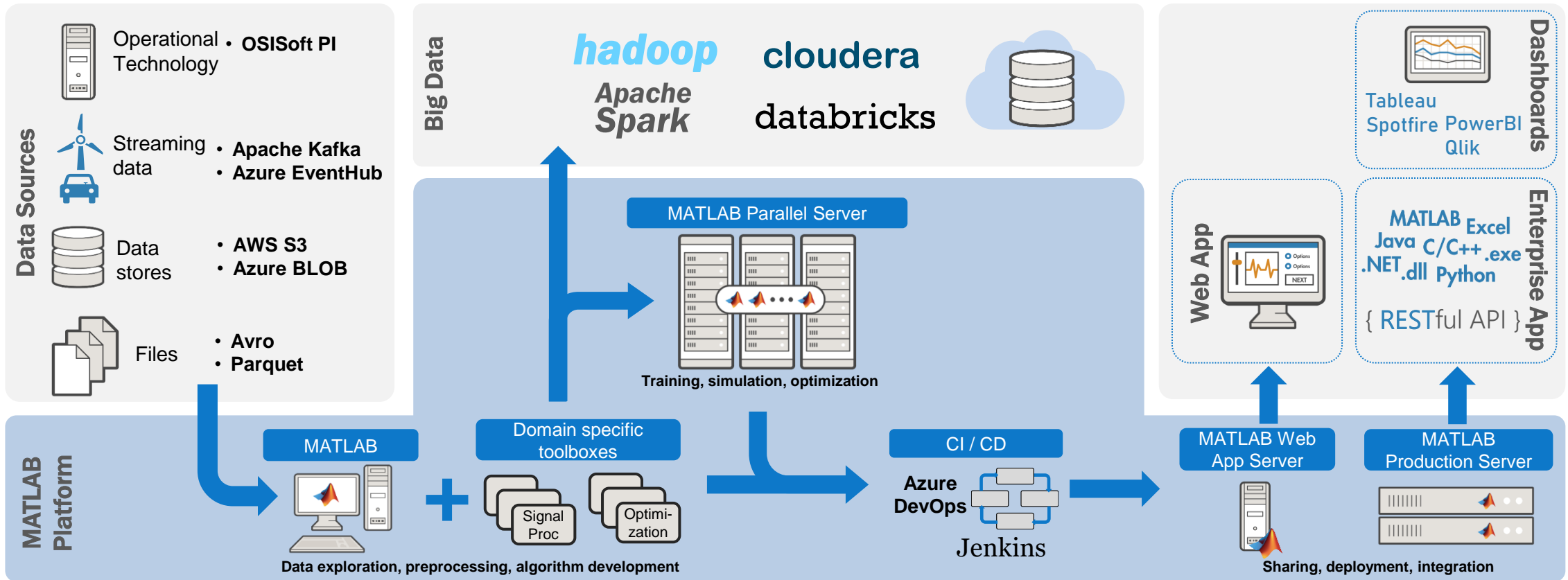
```
param_dist          = struct("n_estimators",uint8(params.n_estimators),...
                             "max_features",uint8(params.max_features),...)
```

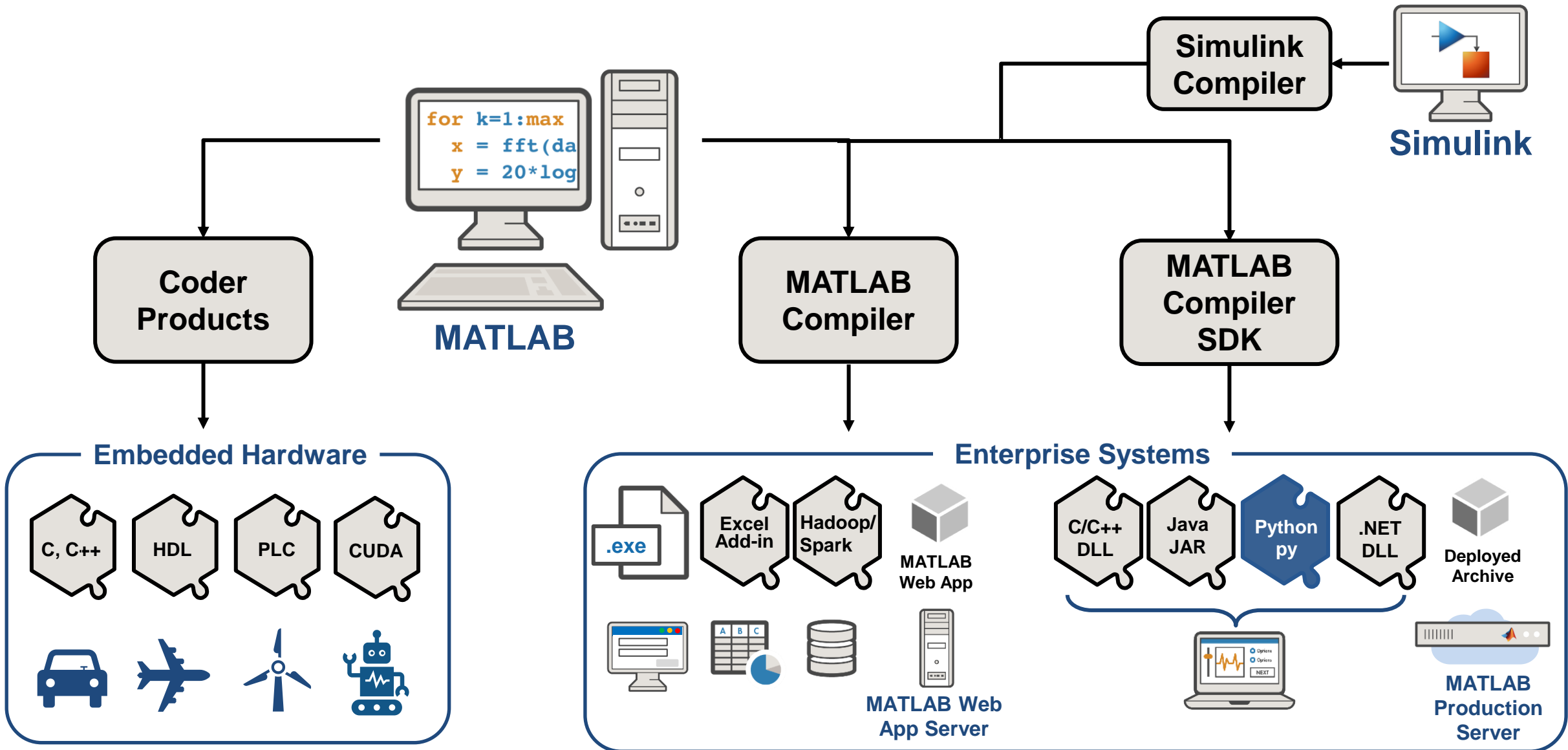




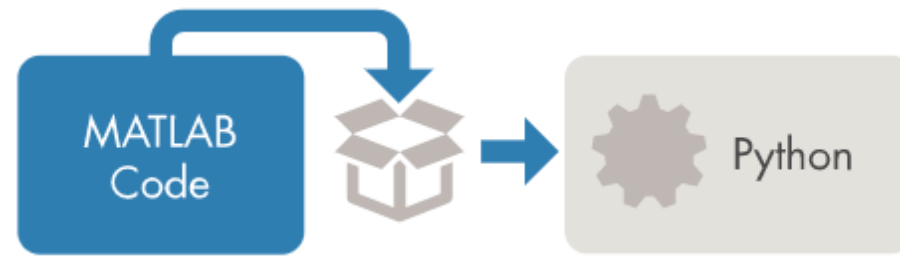
**Deployment**

# Deploy and Integrate Overview

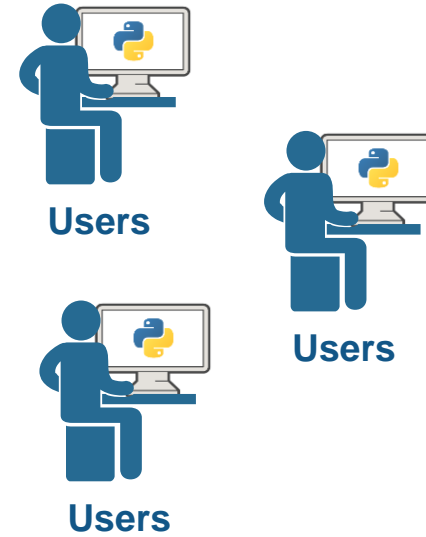




# Generate Python library from MATLAB functions

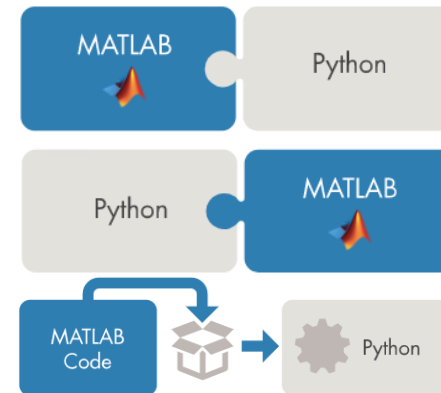


Packaging MATLAB programs for  
scalable deployment with Python



# Key Takeaways

- Three ways to integrate with Python
  - Calling Python from MATLAB
  - Calling MATLAB from Python
  - Generate Python library from MATLAB functions



- Automatically convert into MATLAB data type when passing variables between Python

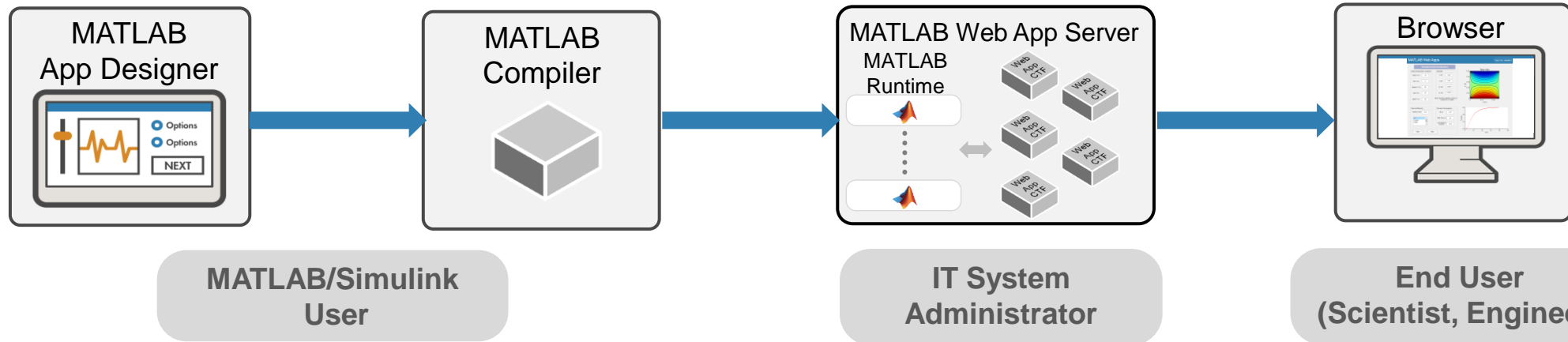
# 製作GUI並以網頁方式分享 – Web App Server

Develop the user interface and callback functions using MATLAB App Designer

Package MATLAB app, data and associated files as MATLAB Web Apps with one click

Deploy MATLAB Web Apps using the upload client directly from the browser

Access and run MATLAB Web Apps directly from a browser



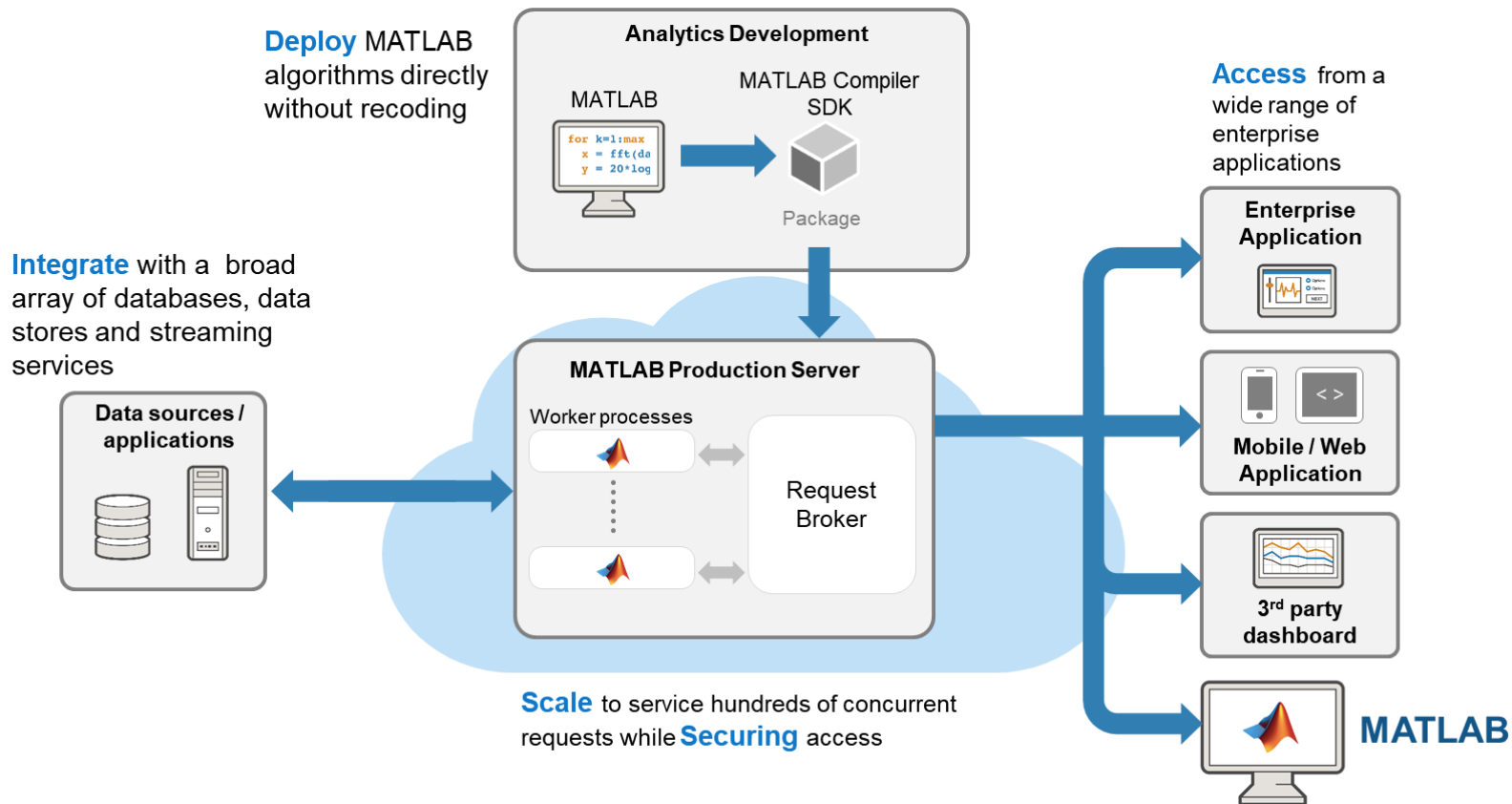
Create MATLAB Web Apps for sharing

Manage the server infrastructure

Easy access to MATLAB Web Apps

# 運行MATLAB核心算法並透過其它程式語言調用

## - MATLAB Production Server



### Deploy AI / MLOps

- Deploy trained AI models for inference
- Models can be stored in a REDIS in-memory database for high-speed access

### Process streaming analytics for IIoT

- Stream data from operational systems through streaming services such as Apache Kafka or Azure EventHub into MATLAB analytics for anomaly detection, condition based monitoring or predictive maintenance

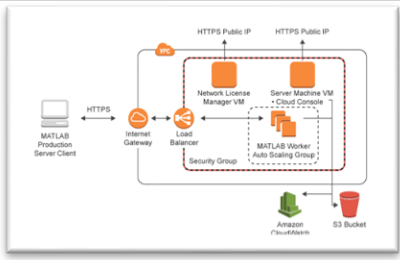
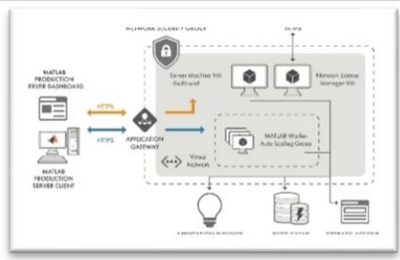
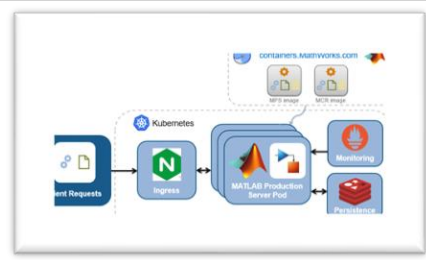
### Host Microservices APIs

- Share MATLAB algorithms and functions as microservice APIs in your corporate service fabric

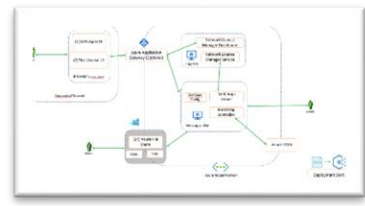
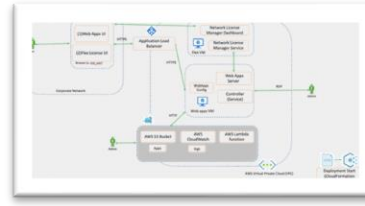
# 雲端架構擴展資源

- Available on:
  - Amazon Web Services
  - Microsoft Azure
  - Cloud Reference Architecture templates
- [Learn more](#)

### MATLAB Production Server

 <p><b>AWS</b></p> <p>Available as:</p> <ul style="list-style-type: none"> <li><a href="#">Cloud reference architecture on GitHub</a></li> <li><a href="#">AWS Marketplace (PAYG)</a></li> </ul>	 <p><b>Azure</b></p> <p>Available as:</p> <ul style="list-style-type: none"> <li><a href="#">Cloud reference architecture on GitHub</a></li> <li><a href="#">Azure Marketplace (BYOL)</a></li> <li><a href="#">Azure Marketplace (PAYG)</a></li> </ul>	 <p><b>Kubernetes</b></p> <p>Available as:</p> <ul style="list-style-type: none"> <li><a href="#">Cloud reference architecture on GitHub</a></li> <li>Works on all cloud providers</li> </ul>
---	---	--

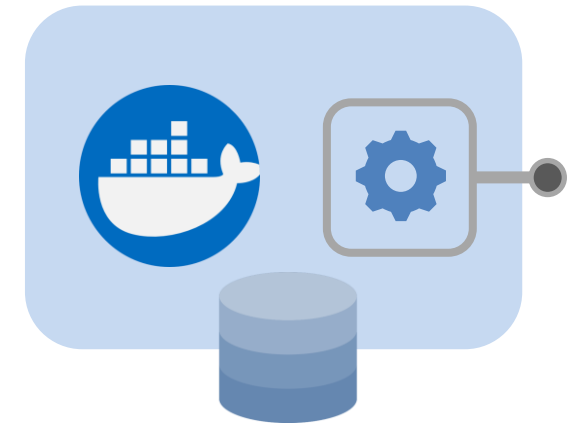
### MATLAB Web App Server

 <p><b>AWS</b></p> <p>Available as:</p> <ul style="list-style-type: none"> <li><a href="#">Cloud reference architecture on GitHub</a></li> </ul>	 <p><b>Azure</b></p> <p>Available as:</p> <ul style="list-style-type: none"> <li><a href="#">Cloud reference architecture on GitHub</a></li> </ul>
--	--

	Cloud Reference Architecture	Azure Marketplace (BYOL)	Azure Marketplace (PAYG)
Ability to customize, maximum flexibility	✓		
Move existing on-premises workloads to cloud		✓	
Long-term trials, proofs-of-concept			✓
Bursty workloads			✓



# Create Microservice Docker Containers from both Linux and Windows



HTTP RESTful Endpoint

```
>> compiler.build.productionServerArchive
```

Compile your code/model into a CTF (encrypted) archive

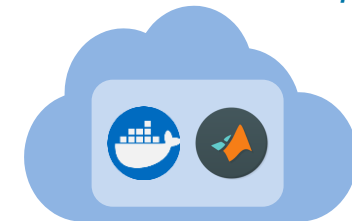
```
>> compiler.package.microserviceDockerImage
```

Creates a Docker container that has:

1. Ubuntu
2. MATLAB Runtime (optimized for size)
3. Your code in the CTF
4. A HTTP server

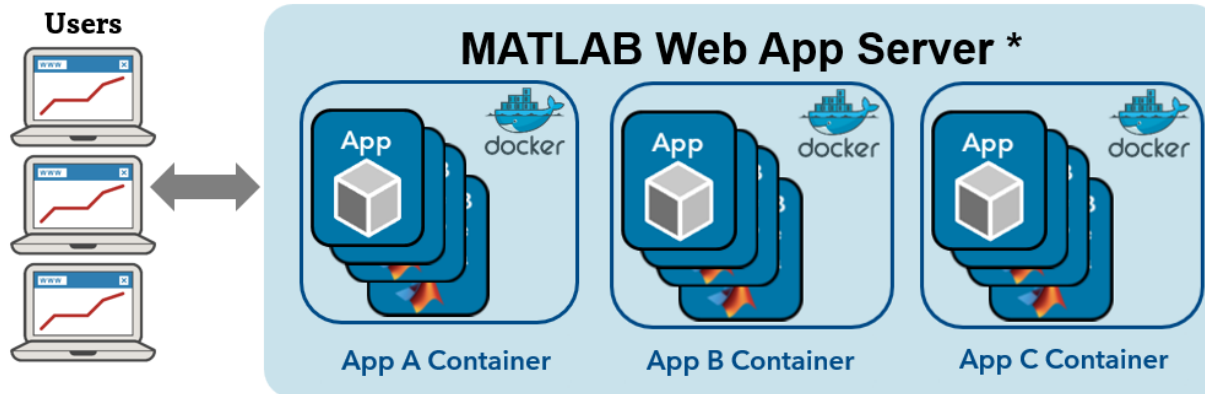


*Docker push to send image to cloud or corporate repository*

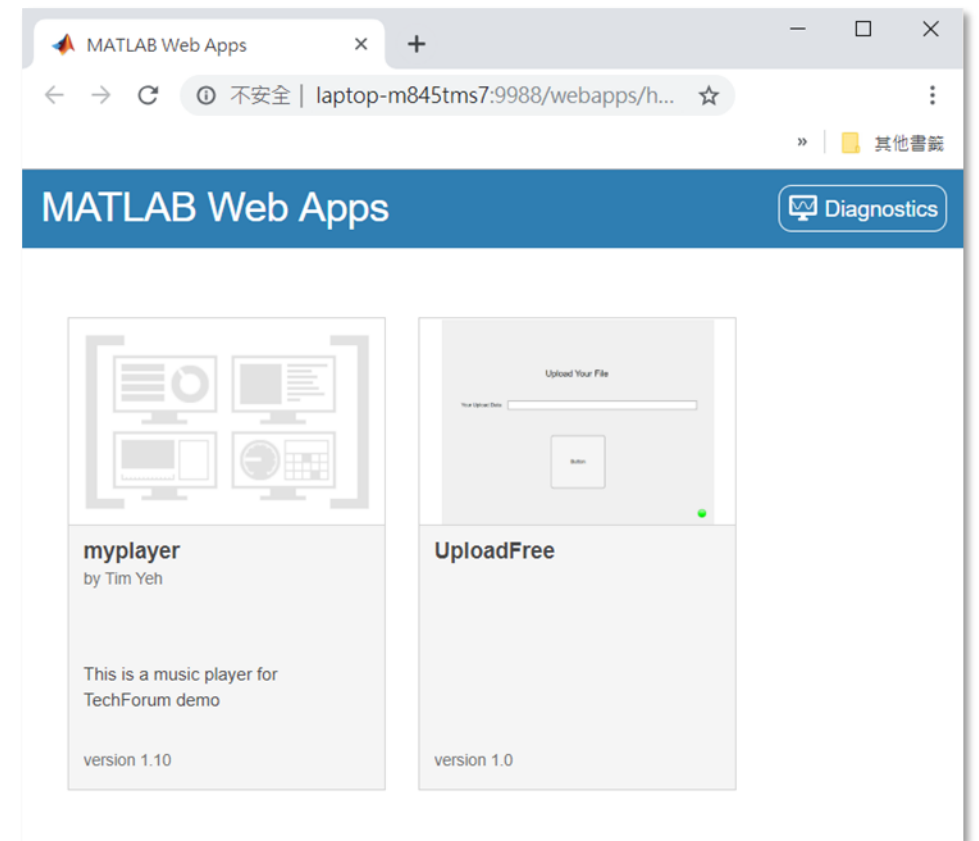


# Docker in MATLAB Web App Server

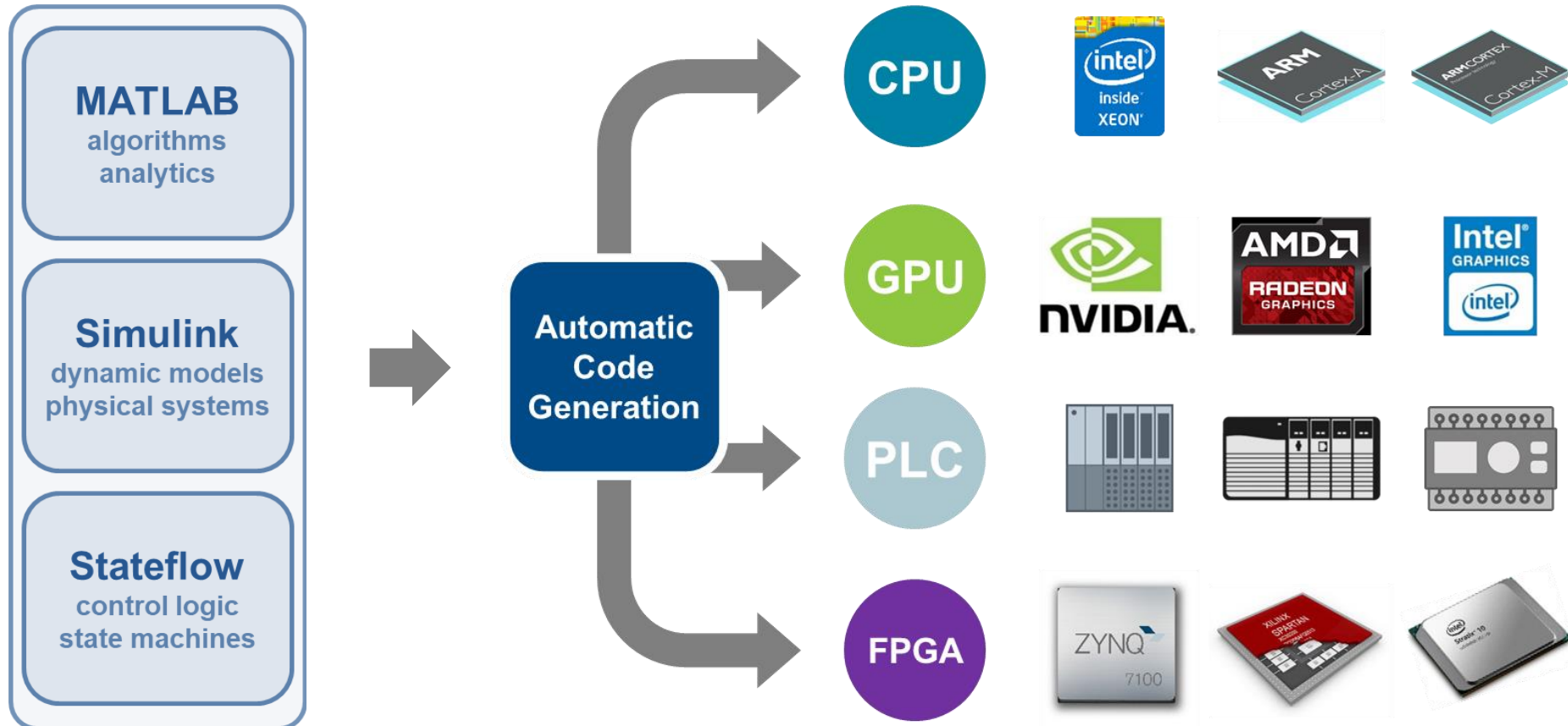
Create docker containers with app, logs and worker



**\*Please note that MATLAB Web App Server is non-dockerized**



# Automatic Code Generation From MATLAB & Simulink Simplifies This Process





# Academic Resources

## 各領域線上課程

程式基礎 / 人工智慧  
技巧課程，上完領  
證書



5G 儲存空間，  
資料存取無障礙

## MATLAB雲端硬碟

## 網頁版MATLAB

在網頁瀏覽器上使用



在手機/平  
板上用  
MATLAB，  
支援  
ios/Andoid

## 手機板MATLAB

# 你擁有的不只是一個軟體!

校園授權享有的資源，比你想像多更多!

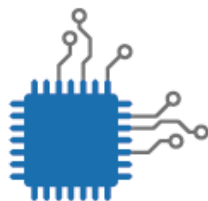


## 叢集&分散式運算

提升平行運算速度

## 支援低成本硬體

Arduino / Raspberry  
Pi / Beagl /  
LEGO



作業自動  
評分系統

## MATLAB 線上作業評分系統

## 軟體下載安裝

不限下載電腦台數 /  
維護期內更新



# 各領域線上課程

自選上課時間，互動式練習實作，完成頒發證書



實際操作範例，系統在過程中自動提供協助



針對成果及時給予回饋



完成頒發證書，證書可即時分享。

# 線上課程

MATLAB and Simulink Training

Search MathWorks.com

Training Overview Find a Course Get Certified Training At Your Facility More » My Courses Contact Training

### Core MATLAB

- Launch** **MATLAB Fundamentals**  
Learn core MATLAB functionality for data analysis, modeling, and programming.  
Access Expires : 2020/8/31  
[View/Share Certificate](#)  
[Settings and Quick Reference](#)
- Launch** **MATLAB for Data P**  
Create custom visualizations  
Access Expires : 2020/8/31
- Launch** **MATLAB Programm**  
Improve the robustness, flexi  
Access Expires : 2020/8/31
- Launch** **MATLAB for Financ**  
Learn MATLAB for financial d  
Access Expires : 2020/8/31

MATLAB and Simulink Training

Search MathWorks.com

Training Overview Find a Course Get Certified Training At Your Facility More » My Courses Contact Training

### Getting Started

- Launch** **MATLAB Onramp**  
Get started quickly with the basics of MA  
Unlimited access
- Details** **Simulink Onramp**  
Get started quickly with the basics of Sim  
Unlimited access
- Launch** **Machine Learning Onramp**  
Learn the basics of practical machine lea  
Unlimited access
- Launch** **Deep Learning Onramp**  
Get started quickly using deep learning r  
Unlimited access
- Details** **Stateflow Onramp**  
Learn the basics of creating, editing, and simulating state machines in Stateflow.  
Unlimited access

Core MATLAB

MATLAB and Simulink Training

Search MathWorks.com

Training Overview Find a Course Get Certified Training At Your Facility More » My Courses Contact Training

### Data Science

- Resume** **Machine Learning with MATLAB**  
Explore data and build predictive models.  
1% Access Expires : 2020/8/31  
[View/Share Certificate](#)  
[Settings and Quick Reference](#)
- Launch** **Deep Learning with MATLAB**  
Learn the theory and practice of building deep neural networks with real-life image and sequence data.  
Access Expires : 2020/8/31  
[View/Share Certificate](#)  
[Settings and Quick Reference](#)

### Computational Mathematics

- Resume** **Introduction to Symbolic Math with MATLAB**  
Get started quickly with an introduction to symbolic math.  
3% Access Expires : 2020/8/31  
[View/Share Certificate](#)  
[Settings](#)
- Launch** **Solving Nonlinear Equations with MATLAB**  
Use root finding methods to solve nonlinear equations.  
Access Expires : 2020/8/31  
[View/Share Certificate](#)  
[Settings and Quick Reference](#)
- Launch** **Solving Ordinary Differential Equations with MATLAB**  
Use MATLAB ODE solvers to numerically solve ordinary differential equations.  
[View/Share Certificate](#)  
[Settings and Quick Reference](#)

# MATLAB Online Course Overview

- 隨時隨地學習：透過網絡瀏覽器或軟體中存取
- 邊做邊學：在 MATLAB 和 Simulink 中完成任務。
- 立即收到反饋跟踪進度並分享完成證書

The screenshot displays the MATLAB Online course interface. At the top, it shows 'MY COURSES' and 'MATLAB Fundamentals (2% complete)'. The current task is '4.3 Creating Evenly-Spaced Vectors: (4/8) Use Colon Operator and Linspace'. The interface is divided into three task sections: Task 1, Task 2, and Task 3. Task 1 shows the code `x = 5:15`. Task 2 shows the code `x = linspace(5,15,13)`. Task 3 shows the code `x = 3:2:11`. The 'Test Results' section indicates 'Correct!' and lists two checks: 'Is x defined correctly?' and 'Does script not contain square brackets?'. The 'Submit' and 'Next task' buttons are visible.

← MY COURSES **MATLAB Fundamentals** (2% complete)

4.3 Creating Evenly-Spaced Vectors: (4/8) Use Colon Operator and Linspace

Task 1

Task 2

Task 3

**TASK**  
Create a variable named `x` that contains the row vector shown below.

3 5 7 9 11

Hint | See Solution | Reset  
Course Quick Reference

Submit Next task

**Test Results: Correct!**

- ✓ Is `x` defined correctly?
- ✓ Does script not contain square brackets?

HOME LIVE EDITOR VIEW

Text Code

EvenlySpacedVectors.mlx \* x +

at a time.

Task 1

1 `x = 5:15`

Task 2

2 `x = linspace(5,15,13)`

Task 3

3 `x = 3:2:11`



# MATLAB 線上作業評分系統(MATLAB Grader)

**Solution 1: 4 of 5 tests passed (80%)** Test Results  
 Submitted on 20 Nov 2018 | ID: 10161219 | Size: 63 ✘ ✔ ✔ ✔ ✔

```

1 syms g x; % insert variable
2 g=piecewise(x<=0, x^2, 0<x<1, x, x>=1, 1+x^2); % insert conditions and values
3
4 % evaluate the function at different points a=g(-2); b=g(0.5); c=g(2)
5 a=subs(g, x, -2)
6 b=subs(g, x, 0.5)
7 c=subs(g, x, 2)
  
```

View:  All Solutions  Test Solutions  Submitted

Size

Order of

Back to Problem

**Learner Status** Learner Solutions

Status Summary  
564 Learners have accessed the problem.

Solved: 43% (240)  
 Not solved: 7% (41)  
 No solutions submitted: 50% (283)

Solved:  
240 Learners have solved the problem.

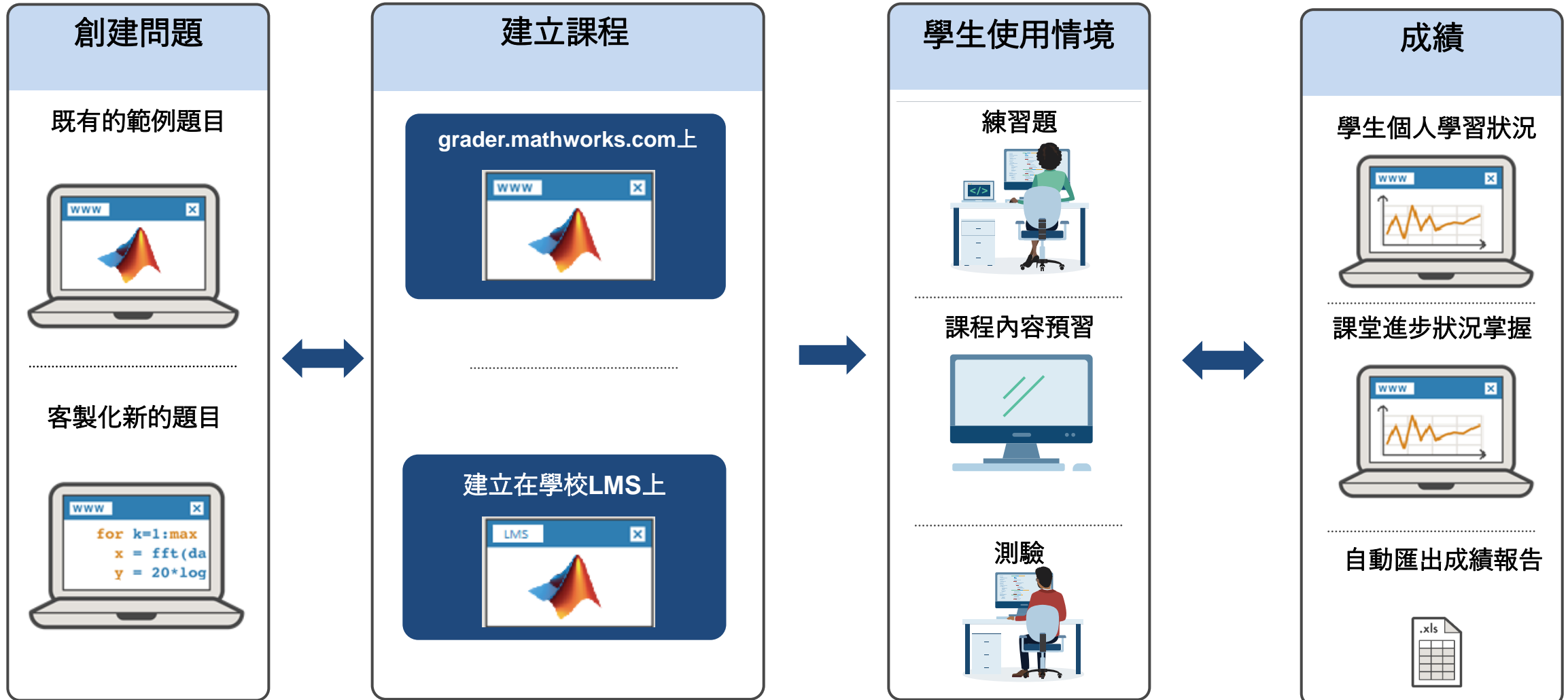
Submissions Required to Solve the Problem  
Mean: 3

Ave. Submissions Required to Pass Each Test

Fix This Solution My Solutions

0%	(20%)
20%	(20%)
20%	(20%)
20%	(20%)
20%	(20%)
<b>Total: 80%</b>	

# MATLAB Grader 使用流程



# 將傳統作業轉換為自動評分系統

Home Tools Assignment\_v2.pdf x ? Sign In

1 / 3 44.6%

## INTRODUCTION TO PROGRAMMING WITH MATLAB

### Assignment 1: Convergent Series

**Background**

In mathematics, a series is the sum of the terms of an infinite sequence of numbers. A series is convergent if the sequence of its partial sums tends to a limit; that means that the partial sums become closer and closer to a given number when the number of their terms increases.

For more details, please refer to the [Wikipedia entry on Convergent Series](#).

**Problem 1b: Estimating the value of Pi using Leibniz Series - Due 9/1**

One of the methods to estimate the value of  $\pi$  is to use the Leibniz series expansion to a reasonably large number of terms and use the expression below to estimate the value of  $\pi$ .

$$\frac{\pi}{4} \approx 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots + \frac{(-1)^{n+1}}{2n-1} = \sum_{k=1}^n \frac{(-1)^{k+1}}{2k-1}$$

LaTeX: `\pi/4 \approx 1 - 1/3 + 1/5 - 1/7 + \dots = \sum_{k=1}^n (-1)^{k+1} / (2k-1)`

Using this expression, write a script to estimate the value of  $\pi$  using  $N$  terms. Your code should include the following variables:

```
N % Number of terms used in the series expansion
estPi % Value of pi estimated using 'N' terms in the series.
```

Determine a value of  $N$  that ensures that the estimated value of  $\pi$  is within 0.1% of the actual value. Start with 10 terms, and increase or decrease the number appropriately to adjust the estimate.

You can use the Learner Template code provided below to develop your solution.

**Learner Template**

```
nTerms = ; % Number of terms to be used in the series expansion
% <Enter your code here>

estPi = ; % Estimated value of Pi for 'N' values.
```

**Check to ensure that:**

- the code does NOT use the variable 'pi' available in MATLAB.
- the output is numerically accurate for the number of series terms used.

**Test Suite 1:** Is MATLAB's built-in variable 'pi' being invoked in your code?

**Feedback:** The variable 'pi' available in MATLAB is being used in your code. Please retain only your estimated value of  $\pi$  under the variable name 'estPi'.

**Test Suite 2:** Is the estimated value of 'pi' acceptably accurate?

**Feedback:** Your estimated value doesn't fall within 0.1% of the expected value of  $\pi$ .



MATLAB Grader Jeff Alderson

CONTENTS Close Courses & Content LMS Integration License Management Documentation & Support

CSU Demo CSU Demo > Week 1 Homework >

## Untitled Problem

< Back to Add Problem | Introduction to Programming Close x

**Vector Creation (Leibniz series terms)** Copy

hide details...

Consider the Leibniz series:

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$$

Write a script to generate a vector of the first twenty terms of this series. Assign the vector of series terms to a row vector variable named **LeibnizTerms**.

Solve this problem using vectorized code (i.e. do not use a loop in your solution.)

**Files Referenced**  
None

**Problem Type**  
Script

**Code**

Reference Solution Learner Template

```
1 k = 0:19;
2 LeibnizTerms = (-1).^k ./ (2 * k + 1);
```

**Assessment**

> **Test 1**

Does variable LeibnizTerms have the correct values?

LeibnizTerms = Reference Solution?

# MATLAB Grader使用範例

## 問題

20200708 Webinar MATLAB Grader サンプルコース > 課題 1 >

### 3元連立方程式を解く

Edit Actions

以下の3元連立方程式を解け。(MATLABにはバックスラッシュ演算子がありますが、それはここでは使ってはいけません)

$$\begin{cases} 5x - 4y + 6z = 8 \\ 7x - 6y + 10z = 14 \\ 4x + 9y + 7z = 74 \end{cases}$$

### Script

Reset MATLAB Documentation

```
1 A = [5 -4 6;7 -6 10;4 9 7];
2 b = [8;14;74];
3
4 X = A\b;
5 x = X(1)
6 y = X(2)
7 z = X(3)
```

## 學生回答

Run Script

### Assessment:

Run Pretest

Submit

Pretest results are not submitted for grading. To run all the tests and submit the results for grading, click Submit.

>  xは正しく計算されているか (Pretest)

>  yは正しく計算されているか (Pretest)

>  zは正しく計算されているか (Pretest)

>  バックスラッシュは使われているか (Pretest)

The submission must not contain the backslash operator: \

## 即時回饋

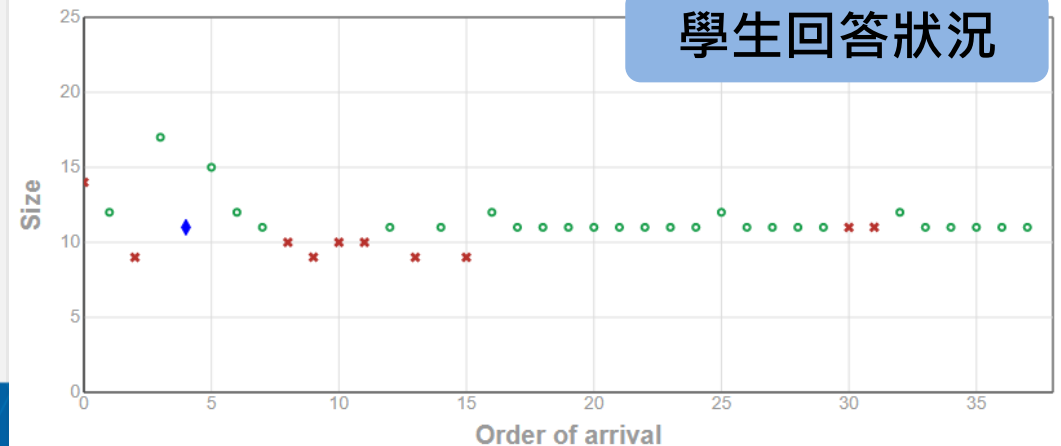
### Learner Analytics

Class Overview Learner Solutions

Map View

List View

Search by last name, code or solutio



## 學生回答狀況

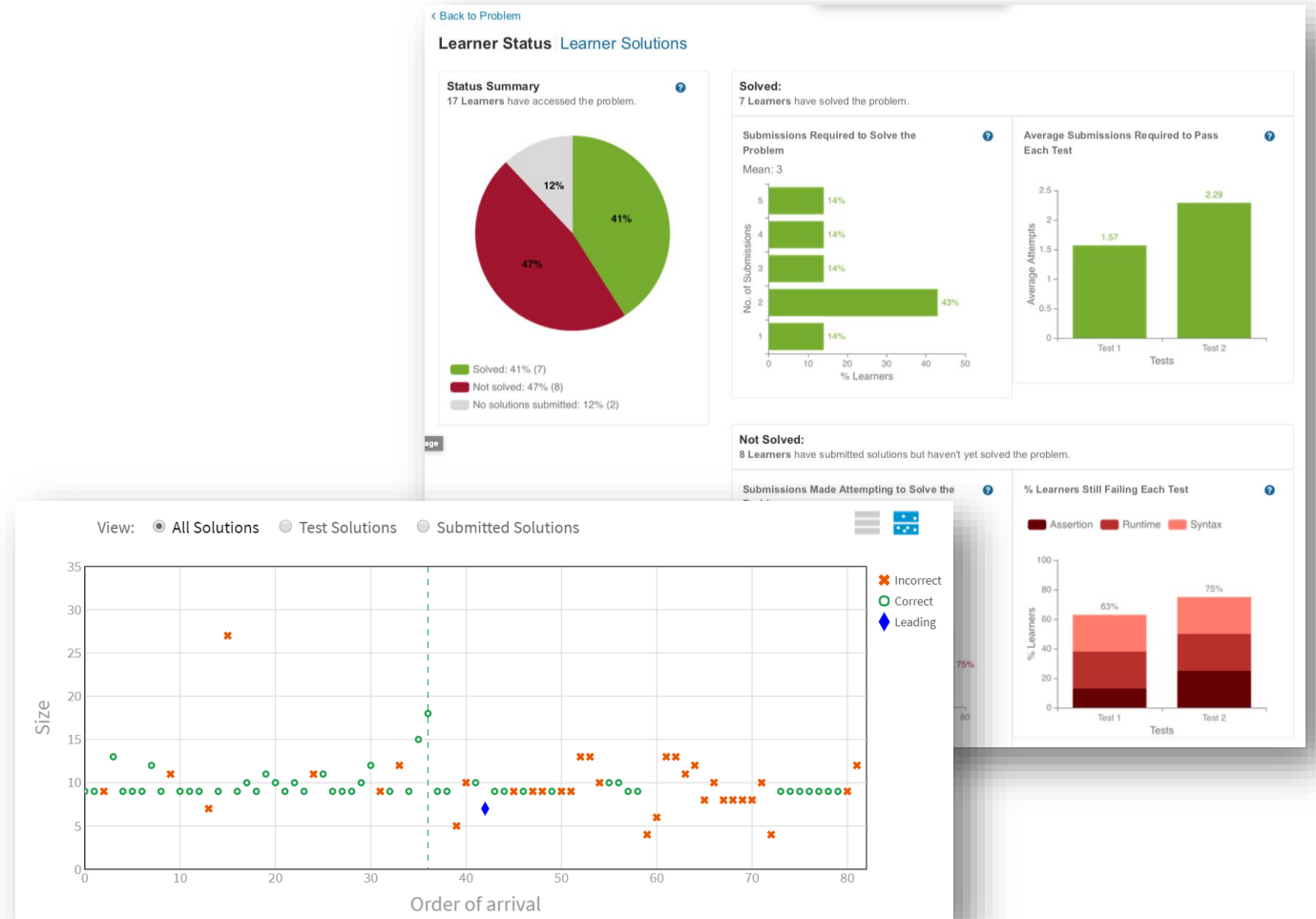
# MATLAB Graderの機能 - 成績管理

- 自動對學生的答案進行評分

- 答對比率分析
- 解答的即時反饋

- 成績分析

- 個別學生答題狀況分析
- 整體學生答題狀況分析
- 教師/助教可以查看分數



# LMS(學習管理系統)結合

- 集成和管理學習材料、作業、成績等。
- 通過連接 MATLAB Grader 和 LMS 彙整成績。



# MATLAB & Simulink技術應用文章/論文、海報徵選

登上MATLAB & Simulink發表舞台

## 挑戰自我，掌握未來

比賽優勝者不但能獲得高額獎金、研究更能被台灣產界、學界的菁英看見

徵選主題

	人工智慧 / 深度學習 / 機器學習 / 預測性維護		訊號處理 / 通訊系統設計
	影像處理 / 機器視覺		機器人開發 / 自動控制
	自動駕駛開發 / 車輛控制		電力系統 / 電力電子 / 新能源
	生醫 / 醫學影像 / 醫學訊號處理		

# MATLAB & Simulink技術應用文章/論文、海報徵選

技術應用文章/論文徵選

## 競賽時程



## 獎項與獎金

第一名	★ 獎金30,000元
第二名	★ 獎金15,000元
第三名	★ 獎金10,000元
優選獎	★ 獎金3,500元
參加獎	★ 參加證明&MATLAB限量周邊



# MATLAB & Simulink技術應用文章/論文、海報徵選

MATLAB創意海報人氣競賽

報名與海報提交	2023年10月1日 23:59前
開放投票	2023年10月17日-2023年11月19日 23:59止
結果公布	2023年11月22日 公布於活動網站

## 獎項

- 最高人氣獎：獎金2000元及MATLAB限量周邊
- 最佳創意獎：獎金2000元及MATLAB限量周邊



Q&A