MATLAB With Python for engineers and data scientists

성 호 현 The MathWorks Korea
Example: Build Air Quality App using MATLAB and Python
Development plan

Data Access

Co-execution

Deployment

Call Python from MATLAB

Call MATLAB from Python
Data Access
Access Data from a Web Service

https://openweathermap.org/

**OpenWeather global services**

Weather forecasts, nowcasts and history in fast and elegant way

2 Billion Forecasts Per Day
2,500 new subscribers a day

2,600,000 customers
20+ weather APIs

**Hourly forecast**

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Precipitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>27°C</td>
<td></td>
</tr>
<tr>
<td>27°C</td>
<td></td>
</tr>
<tr>
<td>28°C</td>
<td></td>
</tr>
<tr>
<td>24°C</td>
<td></td>
</tr>
<tr>
<td>23°C</td>
<td></td>
</tr>
<tr>
<td>22°C</td>
<td></td>
</tr>
<tr>
<td>21°C</td>
<td></td>
</tr>
<tr>
<td>20°C</td>
<td></td>
</tr>
<tr>
<td>19°C</td>
<td></td>
</tr>
</tbody>
</table>

**8-day forecast**

- **Wed, Sep 09**: 27 / 19°C, clear sky
- **Thu, Sep 10**: 25 / 18°C, broken clouds
- **Fri, Sep 11**: 28 / 17°C, scattered clouds
- **Sat, Sep 12**: 25 / 16°C, clear sky
- **Sun, Sep 13**: 28 / 15°C, broken clouds
- **Mon, Sep 14**: 32 / 17°C, scattered clouds
- **Tue, Sep 15**: 27 / 20°C, moderate rain
- **Wed, Sep 16**: 21 / 17°C, light rain

Data Access

Co-Execution
- Call Python from MATLAB
- Call MATLAB from Python

Deployment
What type of data?

Numerical, Textual, Geolocalized, Timeseries, …

Called by:
- geographical coordinates, zip codes, city name, city ID, number of cities (only in current and forecasted APIs)

https://openweathermap.org/
Store & transfer tabular data between languages

Parquet files

- Columnar storage format available to any project in the Hadoop big data ecosystem, regardless of the choice of data processing framework, data model or programming language (More on Parquet)
Co-execution
MATLAB provides flexible integration with multiple languages.
Given: Existing Python Code accessing & preparing weather data

- Data Access
- Co-Execution
  - Call Python from MATLAB
  - Call MATLAB from Python
- Deployment
Call Python from MATLAB

Data Access

Co-Execution
- Call Python from MATLAB
- Call MATLAB from Python

Deployment

Data preparation  Modeling  Deployment

Weather Data

Call Python from MATLAB

Data preparation  Modeling  Deployment

Weather Data

Call Python from MATLAB

Data preparation  Modeling  Deployment

Weather Data
Call MATLAB from Python

Data Access

Co-Execution
- Call Python from MATLAB
- Call MATLAB from Python

Deployment

Weather Data

Data preparation
Modeling
Deployment
Deploy: MATLAB Analytics into Python

Data Access

Co-Execution
- Call Python from MATLAB
- Call MATLAB from Python

Deployment

Data preparation
Modeling
Deployment

Weather Data

Data preparation

Modeling

Deployment

Determine air quality conditions in your area.
Zip code: 02118
Deploy:
MATLAB Analytics into Python

Data Access

Co-Execution
- Call Python from MATLAB
- Call MATLAB from Python

Deployment

Data preparation → Modeling → Deployment

Weather Data

Data interface

My Weather Page
www.myweather.com/stats.html

Weather Data

Determine air quality conditions in your area.
Zip code: 02110

Sine fitting of weather forecast in Paris from 15-Jul-2020

14
Call Python from MATLAB
Why Call Python from MATLAB?

Already working in MATLAB, and:
- Want to reuse existing Python code
- Need functionality that is only available in Python
Use the weather.py module to get the air quality for Paris. This is a user-defined Python module which includes functions to read and parse the current and forecasted weather data by location.

```python
jsonData = py.weather.get_current_weather("Paris", "France", apikey.Key)

jsonData -
Python dict with no properties.

('coord': {'lon': 2.35, 'lat': 48.85}, 'weather': [{'id': 803, 'main': 'Clouds',

Parse the json data returned from the weather API.

The Python dictionary can be represented as a MATLAB struct.

```matlab
weatherData = py.weather.parse_json(jsondata);
struct(weatherData)

ans = struct with fields:
  temp: 18.7100
  feels_like: 17.3000
  temp_min: 17.7800
  temp_max: [1x1 py.int]
```

Use a function (`prepData.m`) to prepare data for machine learning (create a table with the expected variable names, preprocessing steps, etc).

```matlab
currentData = prepData(weatherData)

currentData - 1x12 table

<table>
<thead>
<tr>
<th>DateLocal</th>
<th>city</th>
<th>StateName</th>
<th>T</th>
<th>P</th>
<th>DP</th>
<th>RH</th>
<th>WindDir</th>
<th>WindSpd</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-Jul-2020 11:00</td>
<td>&quot;Paris&quot;</td>
<td>Ile de France</td>
<td>21.6200</td>
<td>20.2600</td>
<td>349.2200</td>
<td>1010</td>
<td>5.1000</td>
<td>73</td>
</tr>
</tbody>
</table>
```
Calling Python libraries from MATLAB

Data Access

Co-Execution
- Call Python from MATLAB
- Call MATLAB from Python

Deployment

Calling Python libraries from MATLAB:

```python
# weather.py
import csv
import datetime
import json
import urllib.request

BASE_URL = 'https://api.openweathermap.org/data/2.5/{}?q={},{}&unit=metric'
FORECAST_KEYS = {'current_time': 'DateTime', 'temp': 'T', 'deg': 'WindDir', 'speed': 'WindSpd', 'humidity': 'RH', 'pressure': 'P'}

def read_backup(city):
    # Read example data from a backup file
    with open('backupdata.csv', newline='') as csvfile:
        reader = csv.DictReader(csvfile)
        for s in [*reader]:
            pass
```
Connect to Python

- Call Python from MATLAB
- Call MATLAB from Python
Data Access

Co-Execution
- Call Python from MATLAB
- Call MATLAB from Python

Deployment
Air Quality Prediction Model

Data Access

• Call Python from MATLAB
• Call MATLAB from Python

Co-Execution

Deployment

Mastering Machine Learning
A Step-by-Step Guide with MATLAB
Train the Air Quality Prediction Model

Model development is illustrated in this webinar MATLAB with Data Science
Call the Air Quality Prediction Model

Use the model to predict the air quality for the new weather data.

'airQualModel.mat' is a pre-trained Bagged Classification Tree/ "Random Forest" classification network. The model was saved as a *.mat file for use in predicting air quality in this demonstration.

```matlab
load airQualModel
```

Use a function (prepData.m) to convert and prepare data for machine learning (create a table with the expected variable names, preprocessing steps, etc).

```matlab
apikey = readtable("accessKey.txt", "TextType", "string");
List = "Houston";
jsonData = py.weather.get_current_weather(List, "US", apikey.Key);
weatherData = py.weather.parse_current_json(jsonData);
currentData = prepData(weatherData);
airQual = predict(model, currentData)
```
Recap: Calling Python from MATLAB
Syntax differences when calling Python from MATLAB

**Python**

```python
>>> import math
>>> math.sqrt(42)

>>> print('hello','world',sep=' ', )
```

**MATLAB**

```matlab
>> py.math.sqrt(42)

>> py.print('hello','world',... pyargs('sep',' ', ))
```
Data are automatically converted where possible

Otherwise convert explicitly

### MATLAB to Python Data Type Mapping

When calling a Python® function, MATLAB® converts MATLAB data into types that best represent the data to the Python language.

#### Pass Scalar Values to Python

<table>
<thead>
<tr>
<th>MATLAB Input Argument Type — Scalar Values Only</th>
<th>Resulting Python <code>py</code> Type</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>double single</td>
<td>float</td>
<td>Use Python Numeric Variables in MATLAB</td>
</tr>
<tr>
<td>Complex single</td>
<td>complex</td>
<td></td>
</tr>
<tr>
<td>Complex double</td>
<td></td>
<td></td>
</tr>
<tr>
<td>int8</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>uint8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>int16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>uint16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>int32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>uint32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>int64</td>
<td>int (version 2.7 only)</td>
<td></td>
</tr>
<tr>
<td>uint64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NaN</td>
<td>float(“nan”)</td>
<td></td>
</tr>
<tr>
<td>Inf</td>
<td>float(“inf”)</td>
<td></td>
</tr>
<tr>
<td>string scalar</td>
<td>str</td>
<td></td>
</tr>
</tbody>
</table>

Model Interoperability

- Call Python from MATLAB
- Call MATLAB from Python

Call MATLAB from Python
Why call 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 MATLAB from Python
To perform advanced analytics

- Calling MATLAB from Python
  - via MATLAB Engine API

```
cd(fullfile(matlabroot,'extern','engines','python'))
system('python setup.py install')
```

1. Start a MATLAB process

```
import matlab.engine
eng = matlab.engine.start_matlab()
```

2. Call MATLAB functions

```
x = eng.sqrt(float(43))
print(x)
```

6.557438524302
Integrate the MATLAB model in a Python environment

CallMATLABfromPython.ipynb

Data Access

- Call Python from MATLAB
- Call MATLAB from Python

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment

Predict Air Quality: Calling MATLAB from Python Using MATLAB Engine API

Import the weather data and use the MATLAB code to predict air quality.

This example uses the MATLAB Engine API, which runs a MATLAB session. More info and setup instructions are included here: https://www.mathworks.com/help/matlab/matlab-engine-for-python.html
Recap: Calling MATLAB from Python
Syntax differences when calling MATLAB from Python

**MATLAB**

```matlab
>> [s, sidx] = sort(x)

>> foo(x)

>> C = A + B
```

**Python**

```python
>>> s = eng.sort(x, nargout=2)

>>> eng.foo(x, nargout=0)

>>> C = eng.plus(A, B)
```
Data are automatically converted where possible

### Python Type to MATLAB Scalar Type Mapping

When you pass Python® data as input arguments to MATLAB® functions, the MATLAB Engine for Python converts the data into equivalent MATLAB data types.

<table>
<thead>
<tr>
<th>Python Input Argument Type — Scalar Values Only</th>
<th>Resulting MATLAB Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>float</td>
<td>double</td>
</tr>
<tr>
<td>complex</td>
<td>Complex double</td>
</tr>
<tr>
<td>int</td>
<td>int64</td>
</tr>
<tr>
<td>long (Python 2.7 only)</td>
<td>int64</td>
</tr>
<tr>
<td>inf</td>
<td>Inf</td>
</tr>
<tr>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>bool</td>
<td>logical</td>
</tr>
<tr>
<td>str</td>
<td>char</td>
</tr>
<tr>
<td>unicode (Python 2.7 only)</td>
<td>char</td>
</tr>
<tr>
<td>dict</td>
<td>Structure if all keys are strings not supported otherwise</td>
</tr>
</tbody>
</table>

### Python Container to MATLAB Array Type Mapping

<table>
<thead>
<tr>
<th>Python Input Argument Type — Container</th>
<th>Resulting MATLAB Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>matlab numeric array object (see MATLAB Arrays as Python Variables)</td>
<td>Numeric array</td>
</tr>
<tr>
<td>bytearray</td>
<td>uint8 array</td>
</tr>
<tr>
<td>bytes (Python 3.x)</td>
<td>uint8 array</td>
</tr>
<tr>
<td>bytes (Python 2.7)</td>
<td>char array</td>
</tr>
<tr>
<td>list</td>
<td>Cell array</td>
</tr>
<tr>
<td>set</td>
<td>Cell array</td>
</tr>
<tr>
<td>dict</td>
<td>Structure if all keys are strings not supported otherwise</td>
</tr>
</tbody>
</table>

Deployment
Generate Python library from MATLAB functions

Data Access
- Call Python from MATLAB
- Call MATLAB from Python

Co-Execution
- Call Python from MATLAB
- Call MATLAB from Python

Deployment
- Generate Python library from MATLAB functions
Generate Python library from MATLAB functions

Data Access

• Call Python from MATLAB
• Call MATLAB from Python

Deployment

Co-Execution

Generate Python library from MATLAB functions
Generate Python library from MATLAB functions

Co-Execution
- Call Python from MATLAB
- Call MATLAB from Python

Deployment
- Generate Python library from MATLAB functions
- Install library
Execute Python library from MATLAB functions

Predict Air Quality: Calling MATLAB from Python Using MATLAB Runtime

Import the weather data and use the MATLAB code to predict air quality.

Use the MATLAB Runtime to call the model and code, which does
Calling our function:

```
{"nargout":1,"rhs":["input"]}
```

Getting the result:

```
{"lhs":[{"mwdata":["output"],"mwsize":[1,6],"mwtype":"char"}]
```
Execute Python library from MATLAB functions

Air Quality Conditions

Determine air quality conditions in your area.

Location: Seattle, WA

Submit

The air quality is Good.
The current temperature is 30.15 F.
Share MATLAB App in the Web – Central Deployment

Data Access

Co-Execution
- Call Python from MATLAB
- Call MATLAB from Python

Deployment
MATLAB App Designer

App development for Non-Programmers

- Data Access
- Co-Execution
  - Call Python from MATLAB
  - Call MATLAB from Python
- Deployment
MATLAB Web App Server – Central Deployment

Data Access

Co-Execution
- Call Python from MATLAB
- Call MATLAB from Python

Deployment

• Call Python from MATLAB
• Call MATLAB from Python
Integrate your Production System in an IT ecosystem

Data Access

Co-Execution
- Call Python from MATLAB
- Call MATLAB from Python

Deployment

Data
- Streaming data
- Data stores

Production System
- MATLAB Production Server
  - Worker processes
  - Request Broker

Analytics Development
- Compiler SDK
- MATLAB
- Package & Deploy
- Model

Business Decisions
- Business Decisions
- Web App
- Dashboards
- Enterprise App

Big Data
- Big Data
- cloudera
- Spark
- databricks

Storage Layer
- Storage Layer
- elastic

Data
- Data
- Files

Deployment
- Integrate your Production System in an IT ecosystem
Use MATLAB Reference Architectures for easy cloud setup, Dockerfiles, and interfaces to OSS

- [https://github.com/mathworks-ref-arch/matlab-dockerfile](https://github.com/mathworks-ref-arch/matlab-dockerfile)

https://github.com/mathworks-ref-arch
MATLAB Integration for Jupyter

Data Access

Co-Execution
- Call Python from MATLAB
- Call MATLAB from Python

Deployment

MATLAB Integration for Jupyter (mathworks.com)
Summary:
Using MATLAB with Python

- **Access Data**
  - Weather App example

- **Interoperability**
  - Calling libraries written in Python from MATLAB
  - Calling MATLAB from Python

- **Deploy Apps & Algos**
  - Web App
  - Production API
Additional resources
Resources

- **General:**

- **Python from MATLAB:**

- **MATLAB from Python:**
  - **MATLAB Engine API:**

  - **MATLAB Compiler SDK:**
    - [https://www.mathworks.com/help/compiler_sdk/python_packages.html](https://www.mathworks.com/help/compiler_sdk/python_packages.html)

  - **Data type conversions:**

- **Example:**
  - [https://github.com/mathworks/matlab-with-python](https://github.com/mathworks/matlab-with-python)
Cheatsheet

Using MATLAB® and Python® Together

The ≥ icon provides links to relevant sections of the MATLAB documentation to learn more.

Call Python in MATLAB

- Access settings and status of Python interpreter:
  ```
  >> ps = pyenv
  ```
- Specify version to use:
  ```
  >> ps = pyenv('Version',3.7)
  ```
- Call Python modules and functions:
  ```
  py.module_name.function_name
  ```
  ```
  >> py.math.sqrt(42)
  ```
- Pass keyword arguments
  ```
  Use pyargs to pass keyword arguments
  ```
  ```
  >> foo(5, bar=42)
  ```
  ```
  >> py.foo(5, pyargs('bar', 42))
  ```
- Reload modules
  ```
  Reload the module after making updates:
  ```
  >> py.importlib.reload(module)
  ```

Call MATLAB in Python

- Install MATLAB Engine API for Python ≥
  ```
  Run setup.py from an OS command window
  ```
  ```
  $ cd /matlabroot/extern/engines/python
  $ python setup.py install
  ```
- Call MATLAB functions
  ```
  Import the module and start the engine
  ```
  ```
  >> import matlab.engine
  ```
  ```
  >> eng = matlab.engine.start_matlab()
  ```
  Call functions through the engine
  ```
  >> x = eng.sqrt(42.0)
  ```
  Capture multiple outputs
  ```
  >> x = eng.god(42,0,8,0,nargout=3)
  ```
- Stop the engine
  ```
  >> eng.quit()
  ```

Create Python Package

- Package MATLAB functions ≥
  ```
  Use the Library Compiler App to create a Python package for MATLAB functions
  ```
- Invoke MATLAB functions from the Python package
  ```
  >> import PackageName
  ```
  ```
  >> pkg = PackageName.initialize()
  ```
  ```
  >> result = pkg.foo()
  ```
- Close package
  ```
  >> pkg.terminate()
MATLAB Answers – tag:"python"

https://www.mathworks.com/matlabcentral/answers/?term=tag%3A%22python%22
Additional Resources

- **NEW** Release

- Predictive Analytics
  - [https://www.mathworks.com/discovery/predictive-analytics.html](https://www.mathworks.com/discovery/predictive-analytics.html)

- Deep Learning

- Reinforcement Learning