MathWorks®
Accelerating the pace of engineering and science

Calling MATLAB from Python

This reference shows common use cases but is by no means comprehensive. The >> icon provides links to relevant sections of the MATLAB® documentation. For general information, see https://www.mathworks.com/products/matlab/matlab-and-python.html.

Setup

Requirements >>

Install MATLAB Engine API for Python
To use MATLAB Engine API for Python®, you need to have both installed on the system. Install the engine API as a Python package.

Install Using `pip` >>
From the system prompt:

```
$ python -m pip install
matlabengine
```

Install Using `setup.py`
Run the `setup.py` file from the system prompt:

```
$ cd [matlabroot]/extern/
engines/python
$ python setup.py install
```

For troubleshooting, ensure privileges (run as administrator) and check the `PYTHONPATH` and system paths.

MATLAB Engine API

Use MATLAB Engine API to call MATLAB from Python. >>
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.
gcd(42.0,8.0,nargout=3)
>>> x = eng.plot(x,y,nargout=0)
```

Stop the engine:

```
>>> x = eng.exit()
```

Data Type Conversions

Data types will be automatically converted where possible. >>

Python	MATLAB
`float`	`double`
`complex`	`complex double`
`int`	`int64`
`float(nan)`	`NaN`
`float(inf)`	`Inf`
`bool`	`logical`
`str`	`char`
`dict`	`struct`
`list`	`cell array`
`set`	`cell array`
`tuple`	`cell array`

You can create MATLAB arrays in Python to pass data easily to MATLAB functions:

```
>>> x = matlab.
double([1,4,9])
```

Using MATLAB Apps

Use MATLAB Engine Workspace >>
The MATLAB engine workspace can be used to access variables from MATLAB and Python and facilitates app use.
Add a variable to the workspace:

```
>>> x = 4.0
>>> eng.workspace['y'] = x
```

Access the variable MATLAB workspace:

```
>>> x = eng.workspace['y']
```

Execute statements using the `eval` function:

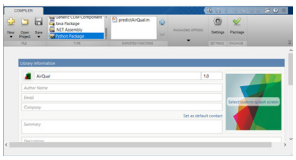
```
>>> a = eng.eval('sqrt(y)')
```

You can open apps in MATLAB from Python by using the command for the app:

```
>>> eng.signalAnalyzer()
>>> eng.classificationLearner()
```

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

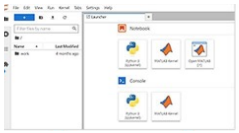
```
>>> import PackageName
>>> pkg = PackageName.
initialize()
>>> result = pkg.Foo()
```

Close Package

```
>>> pkg.terminate()
```

MATLAB in Other IDEs

You can use MATLAB from Jupyter® or VSCode. The installation and setup steps are detailed in the links included here. >>



Install MATLAB Integration for Jupyter >>
For example, install the package to execute MATLAB from Jupyter.

```
$ python pip install
jupyter-matlab-proxy
$ install-matlab-kernelspec
```

Now you can use MATLAB in a browser or run code in your notebook with the MATLAB kernel.