Instruction Manual

Team Disrupt

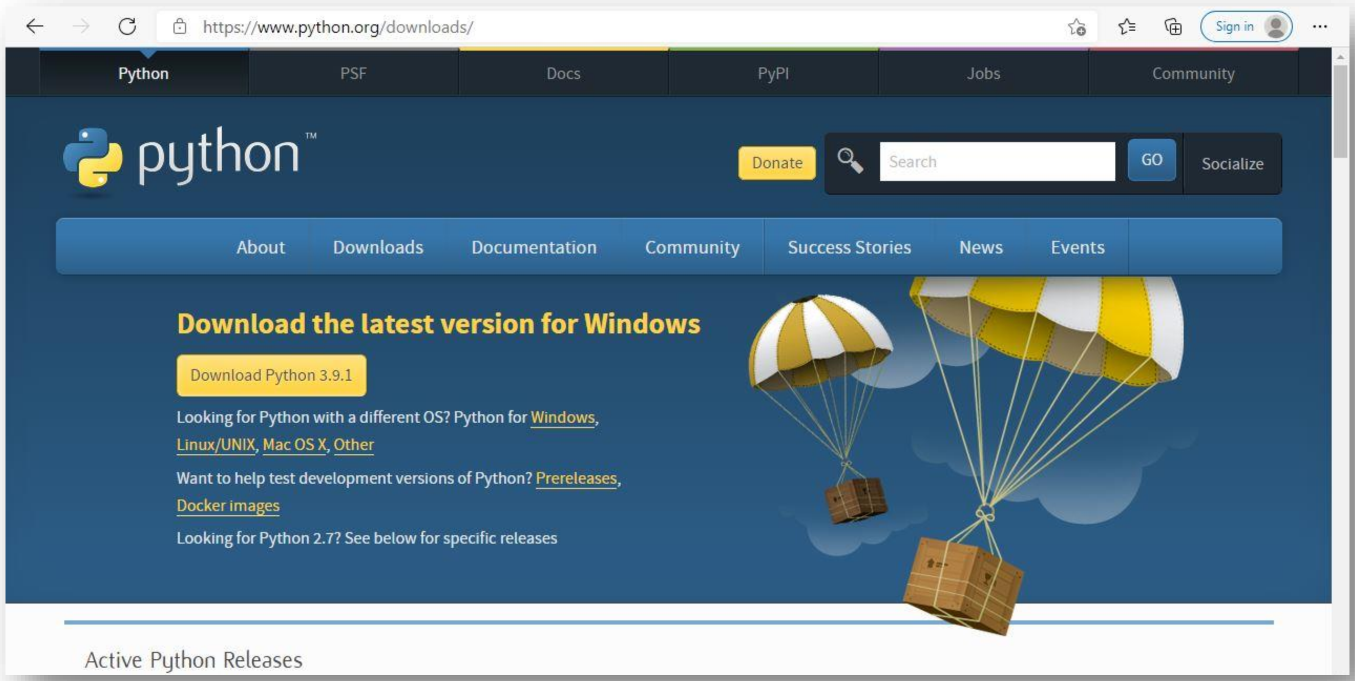
A Python package for 1-D blood flow simulation

By Sanaul Malik Shaheer, V Abhijith Narayan Rao, Sharon Sajimakil and Raidha Anwar

Installing Python

Step 1: Download the installer

An installer has been included in the files of the script, but in case any issues arise, it can also be downloaded from the official Python website. You can see on the Python home page (www.python.org), in the download section as shown in the below figure. The file download size is around 26.9 Mb. Download the installer from here. By default, a 32-bit version of Python gets downloaded.



Download the latest version for Windows

Download Python 3.9.1

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Prereleases, Docker images

Looking for Python 2.7? See below for specific releases

Active Python Releases

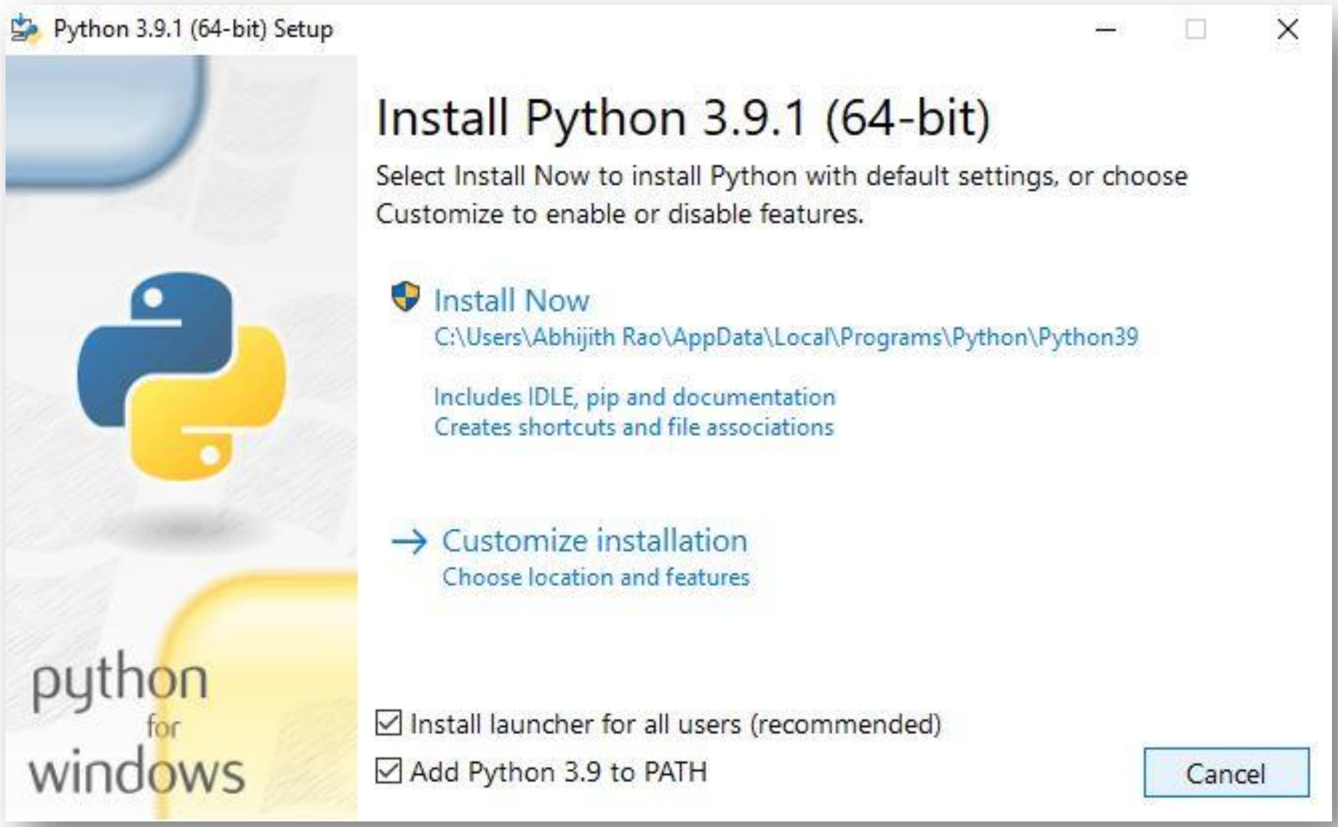
Step 2: Start the installation process

Double click on the installer file to start the installation process. The process is any installation you would have don on windows 10.

If you see any security warnings, click yes to continue.

Step 3: Start the installation process

Make sure you check the add Python 3.x.x to path checkbox before processing.


Python 3.9.1 (64-bit) Setup

Install Python 3.9.1 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\Abhijith Rao\AppData\Local\Programs\Python\Python39

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

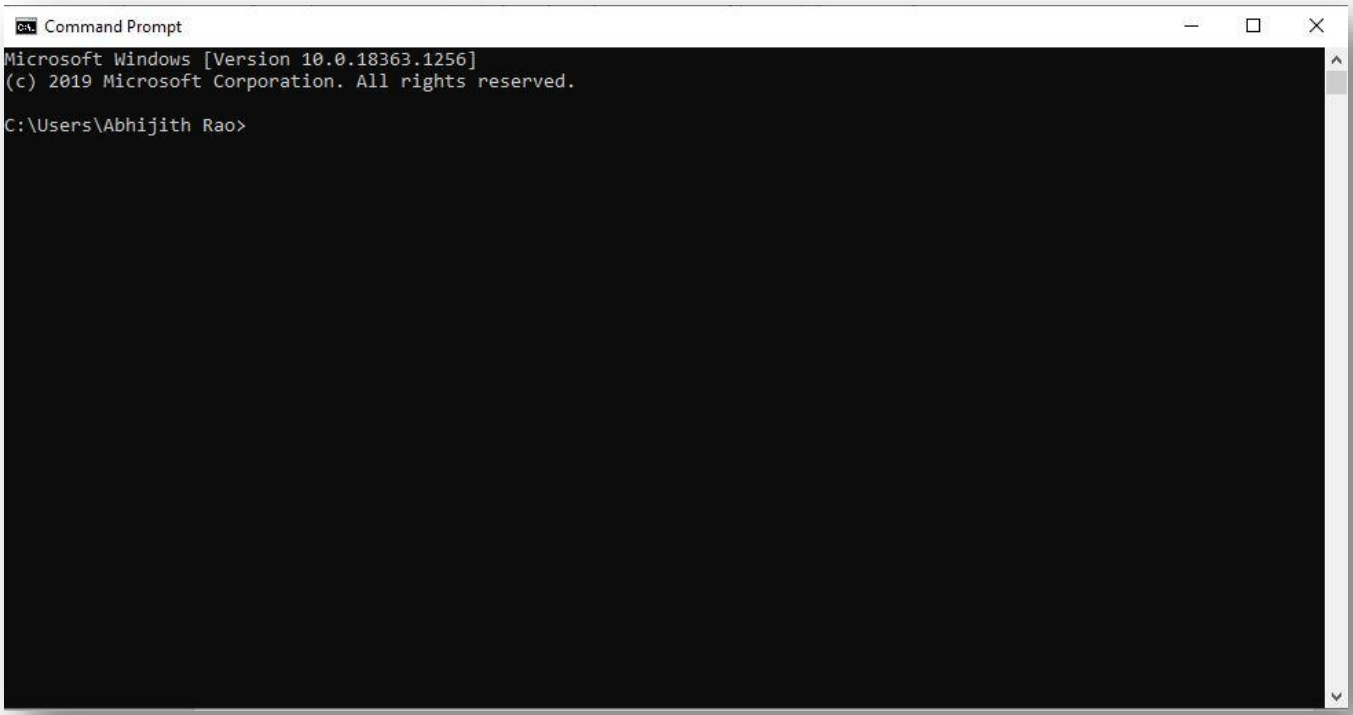
☑ Install launcher for all users (recommended)
☑ Add Python 3.9 to PATH

Once you have checked the required tick-box, click install now, and the wizard will guide you through the rest.

Installing Elixir

Step 1: Open the command prompt

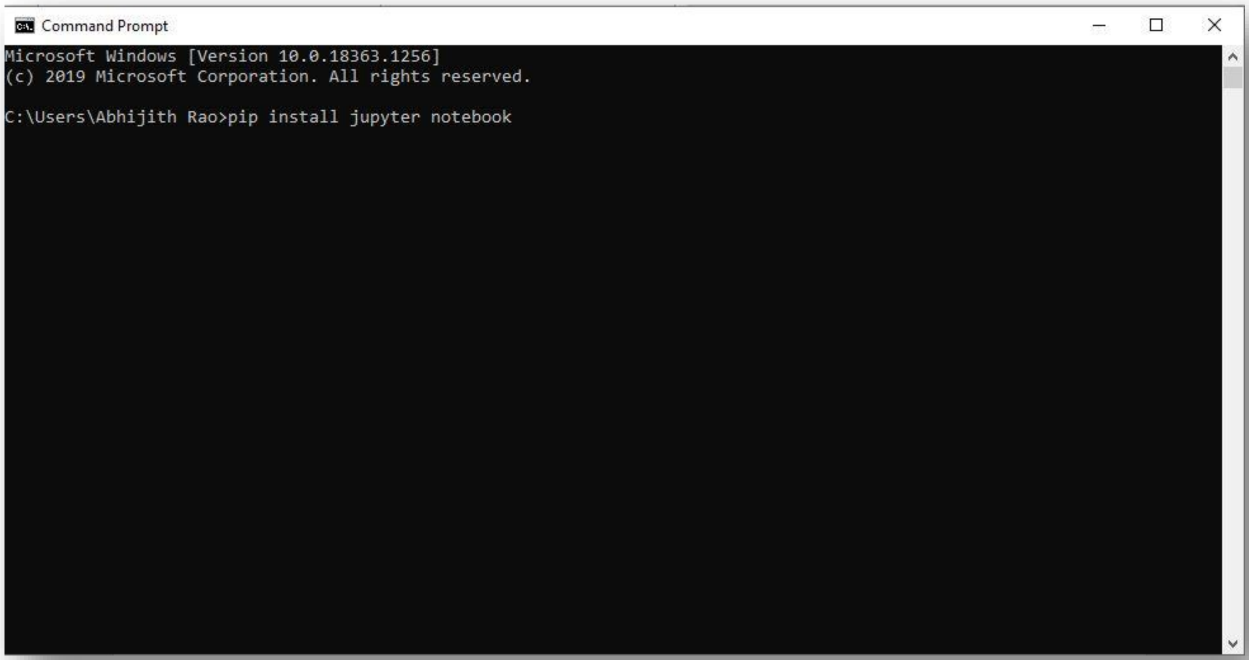
Not much hassle here, simply type cmd into the start menu and open the command prompt which must look something like this.



```
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Abhijith Rao>
```

Step 2: Using pip to install the packages

Let's begin by first installing jupyter notebook, which we will be using to act as the IDE for the simulation. Type pip install jupyter notebook and press enter to install jupyter notebook.

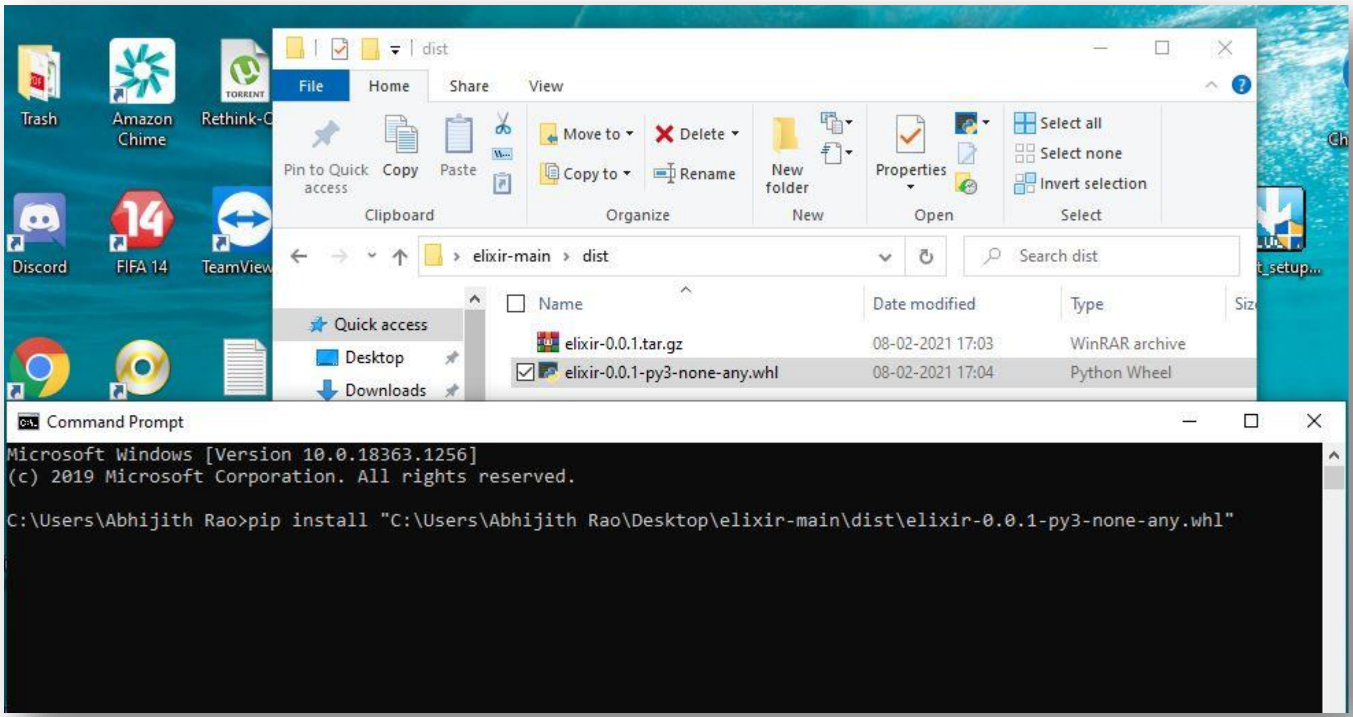


```
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Abhijith Rao>pip install jupyter notebook
```

Followed by this navigate to the folder where elixir-0.0.1-py3-none-any.whl is located using cd in the terminal. Once there, copy and paste the following command into the terminal.

pip install elixir-0.0.1-py3-none-any.whl

You could also simply type pip install and then drag and drop the file into the terminal to install it.



Name	Date modified	Type
elixir-0.0.1.tar.gz	08-02-2021 17:03	WinRAR archive
elixir-0.0.1-py3-none-any.whl	08-02-2021 17:04	Python Wheel

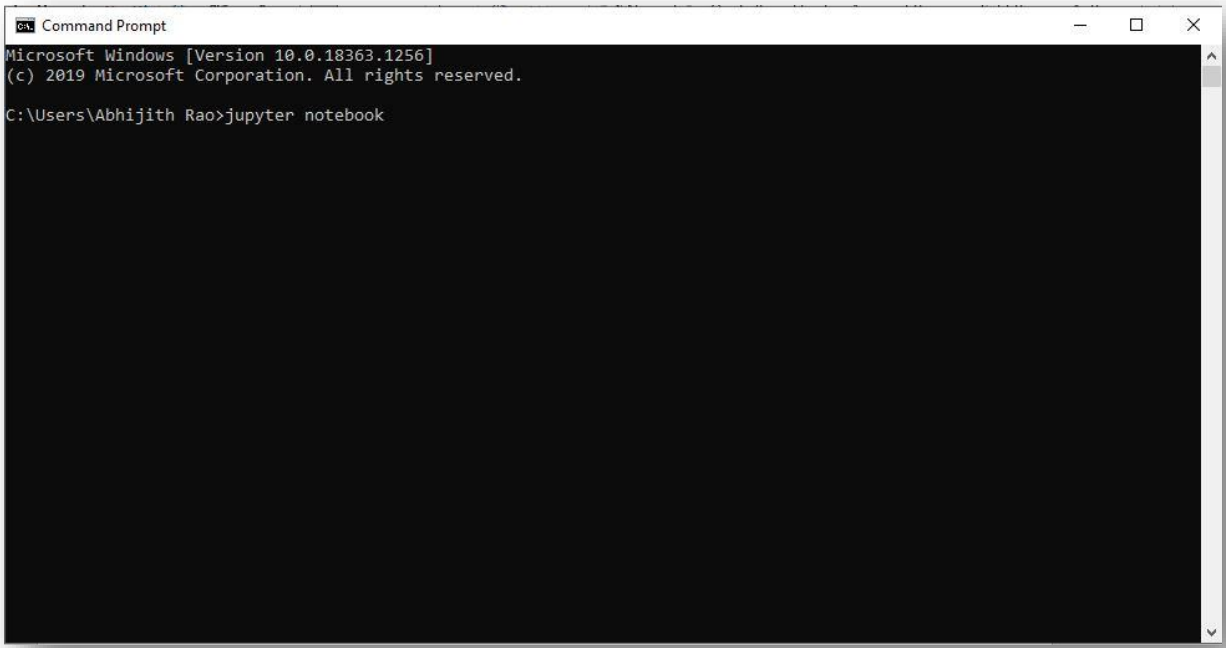
```
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Abhijith Rao>pip install "C:\Users\Abhijith Rao\Desktop\elixir-main\dist\elixir-0.0.1-py3-none-any.whl"
```

Pressing enter will now install elixir into your system.

Simulating Blood Flow Using Elixir

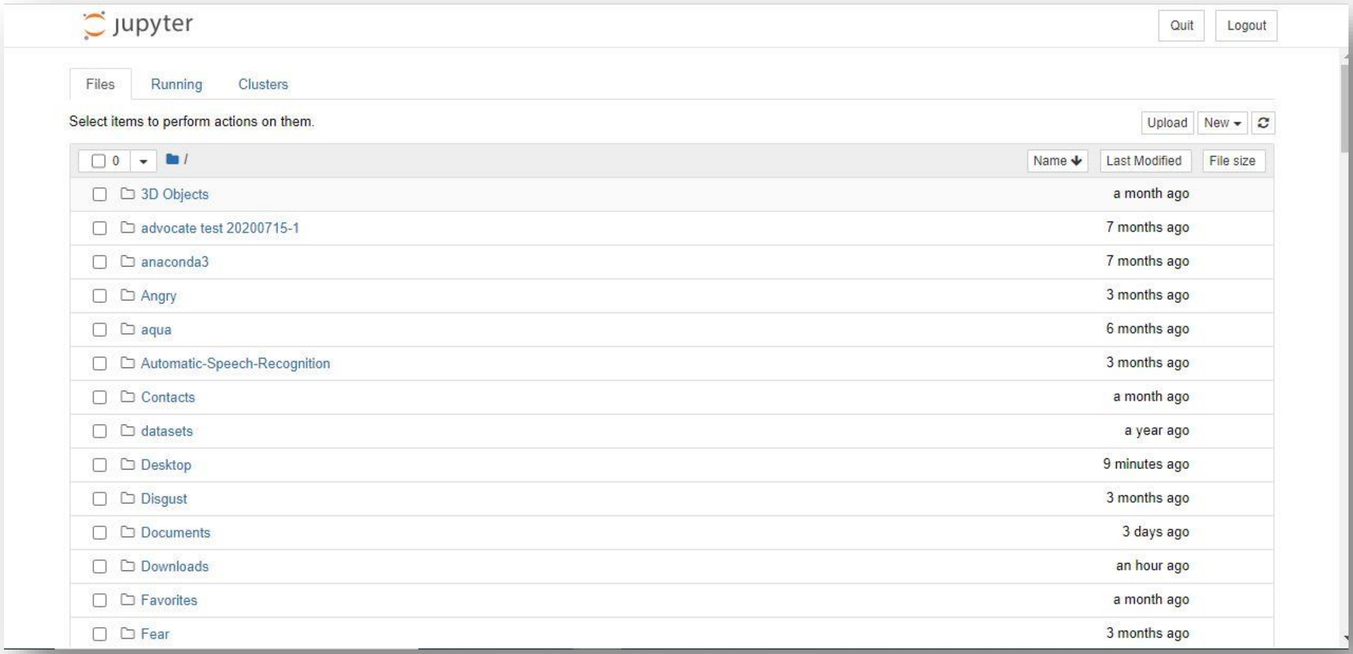
Step 1: Open the command prompt

Once you have opened command prompt, type in jupyter notebook as shown.



```
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Abhijith Rao>jupyter notebook
```

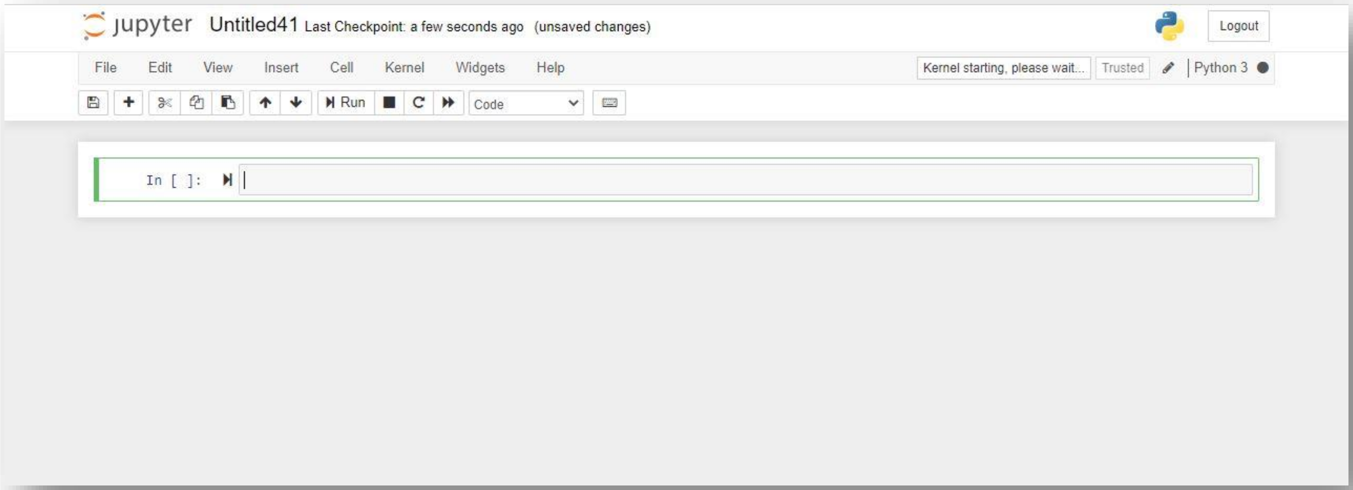
Pressing will then open the jupyter notebook in your default browser.



Select items to perform actions on them.

```
3D Objects                      a month ago
advocate test 20200715-1        7 months ago
anaconda3                       7 months ago
Angry                           3 months ago
aqua                            6 months ago
Automatic-Speech-Recognition    3 months ago
Contacts                        a month ago
datasets                        a year ago
Desktop                         9 minutes ago
Disgust                         3 months ago
Documents                       3 days ago
Downloads                       an hour ago
Favorites                       a month ago
Fear                            3 months ago
```

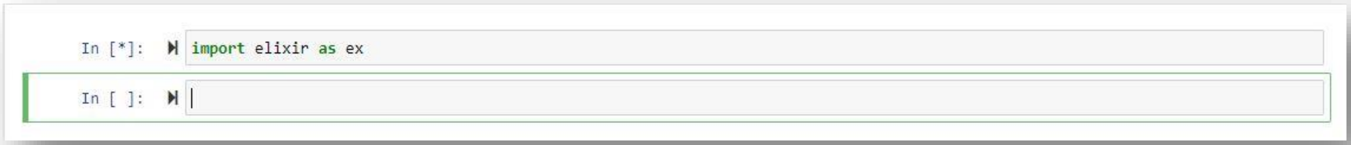
Click “New”, and choose Python3 to start a new notebook instance.


jupyter
Untitled41 Last Checkpoint: a few seconds ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Kernel starting, please wait...
Trusted
Python 3
Run
Code

```
In [ ]:
```

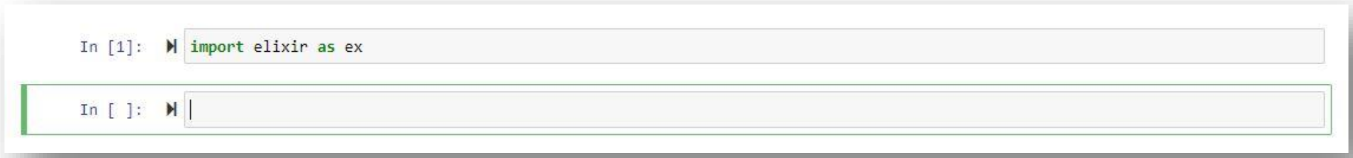
Step 2: Import Elixir

We import Elixir the same we import any other library in python, by typing in “import Elixir as ex”, and then pressing Shift+Enter to execute the cell.



```
In [*]:  import elixir as ex

In [ ]:
```

We notice that the command is executed when the asterisk (*) is replaced by the total number of commands executed thus far.

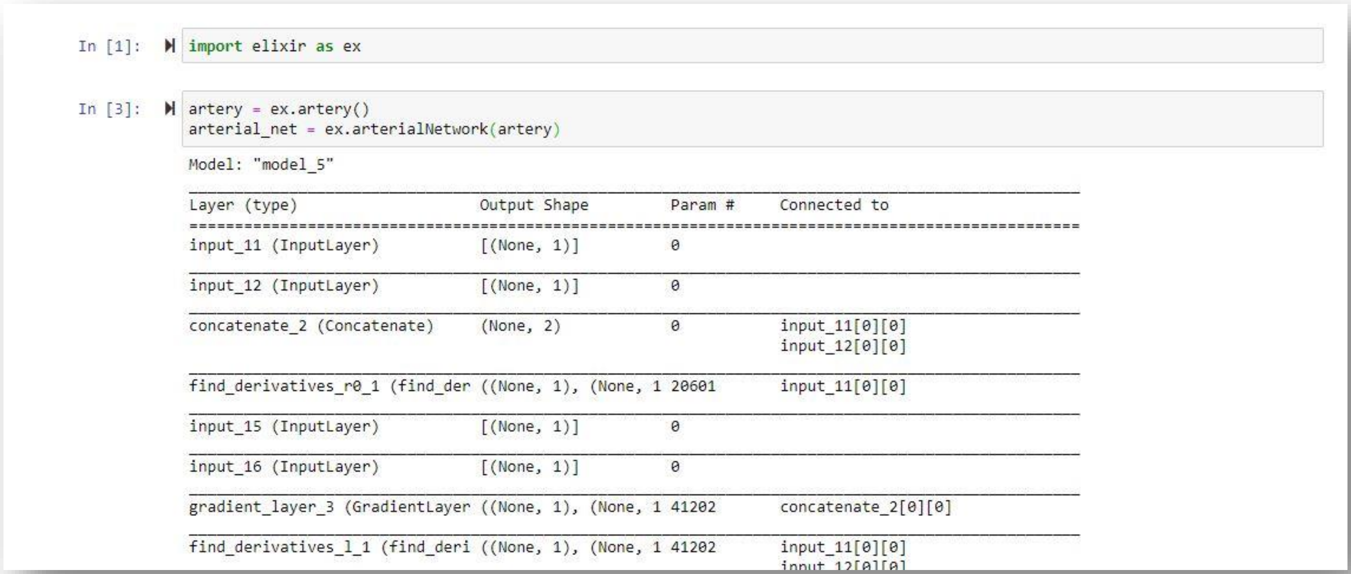


```
In [1]:  import elixir as ex

In [ ]:
```

Step 3: Defining the Artery

We now proceed to defining the artery. Here, we explain the definition of an artery with only the default values. To know more about the various modifications, you could bring to the artery you are simulating please visit the project's Github page, mentioned at the end.

We define an artery as follows:

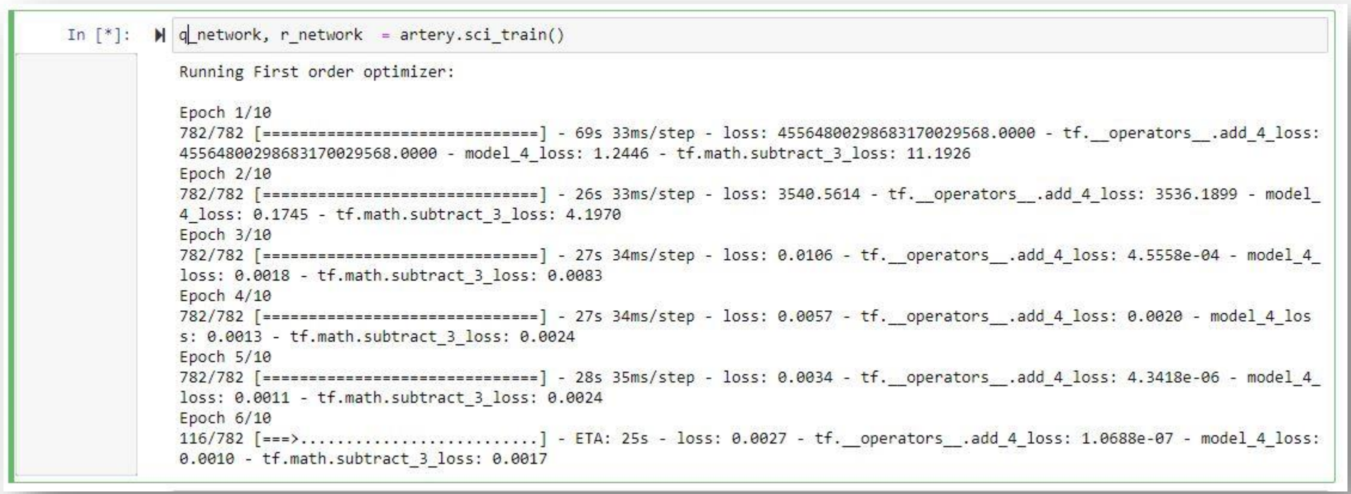


```
In [1]:  import elixir as ex

In [3]:  artery = ex.artery()
         arterial_net = ex.arterialNetwork(artery)

         Model: "model_5"
         __________________________________________________________________________________________________
         Layer (type)                    Output Shape         Param #     Connected to
         ==================================================================================================
         input_11 (InputLayer)           [(None, 1)]          0
         __________________________________________________________________________________________________
         input_12 (InputLayer)           [(None, 1)]          0
         __________________________________________________________________________________________________
         concatenate_2 (Concatenate)     (None, 2)            0           input_11[0][0]
                                                                          input_12[0][0]
         __________________________________________________________________________________________________
         find_derivatives_r0_1 (find_der ((None, 1), (None, 1 20601       input_11[0][0]
         __________________________________________________________________________________________________
         input_15 (InputLayer)           [(None, 1)]          0
         __________________________________________________________________________________________________
         input_16 (InputLayer)           [(None, 1)]          0
         __________________________________________________________________________________________________
         gradient_layer_3 (GradientLayer ((None, 1), (None, 1 41202       concatenate_2[0][0]
         __________________________________________________________________________________________________
         find_derivatives_l_1 (find_deri ((None, 1), (None, 1 41202       input_11[0][0]
                                                                          input_12[0][0]
```

This step automatically defines all the parameters of the artery. It also defines the neural network we would be using to simulate the blood flow, the table that generates, represents the same.

Step 4: Training the models

We can now proceed to training the models. This can simply be done by entering, "q_network, r_network = artery.sci_train()".



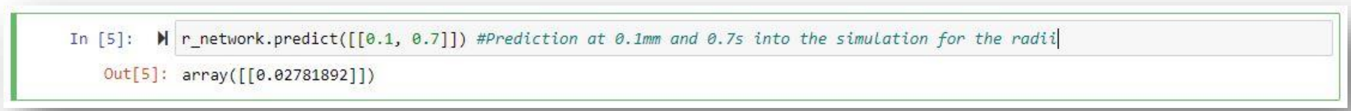
```
In [*]:  q_network, r_network  = artery.sci_train()

         Running First order optimizer:

         Epoch 1/10
         782/782 [==============================] - 69s 33ms/step - loss: 455648002986831700295680.0000 - tf.__operators__.add_4_loss:
         455648002986831700295680.0000 - model_4_loss: 1.2446 - tf.math.subtract_3_loss: 11.1926
         Epoch 2/10
         782/782 [==============================] - 26s 33ms/step - loss: 3540.5614 - tf.__operators__.add_4_loss: 3536.1899 - model_
         4_loss: 0.1745 - tf.math.subtract_3_loss: 4.1970
         Epoch 3/10
         782/782 [==============================] - 27s 34ms/step - loss: 0.0106 - tf.__operators__.add_4_loss: 4.5558e-04 - model_4_
         loss: 0.0018 - tf.math.subtract_3_loss: 0.0083
         Epoch 4/10
         782/782 [==============================] - 27s 34ms/step - loss: 0.0057 - tf.__operators__.add_4_loss: 0.0020 - model_4_los
         s: 0.0013 - tf.math.subtract_3_loss: 0.0024
         Epoch 5/10
         782/782 [==============================] - 28s 35ms/step - loss: 0.0034 - tf.__operators__.add_4_loss: 4.3418e-06 - model_4_
         loss: 0.0011 - tf.math.subtract_3_loss: 0.0024
         Epoch 6/10
         116/782 [===>..........................] - ETA: 25s - loss: 0.0027 - tf.__operators__.add_4_loss: 1.0688e-07 - model_4_loss:
         0.0010 - tf.math.subtract_3_loss: 0.0017
```

This will run the first order trainer to train the models to simulate blood flow.

NOTE: If the error at the tenth epoch is very large (>>0.003), then please restart the simulation. This occurs because of bad initialization, and no remedy to the problem has been found as of yet.

Step 5: Making predictions

Once, the networks have converged to an error of approximately 0.0036, we can start utilizing the networks to make predictions on the flow and radii of the artery at any point in space and time.



```
In [5]:  r_network.predict([[0.1, 0.7]]) #Prediction at 0.1mm and 0.7s into the simulation for the radii
 Out[5]: array([[0.02781892]])
```

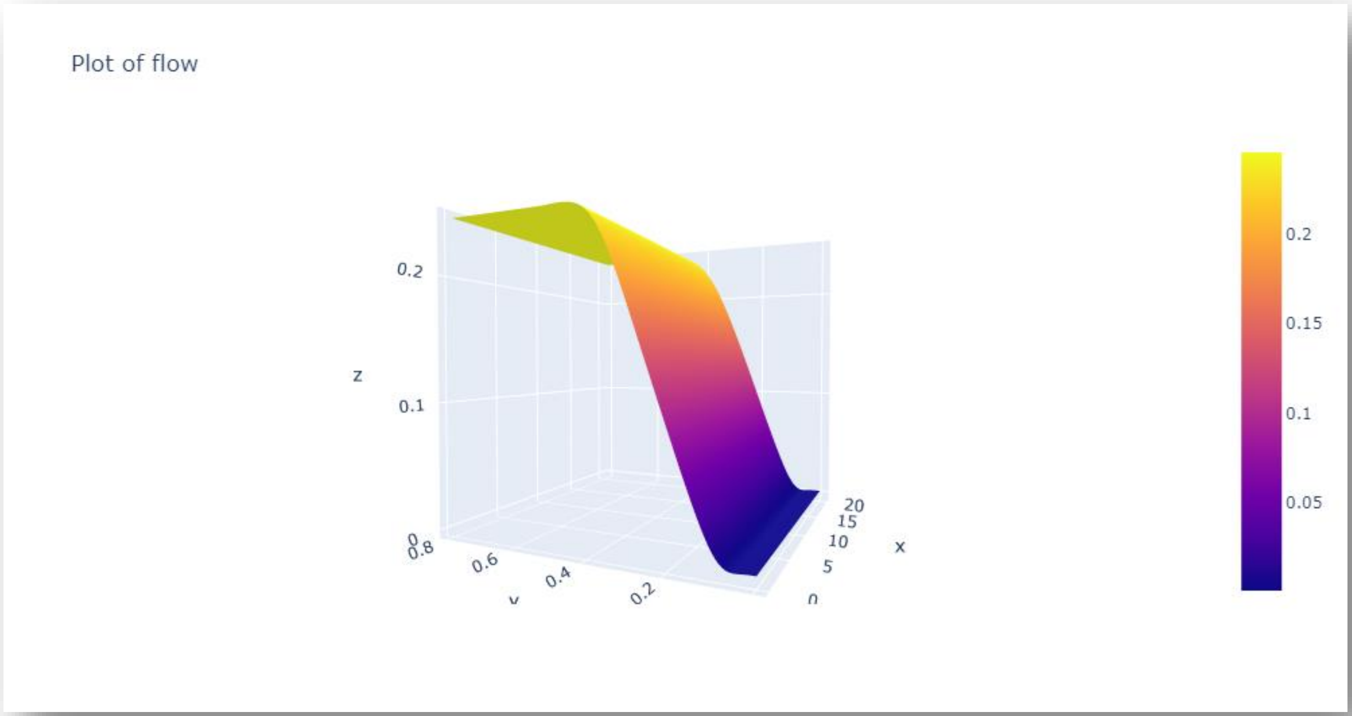
NOTE: The above arterial networks have been trained in the domain (0, 20.8) for length and (0, 0.8) for time. Make sure to predict within the domain itself for accurate results or extend the domain to make predictions appropriately.

Step 6: Plotting Graphs

Elixir allows plotting beautiful interactive plots which just one command. Once the models have been training, please type

“artery.plot_flow()” or “artery.plot_radius()”

To plot the appropriate graphs.


Plot of flow
z
y
x