**1. Establishing connection between Coppeliasim and Python (or MATLAB)**

* Prerequisites:
  + Successful completion of Week 0 module.
* Steps to execute your Coppeliasim scene with your Python (or MATLAB) code:
  + Run the Coppeliasim scene (green play button). Since the child script of the Pioneer is disabled, it would not exhibit any motion.
  + With the scene running in the background, run your Python/MATLAB file. Now, your robot should move according to the instructions given in your code.
  + If the connection is not established properly, you would see the message in your Python console.
* Command to start the RemoteAPI Binding
  + **This is already done in the scene we have provided for you**. This is for your reference when you want to create your own scene.
  + Select any shape from the scene. If there’s none, create a primitive shape.
  + Right click on its name in the Scene Hierarchy window and select **Add >> Associated child script >> Threaded.**
  + A small picture of a paper will appear next to its name in the Scene Hierarchy window.
  + Double click the icon. A programming window will open up. Add the following line as initialisation code (the comments will guide you).

simRemoteApi.start(19999)

This line should be executed only once to start the connection setup. DO NOT REMOVE anything from that script.

**2. Understanding scripts**

All robots in Coppeliasim have their own in-built scripts. That's what makes them what they are and defines how they work. When you ran your Coppeliasim scene in Week 0, you saw that the Khepera automatically started moving around and avoiding obstacles. That was because of its associated child script.

But why does the Pioneer not move like that? That’s because when we gave you the scene, we already disabled its script because we will be writing our own script for our robot to follow.

To enable or disable child scripts, go to Tools >> Scripts and click on ‘Non-threaded child script (Pioneer\_p3dx)’. The checkbox below should indicate if it is disabled or not. Upon enabling it, your Pioneer will move and avoid obstacles automatically. Keep it DISABLED to use your code.

**3. Understanding the Base Code and RemoteAPI functions**

If you find the base code intimidating, worry not. The objects and methods you see there are the RemoteAPI functions – they aid communication between your robot in Coppeliasim and the code in Python.

So, all these functions that help us communicate with coppeliasim are in the sim.py file that you copied into your project folder in week 0. We first import that file. Now we can use all the function available.

simxFinish() – It closes any existing connection with the coppeliasim server

simxStart() – It establishes a connection with the coppeliasim server. It returns a value which is -1 if the code is unable to establish connection with the server.

simxGetObjectHandle() – Gets you access to the motors. It returns two integer values, one is the errorcode(check the api documentation to know what each errorcode value means) and the second is an integer that represents that object called objectHandle. We have used this to access the motors. Since the motors are essentially rotational joints, we can use this objectHandle to control this joint. We can set its velocity or position much like we control any motor.

We encourage you to go through the [API documentation](https://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsPython.htm) to look at what functions are available for you to use.

Useful Links:

1. [04: Python Robot Simulation with V-Rep/CoppeliaSim | Tutorial - YouTube](https://www.youtube.com/watch?v=SQont-mTnfM)
2. [Remote API functions (Python) (coppeliarobotics.com)](https://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsPython.htm)
3. [(1364) 01: Line-Following Robot in V-Rep / CoppeliaSim | Tutorial - YouTube](https://www.youtube.com/watch?v=xI-ZEewIzzI)(this is for those who want to explore v-rep to the fullest extent)