Video tutorials for all installations

Watch the video tutorial according to your Operating system and follow the instructions on this installation guide.

Windows: https://youtu.be/mD6IEto-e4s

Ubuntu: https://youtu.be/SQNMtq07dpw

Mac: https://www.youtube.com/watch?v=sIuFBj0bCPI

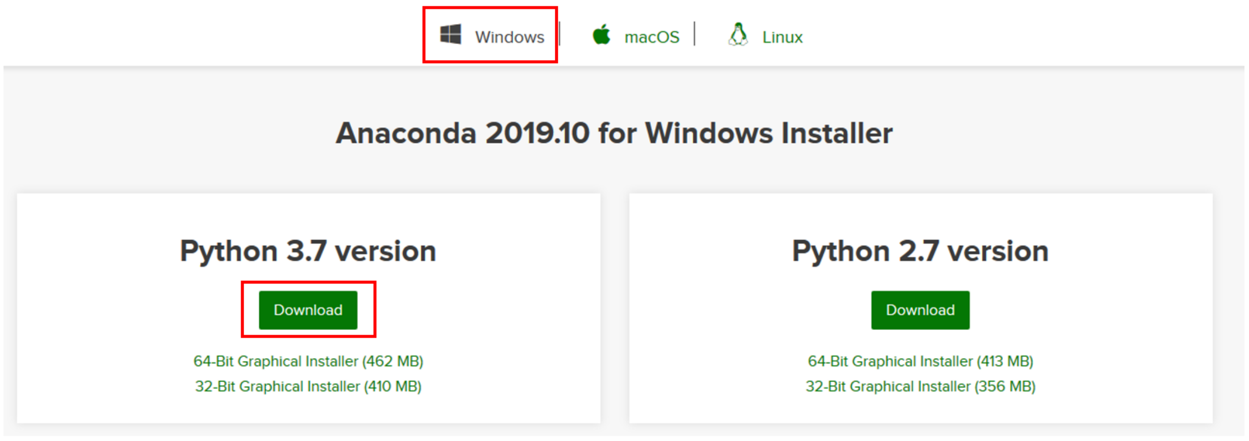
Instructions to preform the installation of Anaconda, python, keras, tensorflow and the v-rep python API.

Step 1. Install the necessary packages for a deep learning environment on your computer

STEP BY STEP WINDOWS INSTALLATION INSTRUCTIONS

Follow the installation video instructions and the following steps for creating a AI environment on your windows computer.

1. Visit Anaconda.com/downloads
2. Select your operating system (Windows)


Windows
macOS
Linux
Anaconda 2019.10 for Windows Installer
Python 3.7 version
Download
64-Bit Graphical Installer (462 MB)
32-Bit Graphical Installer (410 MB)
Python 2.7 version
Download
64-Bit Graphical Installer (413 MB)
32-Bit Graphical Installer (356 MB)

3. Download the ***.exe*** installer of the Python 3.7 version
4. Open and run the ***.exe*** installer
5. Open the **Anaconda Prompt** and run some Python code

Now create a conda environment for installing tensorflow and keras:

Open the anaconda prompt (right click-> run as administrator)

In the console type the following command:

```
conda create -n tensorflow_cpu pip python=3.6
```

The command will create a conda environment called tensorflow_cpu that uses python 3.6 and install pip. A conda environment will create a separate environment where all your installations will occur, it is important to create an environment before proceding with the installation so that all your changes happen in a controlled space and don't interfere with other packages installed on your computer.

Now activate the environment you create to start making changes in it, by running the following command:

```
activate tensorflow_cpu
```

Once you activate the conda virtual environment all the changes you make will be reflecting inside the environment. So, we can proceed with the tensorflow and keras installations, run on the console the following commands one by one:

```
pip install --upgrade tensorflow==1.13.1

pip install keras

pip install pillow, matplotlib

conda install pandas
```

STEP BY STEP UBUNTU INSTALLATION INSTRUCTIONS

Follow the installation video instructions and the following steps for creating a AI environment on your ubuntu computer.

Retrieve the latest distribution of Anaconda by downloading it from the following webpage

https://www.anaconda.com/distribution/

Assuming that the downloaded file is located in the folder /tmp run the following commands from your console:

```
$ cd /tmp
$ curl -O https://repo.anaconda.com/archive/Anaconda3-2020.02-Linux-x86_64.sh
```

Run the Anaconda script (check you are in the folder where the file has downloaded and that the name shown is the correct name of the downloaded file).

```
$ bash Anaconda3-2020.02-Linux-x86_64.sh
```

Activate the installation

```
$ source ~/.bashrc
```

Test Installation

```
$ conda list
```

Create a conda environment for installing tensorflow and keras:

```
$ conda create -n tensorflow_cpu pip python=3.6
```

The command will create a conda environment called tensorflow_cpu that uses python 3.6 and install pip. A conda environment will create a separate environment where all your installations will occur, it is important to create an environment before proceding with the installation so that all your changes happen in a controlled space and don't interfere with other packages installed on your computer.

Now activate the environment you create to start making changes in it, by running the following command:

```
$ activate tensorflow_cpu
```

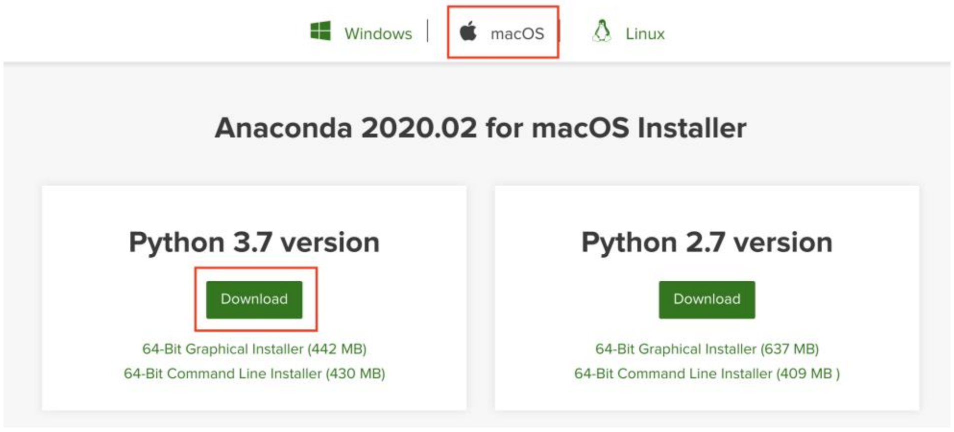
Once you activate the conda virtual environment all the changes you make will be reflecting inside the environment. So, we can proceed with the tensorflow and keras installations, run on the console the following commands one by one:

```
$ pip install --upgrade tensorflow==1.13.1

$ pip install keras

$ pip install pillow matplotlib

$ conda install pandas
```

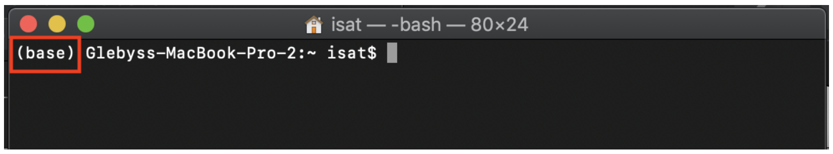
STEP BY STEP MAC-OS INSTALLATION INSTRUCTIONS

Follow the installation video instructions and the following steps for creating a AI environment on your windows computer.

1. Visit Anaconda.com/downloads
2. Select your operating system (MacOs)
3. Download the ***.exe*** installer of the Python 3.7 version


Windows
macOS
Linux
Anaconda 2020.02 for macOS Installer
Python 3.7 version
Download
64-Bit Graphical Installer (442 MB)
64-Bit Command Line Installer (430 MB)
Python 2.7 version
Download
64-Bit Graphical Installer (637 MB)
64-Bit Command Line Installer (409 MB)

4. Open and run the ***.dmg*** installer. Follow all the installation instructions. When done, you can place the installer in the Trash.
5. Open the spotlight using command(⌘)+space or by clicking the spotlight icon 🔍. Type "**terminal**" and open the terminal.
6. In the terminal you should see (base) at the beginning of the command prompt:



```
(base) Glebyss-MacBook-Pro-2:~ isat$
```

7. Type the following command in the terminal:

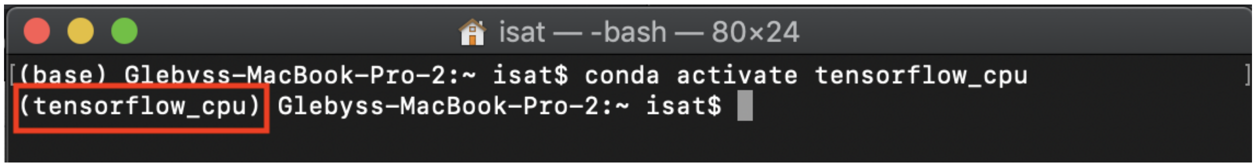
```
conda create -n tensorflow_cpu pip python=3.6
```

The command will create a conda environment called tensorflow_cpu that uses python 3.6 and install pip. A conda environment will create a separate environment where all your installations will occur, it is important to create an environment before proceeding with the installation so that all your changes happen in a controlled space and don't interfere with other packages installed on your computer.

8. Now activate the environment you create to start making changes in it, by running the following command:

```
conda activate tensorflow_cpu
```

If the activation was successful, the new environment should be visible at the beginning of the command prompt.



```
(base) Glebyss-MacBook-Pro-2:~ isat$ conda activate tensorflow_cpu
(tensorflow_cpu) Glebyss-MacBook-Pro-2:~ isat$
```

9. Once you activate the conda virtual environment all the changes you make will be reflecting inside the environment. Sov we can proceed with the tensorflow and keras installations, run on the console the following commands one by one:

```
conda install tensorflow=1.13.1

conda install keras

conda install pillow matplotlib pandas
```

NOTE:

If you encounter any errors after testing the installation, such as:

```
ModuleNotFoundError: No module named 'modulename'
```

Try the following steps:

9.1 First, try to install the module through conda:

```
conda install modulename
```

9.2 If conda cannot find the package, that means the module has a different name. Google the command conda install modulename and find out the real name of the package. Then, redo the pip installation with the correct name:

```
conda install new_modulename
```

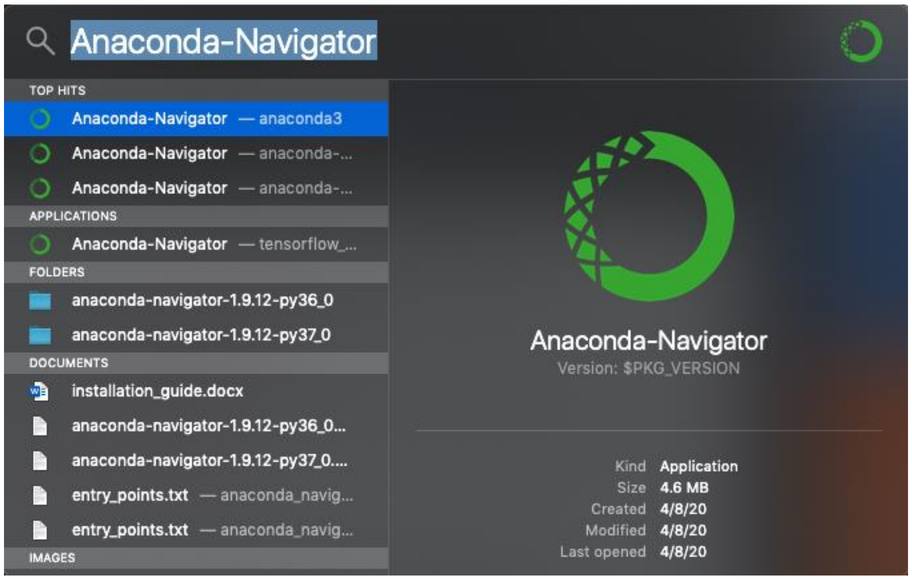
9.3 If your import or installation step still fails after this, install the module using pip instead:

```
pip install new_modulename
```

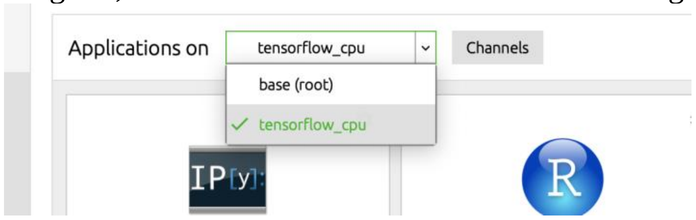
10. To install the python editor, we first install the anaconda-navigator by running in the terminal:

```
conda install anaconda-navigator
```

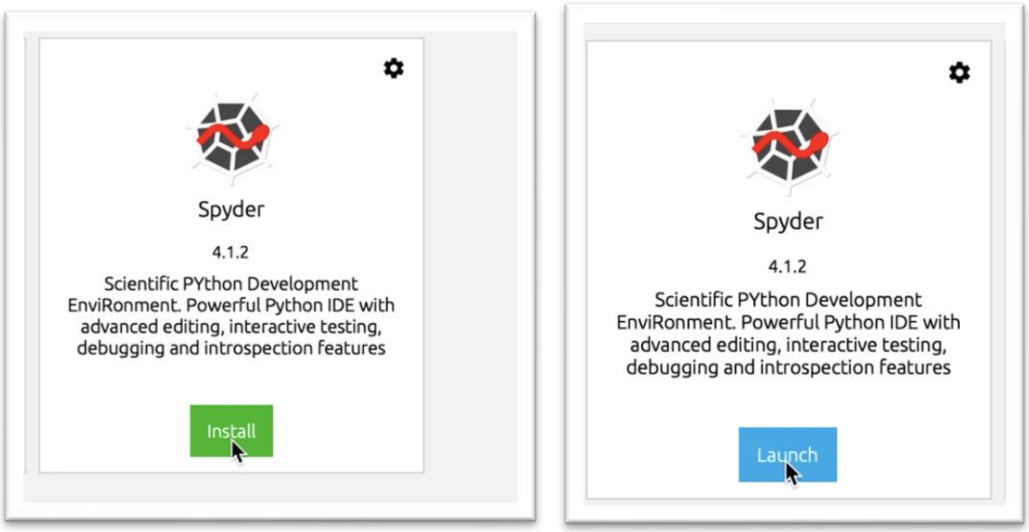
11. Then, open the spotlight (command(⌘)+space or by clicking the spotlight icon 🔍) and open the anaconda navigator:


Anaconda-Navigator
TOP HITS
Anaconda-Navigator — anaconda3
Anaconda-Navigator — anaconda-...
Anaconda-Navigator — anaconda-...
APPLICATIONS
Anaconda-Navigator — tensorflow_...
FOLDERS
anaconda-navigator-1.9.12-py36_0
anaconda-navigator-1.9.12-py37_0
DOCUMENTS
installation_guide.docx
anaconda-navigator-1.9.12-py36_0...
anaconda-navigator-1.9.12-py37_0....
entry_points.txt — anaconda_navig...
entry_points.txt — anaconda_navig...
IMAGES
Anaconda-Navigator
Version: $PKG_VERSION
Kind Application
Size 4.6 MB
Created 4/8/20
Modified 4/8/20
Last opened 4/8/20

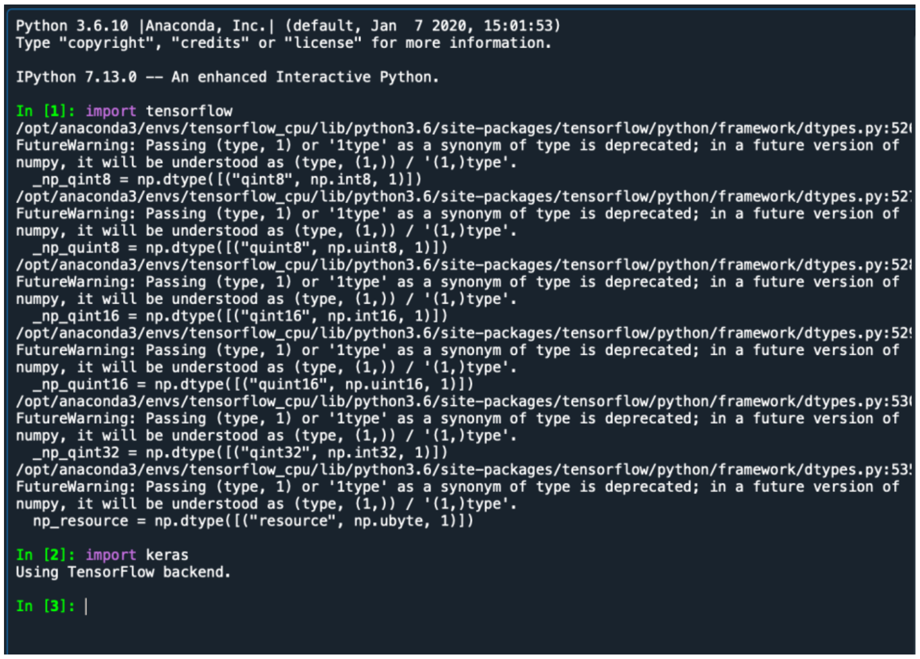
12. On the anaconda navigator, select the environment that we are working on:



13. Go to go to spider application, and press install. When the download is ready, launch it to test it.


Spyder
4.1.2
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Install
Spyder
4.1.2
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch

You should be able to import the installed libraries on the interactive console:



```
Python 3.6.10 |Anaconda, Inc.| (default, Jan  7 2020, 15:01:53)
Type "copyright", "credits" or "license" for more information.

IPython 7.13.0 -- An enhanced Interactive Python.

In [1]: import tensorflow
/opt/anaconda3/envs/tensorflow_cpu/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:52
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of
numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
/opt/anaconda3/envs/tensorflow_cpu/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:52
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of
numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
/opt/anaconda3/envs/tensorflow_cpu/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:52
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of
numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
/opt/anaconda3/envs/tensorflow_cpu/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:52
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of
numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
/opt/anaconda3/envs/tensorflow_cpu/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:53
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of
numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
/opt/anaconda3/envs/tensorflow_cpu/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:53
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of
numpy, it will be understood as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])

In [2]: import keras
Using TensorFlow backend.

In [3]: |
```

Step 2. Install the python API for VREP

To enable the python API for using python scripts with vrep we need to copy the following files into the lab 5 main folder:

- vrep.py
- vrepConst.py
- remoteApi.dll, remoteApi.dylib or remoteApi.so (depending on your operating system)

For all operating systems the files sim.py and simConst.py are located under *programming/remoteApiBindings/python*. Find these files and copy them in your lab5 folder (where all your python scripts are stored.) Then locate the remoteApi file according to your operating system and locate it in the same folder as all your python scripts.

For windows:

Copy the remoteApi.dll file into your lab5 folder, this file is located in the following path inside your v-rep installation folder:

C:\Program Files\V-REP3\V-REP_PRO_EDU\programming\remoteApiBindings\lib\lib\Windows\64Bit

Copy the files vrep.py and vrepConst.py into your lab 5 folder. These files are located in the following path inside your vrep installation folder:

C:\Program Files\V-REP3\V-REP_PRO_EDU\programming\remoteApiBindings\python\python

For Ubuntu:

Copy the remoteApi.so file located in the installation folder into your lab5 folder:

Path_to_vrep_installation\V-REP_PRO_EDU\programming\remoteApiBindings\lib\lib\Linux\64Bit\

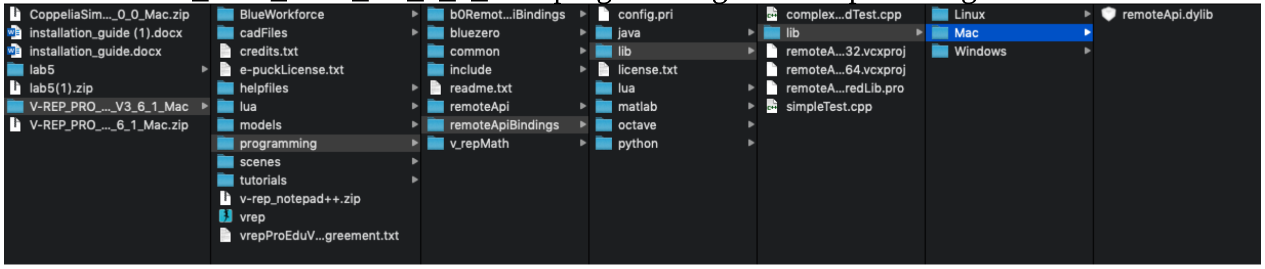
Copy the files vrep.py and vrepConst.py into your lab 5 folder. These files are located in the following path inside your vrep installation folder:

Path_to_vrep_installation \V-REP_PRO_EDU\programming\remoteApiBindings\python\python\

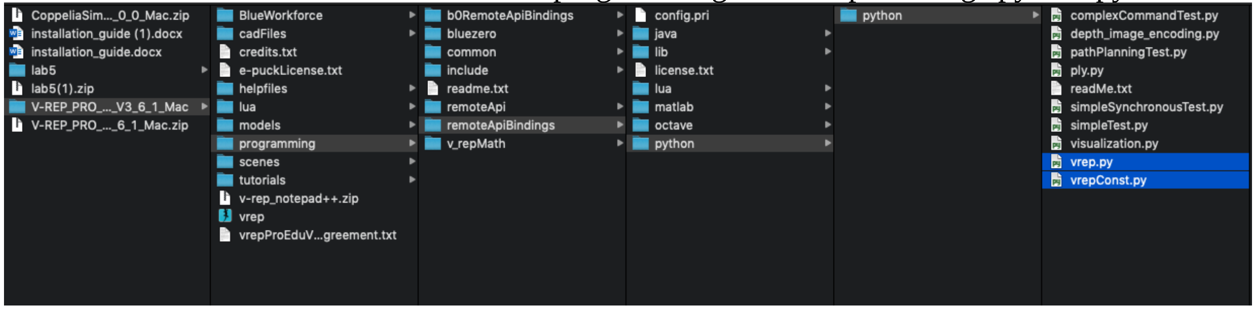
Change the port number in the RemoteApiConnections.txt to the port 19999, the file is found in the main v-rep installation folder. (see video tutorial).

For Mac:

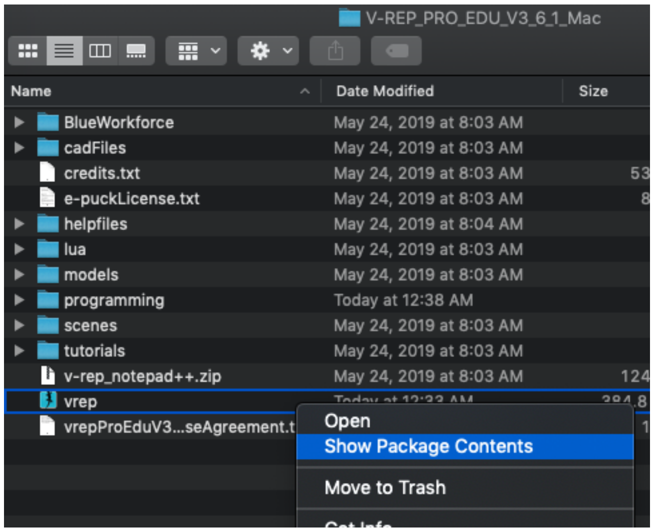
1. Copy the **remoteApi.dylib** file located in the installation folder into your lab5 folder. This file is located in your VREP folder in the following location:
   V-REP_PRO_EDU_V3_6_1_Mac/programming/remoteApiBindings/lib/lib/Mac/



2. Copy the files vrep.py and vrepConst.py into your lab 5 folder. These files are located in the following path inside your vrep installation folder:
   V-REP_PRO_EDU_V3_6_1_Mac/programming/remoteApiBindings/python/python/

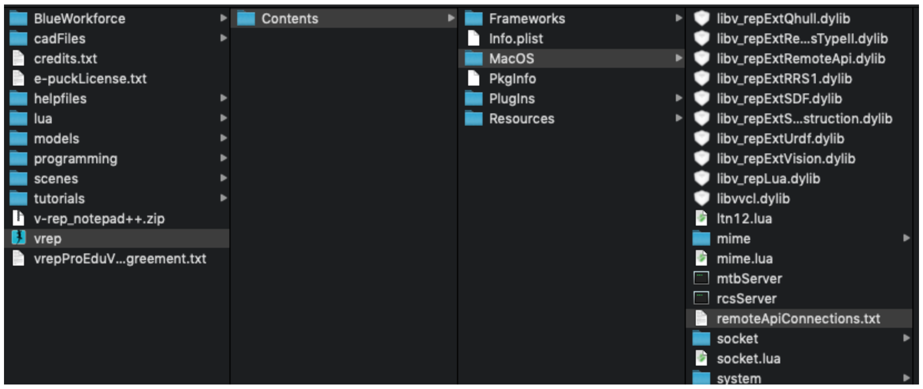


3. Change the port number in the RemoteApiConnections.txt to the port 19999, the file is found inside the vrep package contents:
   3.1. First open the V-rep package contents, by doing a right-click on the V-rep application:

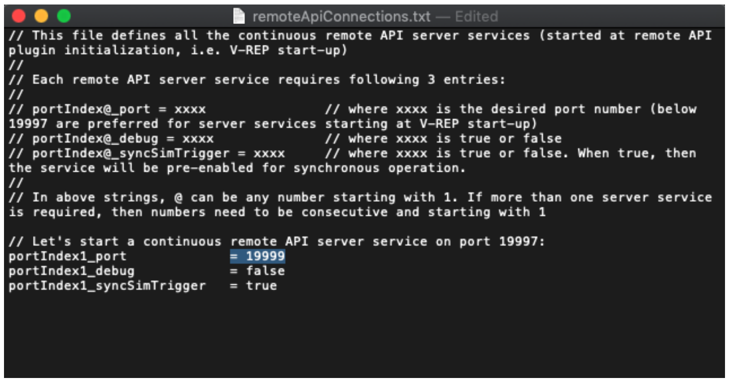


3.2 Open the RemoteApiConections.txt file, found in Contents/MacOS/

:



3.3 Modify the port to 19999 and save the file:



```
// This file defines all the continuous remote API server services (started at remote API
plugin initialization, i.e. V-REP start-up)
//
// Each remote API server service requires following 3 entries:
//
// portIndex@_port = xxxx                // where xxxx is the desired port number (below
19997 are preferred for server services starting at V-REP start-up)
// portIndex@_debug = xxxx               // where xxxx is true or false
// portIndex@_syncSimTrigger = xxxx      // where xxxx is true or false. When true, then
the service will be pre-enabled for synchronous operation.
//
// In above strings, @ can be any number starting with 1. If more than one server service
is required, then numbers need to be consecutive and starting with 1

// Let's start a continuous remote API server service on port 19997:
portIndex1_port              = 19999
portIndex1_debug             = false
portIndex1_syncSimTrigger    = true
```

Step 3. Run the python code along with your VREP scene

1. Once you follow the instructions for your OS, open the vrep scene for this laboratory lab5_2020_scene.ttt.

For Windows/Ubuntu:

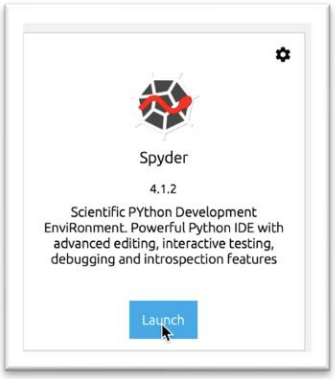
2. With the v-rep scene open, test the python API by running from the console (the console of your OS system)

```
$ python test_python_API.py
```

The previous command will run the python script test_python_API.py which should automatically start the simulation in vrep and print whether the python API connections was successful or not.

For Mac:

2. With the v-rep scene open, test the python API by running the test_python_API.py from Spider.
   2.1 First, open spider by following steps 11 and 12 of the MAC-OS installation guide.
   2.2 Then, launch Spyder:

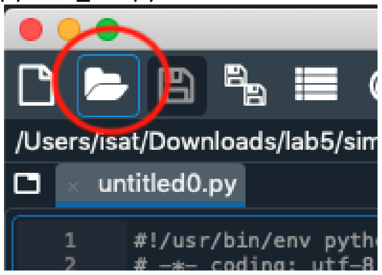


Spyder

4.1.2

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

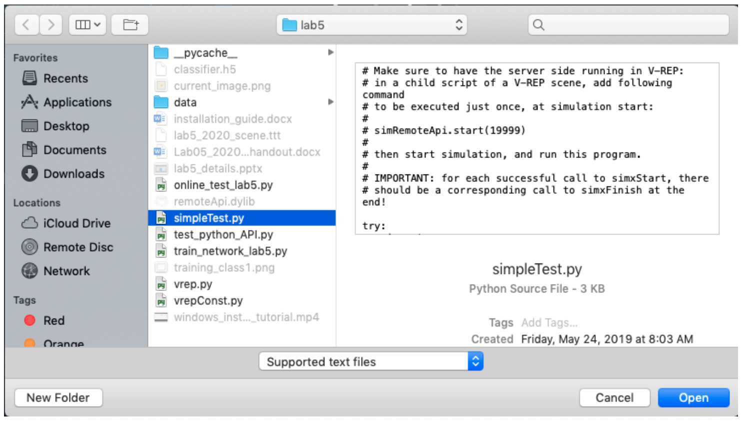
   2.3 In spider, open the file test_python_API.py



/Users/isat/Downloads/lab5/sim

untitled0.py

```
#!/usr/bin/env pyth
# -*- coding: utf-8
```



```
# Make sure to have the server side running in V-REP:
# in a child script of a V-REP scene, add following
command
# to be executed just once, at simulation start:
#
# simRemoteApi.start(19999)
#
# then start simulation, and run this program.
#
# IMPORTANT: for each successful call to simxStart, there
# should be a corresponding call to simxFinish at the
end!

try:
```

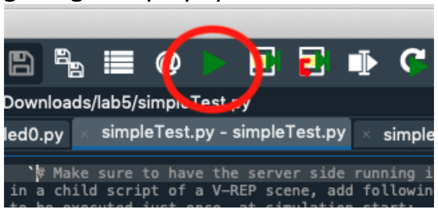
simpleTest.py

Python Source File - 3 KB

Tags Add Tags...

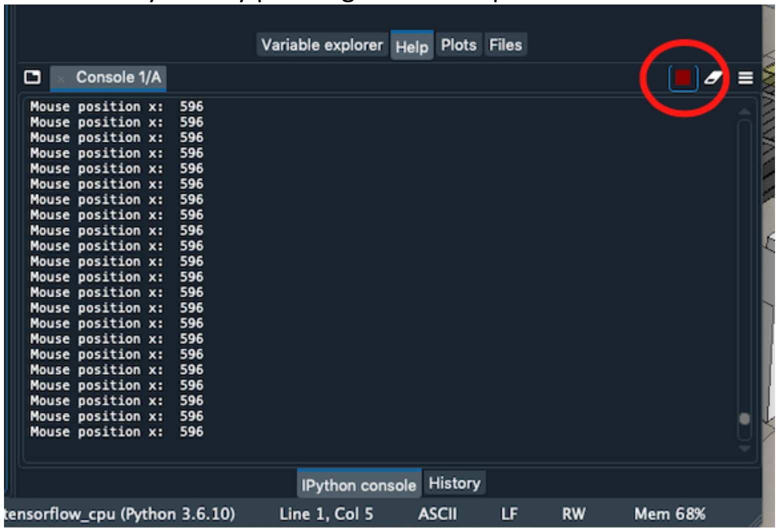
Created Friday, May 24, 2019 at 8:03 AM

2.4 Run the file by pressing the green play symbol



```
`# Make sure to have the server side running i
in a child script of a V-REP scene, add followin
```

2.5 Stop the execution anytime by pressing the red stop button



```
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
Mouse position x:  596
```

Further information about the vrep python API in :
https://www.coppeliarobotics.com/helpFiles/en/remoteApiClientSide.htm