Robotics Lab

Chand T. John First Bytes & Code Longhorn June 2019

1. Log into your account in the computer lab.

2. Click the 9 white dots at the bottom left corner of the screen.



3. Type terminal in the search box at the top.

Activities		Wed 11:20	•) () -
		Q terminal Q	
	· · · · ·		
	MATE Terminal Rxvt Color	Uni Xfce Terminal Terminal CNU Emacs 25	
	Ubuntu Software 10 more	💂 PuTTY Terminal Emulator Start a PuTTY terminal session	
		Terminal Command Line Interface	
		Sudoku console based sudoku game	
		Sakura Use the command line	
		robotfindskitten zen simulation of robot finding kitten	
			j

4. Click on the icon labeled Terminal.



5. In the window that pops up, type //lusr/bin/pycharm and press Enter.



6. Wait for the app called PyCharm Community Edition to launch.



You might see a bunch of WARNINGs here That's okay! Just ignore them.



7. Accept the license agreement if it asks you.

8. It doesn't matter whether you choose to send usage statistics or not. Just pick a choice. I prefer not to send usage statistics.

9. Choose + Create New Project.



10. A window will appear that allows you to select a folder where you'd like to store files for the robotics program you'll soon create.

	New Project	00
Location:	/u/chand/PycharmProjects/ <mark>untitled</mark>	5
▶ Project	Interpreter: New Virtualenv environment	
	Create	ancel

11. Choose a folder name, for example, MyRobotics, and click the Create at the bottom right.



12. Click the Close button on the Tip of the Day pop-up window.



13. Go to the File menu and choose Settings...

	MyRobotics [~/PycharmProjects/MyRobotics] - PyCharm	🖨 🖻 😣
<u>File Euro</u> Navigate <u>C</u> ode <u>R</u> efactor R <u>u</u> n	<u>I</u> ools VC <u>S</u> <u>W</u> indow <u>H</u> elp	
MyRobotics		Add Configuration 🕨 🔅 🔳 🔍
Project ▼ Project ▼ ■ MyRobotics -//PycharmProjects//MyRobotic ⇒ ■ Me External Libraries Scratches and Consoles 		
	Search Everywhere Double Shift	
	Navigation Bar Alt+Home	
	- Dron files bere to open	
rites		
Favo		
★		
e		
₩ 2: struct		A External file Project files c network mount?)
Python Console B Terminal ≡ 6: TODO		1 Event Log
External file changes sync may be slow: Project file	les cannot be watched (are they under network mount?) (a minute ago)	2 ±

14. Click the little triangle to the left of Project: MyRobotics (or whatever you named your project)

	Settings	8
Q.	Appearance & Behavior > Appearance	
 Appearance & Behavior Appearance Menus and Toolhars 	Theme: Darcula	
► System Settings File Colors Scopes Notifications	Accessibility Support screen readers (requires restart) Adjust colors for red-green vision deficiency (protanopia, deuteranopia) How it works	
Quick Lists Keymap Editor Plugins Version Control Pioject: MyRobotics	UI Options Background Image Cyclic scrolling in list Show icons in quick navigation Automatically position mouse cursor on default button Hide navigation popups on focus loss Drag-p-Drop with AI Toressed only	
 ► Languages & Frameworks ► Tools 	Tooltip initial delay (ms):	
?	Animate windows ✓ Show tool window bars Show memory indicator ✓ Show tool window numbers Disable mnemonics in menu ✓ Allow merging buttons on dialogs OK Cancel Ap	

15. In the expanded menu, choose Project Interpreter.

	Set	tings	8
Q.	Project: MyRobotics > Project In	t erpreter 🖷 For current proje	
• Appearance & Behavior Appearance Menus and Toolbars • System Settings File Colors Image: Second Secon	Project: MyRobotics > Project Int Project Interpreter: Python 3.6 Package pip setuptools	terpreter ® For current proje (MyRobotics) -/PycharmProjects/МуRot Version 19.0.3 40.8.0	ect botics/venv/bin/python Latest version 19.1.1 41.0.1
Tools			OK Cancel Apply

16. Click the + sign at the top right.

	Set	tings		8
	Project: MyRobotics > Project Int	erpreter @ For current proje		
 Appearance & Behavior Appearance Menus and Toolbars System Settings 	Project Interpreter: 🔹 Python 3.6 (Package Pip setuntools	MyRobotics)/PycharmProjects/MyRob Version 19.0.3 40.8 0	Latest version ▲ 19.1.1	+
File Colors © Scopes © Notifications Quick Lists Keymap ▶ Editor Plugins ▶ Version Control © ♥ Project: MyRobotics ©				0
 Project Interpreter Project Structure Build, Execution, Deployment Languages & Frameworks Tools 				
			OK Cancel App	ly

17. Type robopy into the search box.



18. Select robopy in the list if it isn't highlighted



19. Click Install Package.



20. A notification should pop up in the lower right once the robopy package has been installed.



21. Now type numpy into the search box.

	Available Packages 🧧
Q-numpy	
numpy	S Description
numpy-cloud	NumBy is the fundamental package for array computing with Dither
numpy-groupies	Numey is the fundamental package for array computing with Python.
numpy-html	Version
numpy-image-widget	1.16.4
numpy-indexed	Author
numpy-mkl	Travis E. Oliphant et al.
numpy-partition	
numpy-posit	
numpy-quaternion	
numpy-stl	
numpy-sugar	
numpy-turtle	
numpy-utils	
numpy2gif	
numpy_display	
numpy_ringbuffer	
numpycnn	
numpydoc	
numpyencoder	
numpyro	
numpysane	
numpyson	
numpythia	
numpytimebuilder	Sperifyversion 1164
numpyx	
pnumpy	Options
Package 'robopy' installed successfully	
Install Package <u>M</u> anage Repositor	

22. Select numpy if it isn't highlighted already.

	Av	ailable Packages	8
Q-numpy			
numpy	G	Description	
numpy-cloud			
numpy-groupies		NumPy is the fundamental package for array computing with Python.	
numpy-html		Version	
numpy-image-widget		1.16.4	
numpy-indexed		Author	
numpy-mkl		Travis E. Oliphant et al.	
numpy-partition			
numpy-posit			
numpy-quaternion			
numpy-stl			
numpy-sugar			
numpy-turtle			
numpy-utils			
numpy2gif			
numpy_display			
numpy_ringbuffer			
numpycnn			
numpydoc			
numpyencoder			
питруго			
numpyson			
numpythia			
numpytimebuilder		Specify version 1.16.4	
pnumpy		Options	
Package 'robopy' installed successfully			
Install Package Manage Repositories			

23. Click Install Package.

	Av	ailable Packages	8
Q- numpy			
	G	Description	
numpy-cloud			
numpy-groupies		NumPy is the fundamental package for array computing with Python.	
numpy-html		Version	
numpy-image-widget		1.16.4	
numpy-indexed		Author	
numpy-mkl		Travis E. Oliphant et al.	
numpy-partition			
numpy-posit			
numpy-quaternion			
numpy-stl			
numpy-sugar			
numpy-turtle			
numpy-utils			
numpy2gif			
numpy_display			
numpy_ringbuffer			
numpycnn			
numpydoc			
numpyencoder			
питруго			
numpyson			
numpythia			
numpytimebuilder		Specify version 1.16.4	
pnumpy			
Package 'robopy' installed successfully			
Install Package Manage Repositories			

24. A notification should pop up indicating that the numpy package was installed successfully.



25. Type vtk into the search box.

	Available Packages
Q-vtk	
PyVTK avtk itk-vtkglue ncvtk omfvtk pvtk pysimplevtk scikit-surgeryvtk swc2vtk vext.vtk Vtk vtk-vacreader vtk_visualizer vtke vtki vtki vtklotter	Description VTK is an open-source toolkit for 3D computer graphics, image processing, and visualization Version 8.1.2 Author VTK Community mailto:vtk-developers@vtk.org https://vtk.org/
	Specify version 8.1.2
	Options
Package 'numpy' installed successfully	
r denage manpy instance successionly	

26. Select vtk if it isn't highlighted already.



27. Click Install Package.

Q-vtk		
РуVТК	S	Description
avtk		VTK is an other source teelly the approximation of the
itk-vtkglue		processing, and visualization
ncvtk		Version
omrvtk		8.1.2
pvetk		Author
pycyck		VTK Community
scikit-surgerwtk		
swc2vtk		
vext.vtk		
vtk-vacreader		
vtk_visualizer		
vtki		
vtkplotter		
		Specify version 8.1.2
		Options
Package 'numpy' installed successfully		
Install Packago Manago Poposito		

28. A notification should pop up indicating that the vtk package was installed successfully.



29. Close the Available Packages window.



30. Click OK to close the Settings window.

Settings					
	Project: MyRobotics > Pr	oject Interpreter 🛛 🖻 For curren			
▼ Appearance & Behavior Appearance Menus and Toolbars	Project Interpreter: Rept	hon 3.6 (MyRobotics) -/PycharmProjects	s/MyRobotics/venv/bin/python Latest version	▼	
 > System Settings File Colors Scopes Notifications Quick Lists Keymap Editor Plugins Version Control Project: MyRobotics Project Interpreter Project Structure Build, Execution, Deploymen Languages & Frameworks Tools 	 Pillow imageio numpy pip robopy scipy setuptools vtk 	2.5.0 2.1.16.4 19.0.3 1.0.8 1.3.0 40.8.0 8.1.2	5.00 2.5.0 1.16.4 ▲ 19.1.1 1.0.8 1.3.0 ▲ 41.0.1 8.1.2	•	
	Package 'vtk' installed succ	cessfully	OK Cancel	Apply	

31. Right-click the name of your project under the Project panel.



32. Hover over "New" and choose "Python file". A little window will appear.



33. Type Main and click OK.



34. Type this code into the Main.py window.

MyRobotics [~/PycharmProjects/MyRobotics]/Main.py [MyRobotics] - PyCharm 😑 💿 🙍		
Eile Edit View Navigate Code Refactor Run Tools VCS Window Help		
🖿 MyRobotics 👌 💑 Main.py		Add Configuration) 🕨 👜 📕 🛛 🗛
ਜ਼ੂ 🔲 Project 🗸 😌 😤 🗢 —	💑 Main.py 👋	
MyRobotics ~/PycharmProjects/MyRobotics	1 Dimport robopy.base.model as model 2 Dimport numpy as np	
 Mykobotics "/PyknatmiPtojects/Mykobotics Main.py Illi External Libraries Scratches and Consoles 	<pre>import roboy.base.model as model import numpy as np import numpy</pre>	<pre>Main.py × Main.py × Main.py × Main.py × mort robopy.base.model as model def main(): robot = model.Puma560() a = np.transpose(np.asmatrix(np.linspace(1, -180, 500))) b = np.transpose(np.asmatrix(np.linspace(1, 180, 500))) c = np.transpose(np.asmatrix(np.linspace(1, 180, 500))) c = np.transpose(np.asmatrix(np.linspace(1, 90, 500))) d = np.transpose(np.asmatrix(np.linspace(1, 450, 500))) e = np.asmatrix(np.zeros((500, 1))) f = np.concatenate((d, b, a, e, c, d), axis=1) </pre>
. 1. 2. Strutture ★ 2. Favorites		<pre>12</pre>
🕏 Python Console 🛛 Terminal 🔠 <u>6</u> : TODO		2 Event Log

Make sure you typed in the program EXACTLY as shown. If the indentation isn't the same, or if even a single punctuation mark is out of place or missing, the program probably won't work!

Computers are really, really picky. They do EXACTLY what you say, and they don't understand what you really meant for them to do.

If something doesn't work in the next step, it's probably because there is a typo in your program somewhere. Don't worry, it happens to all of us. We're humans, not robots. We humans make mistakes, but we're also creative, and that's all part of what makes humankind fascinating.
35. It's time to run your first robotics program! In the Run menu, choose Run...



36. In the little window that pops up, choose Main and Run...

37. Watch the robot move as your program runs!



38. When the robot stops moving (or before that, if you're in a hurry), close the window to stop the program.



39. Run your program again, but this time a faster way: click the Play button at the

top right.

From now on, you can run your program this way, or by typing SHIFT+F10.



40. Click and drag the corners of the robot window to make it bigger.



41. Right-click and drag the mouse pointer inside the robot window to zoom in and out.



42. Click with the middle mouse button (or SHIFT+left-click) and drag the mouse pointer inside the robot window to slide the robot around.



43. Click and drag the mouse around inside the robot window to rotate the robot in 3D.



44. In the Project panel, click the little triangles to expand venv, then lib, then python3.6, then site-packages, then robopy,

and finally, base. Double-click model.py to open it.

😣 🗇 🗊 PyCharm [~/PyCharm]/venv/lib/pytho	n3.6/site-packages/robopy/base/model.py - PyCharm	
🖿 PyCharm 👌 🖿 venv 🤇 🖿 lib 👌 🖿 python 3.6 🖉 🖿 site-	packages) 🖿 robopy) 🖿 base) 🐇 model.py 🛛 👘 🧼 Main	🔹 🕨 🛎 🔳 🔍
ਦੂ 🗐 Project 👻 😳 😤 🗢 —	缘 Main.py × 编 model.py ×	
💆 🔻 🖿 PyCharm ~/PyCharm	1 Created by: Aditya Dua	
🛱 🔻 🖿 venv	2 白海動 uctober 2017 3 白油port	
bin		
🖿 include	112 class Puma560(SerialLink):	
🔻 🐂 lib	13 definit(self, base=None):	
🔻 🖿 python3.6	14 15 self.gn = np.matrix([[0, pi / 4, pi, 0, pi / 4, 0]])	
🔻 🖿 site-packages library root	16 self.qr = np.matrix([[0, pi / 2, -pi / 2, 0, 0, 0]])	
🕨 🖿 imageio	1/ SetT.qz = np.matrix1(10, 0, 0, 0, 0, 0, 0)]) 18 setf.gs = np.matrix1(10, 0, -0i, 72, 0, 0, 0)])	
🕨 🖿 imageio-2.5.0.dist-info	19 self.scale = 1	
🕨 🖿 numpy	20 □ param = { "cube axes x bounds": no.matrix([[-1.5, 1.5]]).	
numpy-1.16.3.dist-info	22 "cube_axes_y_bounds": np.matrix([[-0.7, 1.5]]),	
PIL	23 "cube_axes_z_bounds": np.matrix([1-1.5, 1.5]]), 24 "floor position": np.matrix([100.7, 0]])	
▶ ■ Pillow-6.0.0.dist-info		
pip-19.0.3-py3.6.egg	26	
🔻 🖿 горору	28 Revolute(d=0, a=0, 4318, alpha=0, j=0, theta=0, offset=0, qli=(-45 * pi / 180, 225 * pi / 180)),	
▼ ■ base	29 revolute(two.isoro, amo.ozos, atpia=-pi / 2, j=o, thetamo, offsetmo, 30 qlim=(-225 * pi / 180, 45 * pi / 180),	
initpy	31 Revolute(d=0.4318, a=0, alpha=pi / 2, j=0, theta=0, offset=0, qlim=(-110* pi / 180, 170* pi / 180) Revolute(d=0.4318, a=0, alpha=pi / 2, j=0, theta=0, offset=0, qlim=(-110* pi / 180, 170* pi / 180)	
Check_args.py	32 nevolute(i=0, a=0, atpha=0, j=0, theta=0, offset=0, qtim=(-100 + pi / 100, 100 + pi / 100), 33 ∩ Revolute(i=0, a=0, atpha=0, j=0, theta=0, offset=0, qtim=(-120 + pi / 180, 226 + pi / 180)]	
Common.py	34	
to graphics.py	35 li dase 15 Mone: 36 base 1 triotx(-90, unit='deg')	
f noce py	37 else:	
austernion ov	39 file_names = SerialLinksetup file_names(7)	
serial link ov	<pre>48 colors = graphics.vtk_named_colors(["HotPink", "Orchid", "PaleVioletRed", "LightPink", "Magenta", "DeepPink",</pre>	
Super pose pv	super()init(links=links, base=base, name='puma_560', stl_files=file_names, colors=colors, param=param)	
transforms.pv		
🔏 util.ov		
► ■ media	46	
► 🖿 tests		
🛃initpy	49 sett.scale = 0.01 58 = # Pre-defined Stances	
e 🕨 k 🖿 robopy-1.0.8.dist-info		
रु 🕨 🖿 scipy		
. 🖌 🕨 scipy-1.2.1.dist-info	54 🕴 links = [Revolute(d=0, a=0, alpha=0, j=0, theta=0, offset=0, qlim=0), # Turret	
È ► ► vtk	55 Revolute(i=0.53, a=-0.30399, atpha=0, j=0, theta=0, offset=0, qtim=0), # Snouter 55 Revolute(i=0, a=-1.70384, alpha=0, i=0, theta=0, offset=0, qtim=0), # Elbow	
vtk-8.1.2.dist-info	57 Revolute(d=0, a=-1.36307, alpha=0, j=0, theta=0, offset=0, glim=0), # Wrist	
រ្វ័ 🖞 easy-install.pth	so Revolute(d=0, a=0, atpna=0, j=0, theta=0, offset=0, qtim=0), 59 ⊖ Revolute(d=0, a=0, alpha=0, j=0, theta=0, offset=0, qlim=0)]	
setuptools.pth	60 if here is Norma	
🖉 🕨 🕨 setuptools-40.8.0-py3.6.egg	62 base = tr.trotx(-90, unit='deg')	
k vtkVersion.py		
► ■ lih64 E 6: TODO Terminal Python Console		
IDE and Plugin Updates: The following olugins are	ready to update: BashSuppor(6 minutes ago)	3 ÷ 4 spaces÷ 0 = 1
in the strain operates. The rottoming plugins are		in the second se

45. Examine the code in this file that lists the colors of the robot's links.



46. There are lots of other colors we could use.

https://vtk.org/Wiki/VTK/Examples/Python/Visualization/NamedColorPatches

'Red':['IndianRed', 'LightCoral', 'Salmon', 'DarkSalmon',\ 'LightSalmon', 'Red', 'Crimson', 'FireBrick', 'DarkRed'], 'Pink':['Pink', 'LightPink', 'HotPink', 'DeepPink',\ 'MediumVioletRed', 'PaleVioletRed'], 'Orange':['LightSalmon', 'Coral', 'Tomato', 'OrangeRed'.\ 'DarkOrange', 'Orange'], 'Yellow':['Gold', 'Yellow', 'LightYellow', 'LemonChiffon',\ 'LightGoldenrodYellow', 'PapayaWhip', 'Moccasin',\ 'PeachPuff', 'PaleGoldenrod', 'Khaki', 'Darkkhaki'], 'Purple':['Lavender', 'Thistle', 'Plum', 'Violet', 'Orchid',\ 'Fuchsia', 'Magenta', 'MediumOrchid', 'MediumPurple',\ 'BlueViolet', 'DarkViolet', 'DarkOrchid', 'DarkMagenta', 'Purple', 'Indigo', 'DarkSlateBlue', 'SlateBlue', 'MediumSlateBlue'l. 'Green':['GreenYellow', 'Chartreuse', 'LawnGreen', 'Lime',\ 'LimeGreen', 'PaleGreen', 'LightGreen', 'MediumSpringGreen', 'SpringGreen', 'MediumSeaGreen', 'SeaGreen', 'ForestGreen', 'Green', 'DarkGreen',\ 'YellowGreen', 'OliveDrab', 'Olive', 'DarkOliveGreen', 'MediumAguamarine', 'DarkSeaGreen', 'LightSeaGreen', \ 'DarkCyan', 'Teal'], 'Blue/Cyan':['Aqua', 'Cyan', 'LightCyan', 'PaleTurquoise',\ 'Aquamarine', 'Turquoise', 'MediumTurquoise',\ 'DarkTurquoise', 'CadetBlue', 'SteelBlue', \ 'LightSteelBlue', 'PowderBlue', 'LightBlue',\ 'SkyBlue', 'LightSkyBlue', 'DeepSkyBlue', \ 'DodgerBlue', 'CornflowerBlue', 'RoyalBlue', 'Blue', \ 'MediumBlue', 'DarkBlue', 'Navy', 'MidnightBlue'], 'Brown':['Cornsilk', 'BlanchedAlmond', 'Bisque', 'NavajoWhite',\ 'Wheat', 'BurlyWood', 'Tan', 'RosyBrown', 'SandyBrown', \ 'Goldenrod', 'DarkGoldenrod', 'Peru', 'Chocolate', \ 'SaddleBrown', 'Sienna', 'Brown', 'Maroon'], 'Beige', 'OldLace', 'FloralWhite', 'Ivory',\ 'AntiqueWhite', 'Linen',\ 'LavenderBlush', 'MistyRose'], 'Gray' ['Gainsboro', 'LightGrey', 'Silver', 'DarkGray', 'Gray', 'DimGray', 'LightSlateGray', 'SlateGray', 'DarkSlateGray', \ 'Black']

47. So, change the colors to ones you like! For example:

nite 11

Again, remember to keep all the quotes and other punctuation as they were. If any punctuation is missing or out of place, the program might not run anymore! The computer just isn't intelligent enough to know what you mean unless you type in everything just right!

48. Then run your program again...



(or SHIFT+F10)

49. And enjoy the fancy new colors of your robot!



50. You may now close model.py if you'd like.



51. Now, double-click serial_link.py to open it.

<pre>Project - Colors - Colors</pre>		ychons.o / == sice-package	s /= robopy /= base / te senar_mix.py		
<pre>PyCham _/byCham _/byCham</pre>	Project 👻	😌 🛨 🏛 — 🚜 Mair	n.py × K serial link.py ×		
<pre>vervev * Sum * Toduce * Supphon3.6 * Supphon3.6</pre>	PyCharm ~/PyCharm		.# Created by: Anitya Uua # 30 Sectember 2017		
<pre>bind the decises (binaryroot imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo imagelo</pre>	venv				
<pre>include image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image: image:</pre>					
<pre>**** * Dython3.6 * Dython</pre>	🖿 include				
<pre>*#python36 * With the part of the set o</pre>					
<pre> # integrades libraryroot * imageing * imageing</pre>			SeralLink object class.		
<pre>imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio imageio</pre>	🔻 🖿 site-packages 🛛				
<pre> Imagebc_25.0dist-info Imagebc_</pre>	🕨 🖿 imageio		def		
<pre>image: image: imag</pre>	imageio-2.5.				
<pre>> Innumpy1.16.3.distinfo > Innumpy1.16.3.</pre>	Image: Second				
<pre>> PPL > PPL > PPL > PPL > PPL > PPLO.3.PyS.6.egg > Probaby > Probaby ></pre>	numpy-1,16.	dist-info	:param name: name property of the object. :param hase: hase transform annlied to the Seriallink object.		
<pre>> Pillow-6.0.dist-info > pip-19.0.3-py3.6-egg > Tobopy > Tobopy > Tobopy > Tobopy > Tobopy > Tobopy > Subset = Subs</pre>	► E PIL		:param stl files: STL file names to associate with links. Only works for pre-implemented models in model module.		
<pre>pip-19.0.3-py3.6.egg v = robopy v = base #</pre>	Pillow-6.0.0.	ist-info	param q: initial angles for link joints.		
<pre>int the probability is provided in the probability in the probability is provided in the probability in the provided in</pre>	▶ ■ pin-19 0 3-p	6 eng 31			
<pre>identify the set of the set</pre>	v i robopy		self.pipeline = None		
<pre>image: image: imag</pre>	v = robopy		setT.linkS = linkS if n is None:		
<pre>is the set is there:</pre>	V Dase		<pre>self.q = np.matrix([0 for each in links])</pre>		
<pre>is common.py</pre>	CoIIIL_	Py 37	if base is None:		
<pre>scription (spr) set tope (spr) (spr)</pre>	Check_	gs.py 39	else:		
<pre>\$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$</pre>	io commo	••РУ 40			
<pre>% model.py % post_rool = np.asmatrix(np.eye(4, 4)) % post_rool = np.asmatrix(np.eye(4, 4)) % post_rool = not % post</pre>	i graphi	Py 41 47	self.base = base if tool is None:		
<pre>\$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$</pre>	🔏 model.				
<pre>% guternion.py % scale (https://withousernood) % super_pose.py % transforms.py % transfo</pre>	🖧 pose.pj		else:		
<pre>% super_pose py % stransforms.py % bit fast [files is None:</pre>	🕻 quater	ion.py	asser (typertoor) is inclustration and (toor, shape (+, +))		
if st[:riles is None: if without if without <td>🐇 serial_l</td> <td>ik.py 47</td> <td></td> <td></td> <td></td>	🐇 serial_l	ik.py 47			
<pre>% transforms.py</pre>	👸 super_		If st_files 15 None: # Default stick figure model code goes here		
<pre>\$ @ utiloy \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$</pre>	🖧 transfo				
<pre>> Immedia == strikt("Iter strictiles strictiles = st</pre>	🖧 util.py				
<pre>> Intests ssl1.name = ''</pre>			if name is None:		
<pre></pre>	► 🖿 tests				
<pre>> Im robopy-1.0.8.dist-info > If colors is Now: solit.colors = vtk_named_colors(("Grey"] + len(stt_files)) disciput_2.1.distinfo > Im vtk > Im vtk_1.2.distinfo > Im vtk_2.2.distinfo = disciput_2.2.distinfo = d</pre>	👸 init .o		etse: self_name = name		
<pre>scipy = scipy = scip.colors = vtk_named_colors(['Grey'] * len(stl_files)) else: scipy=1.2.1.dist-info scipy=1.2.1.dist-info wtk=n12.dist-info dia easy-install.pth dia easy-install.pth dis sciptools.pth li setuptools.eth with = sciptools.eth dia estuptools.eth dia estuptools.e</pre>	robopy-1.0.8	dist-info 57	if colors is None:		
<pre>> ■ scipy-1.2.1.distinfo > ■ vtk > ■ vtk-8.12.distinfo G = gayinstall.pth G = gayin</pre>	Scipy		<pre>self.colors = vtk_named_colors(["Grey"] * len(stl_files)) </pre>		
	scipy-1.2.1.d				
▶ Wtk-8.1.2.dist-info 63 ↓ If model destit t pass params, t hen use these default ones ▲ easy-install.pth 63 ↓ self.param = { ▲ easy-install.pth 65 ↓ cube_axesbounds*: np.matrixt[[1-1.5, 1.5]]), ▲ setuptools.pth 66 ↓ cube_axesbounds*: np.matrixt[[1-1.5, 1.5]]), ▲ ■ cube_axesbounds*: np.matrixt[[1-1.5, 1.5]]), * ★ Utbe_axesbounds*: np.matrixt[[1-1.5, 1.5]]), * ★ Utbe_axesbounds*: np.matrixt[[1-1.5, 1.5]]), * ★ Utbe_axesbounds*: np.matrixt[[1.1.5, 1.5]]), * ★ Utbe_axesbounds*: np.matrixt[[1.1.5, 1.5]]), * ★ Utbe_axesbounds*: np.matrixt[[1.1.5, 1.5]]), * ★ * * * ★ * * * ★ * * * * ★ * * * * ★ * * * * ★ * * * * ★ * * * *	I = Jeipy 1.2.1.0		if param is None:		
Control and the set of the	Vtk-8.4.2 diel	info 62	# If model deosn't pass params, then use these default ones		
Casy instant p(1 = 5, 1.5]), Constant p(1 = 6, 1.5, 1.5]), Constant p(1 = 5, 1.5), Constantp(1 = 5, 1.5), Constant p(1 =	d openingtally	-b 64	"cube_axes_x_bounds": np.matrix([[-1.5, 1.5]]),		
Listerprotos.put Code area ≥ bolines : np.matrix([[1,3, 1:3])). Image: transmission in the setup tool = 0.000 area ≥ bolines : np.matrix([[0, -1.5, 0]]) Image: transmission in the setup tool = 0.000 area ≥ bolines : np.matrix([[0, -1.5, 0]]) Image: transmission in the setup tool = 0.000 area ≥ bolines : np.matrix([[0, -1.5, 0]]) Image: transmission in the setup tool = 0.000 area ≥ bolines : np.matrix([[0, -1.5, 0]]) Image: transmission in the setup tool = 0.000 area ≥ bolines : np.matrix([[0, -1.5, 0]]) Image: transmission in the setup tool = 0.000 area ≥ bolines : np.matrix([[0, -1.5, 0]]) Image: transmission in the setup tool = 0.000 area ≥ bolines : np.matrix([[0, -1.5, 0]]) Image: transmission in the setup tool = 0.000 area ≥ bolines : np.matrix([[0, -1.5, 0]]) Image: transmission in the setup tool = 0.000 area ≥ bolines : np.matrix([[0, -1.5, 0]]) Image: transmission in the setup tool = 0.000 area ≥ bolines : np.matrix([[0, -1.5, 0]]) Image: transmission in the setup tool = 0.000 area ≥ bolines : np.matrix([[0, -1.5, 0]]) Image: transmission in the setup tool = 0.000 area > 0.000 area = 0.0000 area = 0.00000 area = 0.0000 area = 0.0000 area = 0.000	a easy-install.		"cube_axes_y_bounds": np.matrix([[-1.5, 1.5]]),		
	a secuptools.p		"floor position": np.matrix([[0, -1.5, 0]])		
serialLink ⇒ animate() > execute() SerialLink ⇒ animate() > execute() (6 TODO	El secuptools-4 # utbac	. 8.0-py3.6.egg			
E CODO DE Terminal de Pethan Console	NCKVersion.p		SerialLink > animate() > execute()		
	6: TODO 🛛 Terminal 📥 P	thon Console		O FM	

52. Scroll down to where it says def animate(...):

		<pre>def animate(self, stances, unit='rad', frame_rate=25, gif=None): """</pre>	
		Animates SerialLink object over nx6 dimensional input matrix, with each row representing list of 6 joint angles. :param stances: nx6 dimensional input matrix. :param unit: unit of input angles. Allowed values: 'rad' or 'deg' :param frame_rate: frame_rate for animation. Could be any integer more than 1. Higher value runs through stances for :return : null	aster
		<pre>if unit == 'deg': stances = stances * (pi / 180)</pre>	
		<pre>self.pipeline = VtkPipeline(total_time_steps=stances.shape[0] - 1, gif_file=gif) self.pipeline.reader_list, self.pipeline.actor_list, self.pipeline.mapper_list = selfsetup_pipeline_objs() self.fkine(stances, apply_stance=True, actor_list=self.pipeline.actor_list) self.pipeline.add_actor(axesCube(self.pipeline.ren))</pre>	
		<pre>def execute(obj, event): nonlocal stances self.pipeline.timer_tick()</pre>	
		<pre>self.fkine(stances, apply_stance=True, actor_list=self.pipeline.actor_list, timer=self.pipeline.timer_count) self.pipeline.iren.GetRenderWindow().Render()</pre>	
		<pre>self.pipeline.iren.AddObserver('TimerEvent', execute) self.pipeline.animate()</pre>	
205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 220	D Q Q	<pre>self.pipeline = VtkPipeline(total_time_steps=stances.shape[0] - 1, gif_file=gif) self.pipeline.reader_list; self.pipeline.actor_list; self.pipeline.mapper_list = selfsetup_pipeline_objs() self.fkine(stances, apply_stance=True, actor_list=self.pipeline.actor_list) self.pipeline.add_actor(axesCube(self.pipeline.ren)) def execute(obj, event): nonlocal stances self.pipeline.timer_tick() self.fkine(stances, apply_stance=True, actor_list=self.pipeline.actor_list, timer=self.pipeline.timer_cou self.pipeline.iren_obj self.pipeline.iren.GetRenderWindow().Render() self.pipeline.iren.AddObserver('TimerEvent', execute) self.pipeline.animate()</pre>	int)

53. To get the program to start with a better view of the robot, add these lines of code:

Between the

lines

```
def animate(self, stances, unit='rad', frame rate=25, gif=None):
                                                      :param stances: nx6 dimensional input matrix.
                                                      :param unit: unit of input angles. Allowed values: 'rad' or 'deg'
                                                      :param frame rate: frame rate for animation. Could be any integer more than 1. Higher value runs through stances faster.
                                                      if unit == 'dea':
                                                      self.pipeline = VtkPipeline(total time steps=stances.shape[0] - 1, gif file=gif)
                                                      self.pipeline.reader list, self.pipeline.actor list, self.pipeline.mapper list = self. setup pipeline objs()
                                                      self.fkine(stances, apply stance=True, actor list=self.pipeline.actor list)
                                                      self.pipeline.add actor(axesCube(self.pipeline.ren))
                                                      def execute(obj, event):
                                                          nonlocal stances
                                                          self.pipeline.timer tick()
                                                          self.fkine(stances, apply stance=True, actor list=self.pipeline.actor list, timer=self.pipeline.timer count)
                                                          self.pipeline.iren = obj
                                                          self.pipeline.iren.GetRenderWindow().Render()
                                                      self.pipeline.iren.AddObserver('TimerEvent', execute)
                                                      renderer = self.pipeline.iren.FindPokedRenderer(0, 0)
                                                      self.pipeline.iren.FlyTo(renderer, 0, -1, 6.9)
                                                      self.pipeline.iren.GetRenderWindow().SetSize(800, 800)
                                                      camera = renderer.GetActiveCamera()
AddObserver
                                                      transform = camera.GetModelViewTransformObject()
                                                      transform.Identity()
and animate()
                                                      transform.RotateX(10)
                                                      transform.RotateY(-20)
                                                      transform.Translate(0.19981, -1.53877e-13, -3.46838)
                                                      camera.ApplyTransform(transform)
                                                      self.pipeline.animate()
```

54. Run the program again.



😕 🗇 🔍 PyCharm [~/PyCharm]/venv/lib/pyth	on3.6/site-packages/robopy/base/serial_link.py - PyCharm	
🖿 PyCharm 👌 🖿 venv 👌 🖿 lib 👌 🖿 python 3.6 👌 🖿 site	-packages 🕽 🖿 robopy 🤉 🖿 base 👌 🚜 serial_link.py	📦 Main 👻 📐 🗯 📒 🔍
👷 🗐 Project 😽 😌 🛨 🗘 —	🖧 Main.py 🛛 🚜 serial_link.py 🖄	
🖉 🔻 🖿 PyCharm ~/PyCharm	183 actor_list[i].GetProperty().SetColor(self.colors[i]) # (R,G,B)	
🛱 🔻 🖿 venv	184 195 conturn reader list actor list manner list	
▶ 🖿 bin	186	
🖿 include	187 @staticmethod	
🔻 🖿 lib	189 file_names = []	
🔻 🖿 python3.6	190 for i in range(0, num): file names annend('link' + str(i) + ' stl')	
site-packages library root		
Imageio	193 A return file_names	
Imageio-2.5.0.dist-info	<pre>195</pre>	
▶ ■ numpy	196 😝 """ 197 Animates Seciallink object over ny6 dimensional input matrix with each row represen	
Image:	198 :param stances: nx6 dimensional input matrix.	
	199 :param unit: unit of input angles. Allowed values: 'rad' or 'deg' 200 :param frame rate: frame rate for animation. Could be any integer more than 1. Higher	
	201 :return: null	
pip-19.0.3-py3.6.egg	202 """ if unit == 'dea':	
	204 stances = stances * (pi / 180)	
v ∎ base	205 286 self nineline = VtkPineline(total time stens=stances shane[8] . 1 oif file=nif)	
to the season of	207 self.pipeline.reader_list, self.pipeline.actor_list, self.pipeline.mapper_list = sel	fsetup_pipeline_objs()
	208 self.fkine(stances, apply stance=True, actor list=self.pipeline.actor_list) 209 self pipeline add actor(axesCube(self pipeline ren))	
a model ov	211 G det execute(ob), event): 212 nonlocal stances	
a nose ny	<pre>213 self.pipeline.timer_tick()</pre>	
auaternion py	214 215 self fkine(stances, apply stance=True, actor list=self pipeline actor list, time	r=self_pipeline_timer_count)
serial link py	216 self.pipeline.iren = obj	
	217 😑 self.pipeline.iren.GetRenderWindow().Render()	
transforms.pv	219 self.pipeline.iren.AddObserver('TimerEvent', execute)	
🖧 util.ov	220 renderer = self.pipeline.iren.FindPokedRenderer(0, 0) 221 self.pipeline.iren.FlvTo(renderer, 0, -1, 6.9)	
► 🖿 media	<pre>222 self.pipeline.iren.GetRenderWindow().SetSize(800, 800)</pre>	
► 🖿 tests	223 camera = renderer.GetActiveCamera() 274 transform = camera.GetModelViewTransformObject()	
🖾 init .py	225 transform.Identity()	
🛛 🕨 🕨 robopy-1.0.8.dist-info	226 transform.RotateX(10) 227 transform.RotateY(-20)	
🖌 🕨 scipy	228 transform.Translate(0.19981, -1.53877e-13, -3.46838)	
scipy-1.2.1.dist-info	229 camera.AppLyTransform(transform) 230	
▶ ■ vtk		
🕨 🖿 vtk-8.1.2.dist-info	232 233 • Octass Link(ABC):	
🖞 📫 easy-install.pth		
🗧 🖞 setuptools.pth		
🗈 🕨 🖡 setuptools-40.8.0-py3.6.egg		
🖌 👘 vtkVersion.py	238 ♥ ♥ der init_(selt,), theta, d, a, alpha, offset=None, kind='', mdh=0, flip=None, qlim=N	
	Senaturity / animately	
I Deve extended a filosoft		
🖵 Run selected configurati		220:1 LF 🗘 UTF-8 🕆 4 spaces 🕻 💆

55. Watch the robot move from the new viewpoint.

TIP: Right-click and drag the mouse inside this window to get a good close-up view of the robot.



Now, let's talk a bit about the robot's structure.

The robot's links



This robot has several links (parts that can move separately from each other).

In this robot, each link has a unique color.

Each of these is a link.

The robot's joints



A joint is what connects two links together.

The links rotate about the joints to which they're connected.

This robot has six joints, labeled J1, J2, J3, J4, J5, and J6 in the diagram to the left.

The robot's joint angles



We can describe the robot's position by listing the angles by which each joint of the robot is rotated.

I know that doesn't make much sense, but the next several slides will show you what I mean.





























Joint angles to describe the robot's pose



We can describe the robot's current pose using its joint angles.

For example, in the image to the left, all 6 joints are at 0° rotation. If we use the variable q1 to represent the angle by which J1 is rotated, q2 to represent the angle by which J2 is rotated, and so on, then we would say that in the image to the left, the robot's pose is:

 $(q1 = 0^{\circ}, q2 = 0^{\circ}, q3 = 0^{\circ}, q4 = 0^{\circ}, q5 = 0^{\circ}, q6 = 0^{\circ}).$

Joint angles to describe the robot's pose



In this image, all joints are at 0° rotation except for J1, which is at +45° rotation. The robot's pose is: (**q1 = +45°**, q2 = 0°, q3 = 0°, q4 = 0°, q5 = 0°, q6 = 0°).

Joint angles to describe the robot's pose



In this image, all joints are at 0° rotation except for J2, which is at +45° rotation. The robot's pose is: $(q1 = 0^\circ, q2 = +45^\circ, q3 = 0^\circ, q4 = 0^\circ, q5 = 0^\circ, q6 = 0^\circ).$

Making the robot move involves nothing more than telling it to follow a sequence of poses!

$$(q1 = 0^{\circ}, q2 = 0^{\circ}, q3 = 0^{\circ}, q4 = 0^{\circ}, q5 = 0^{\circ}, q6 = 0^{\circ})$$

 $(q1 = 0.01^{\circ}, q2 = 0.02^{\circ}, q3 = 0^{\circ}, q4 = 0^{\circ}, q5 = 0^{\circ}, q6 = 0.03^{\circ})$
 $(q1 = 0.02^{\circ}, q2 = 0.04^{\circ}, q3 = 0^{\circ}, q4 = 0^{\circ}, q5 = 0^{\circ}, q6 = 0.06^{\circ})$

$$(q_1 - 0.02, q_2 - 0.04, q_3 - 0, q_4 - 0, q_5 - 0, q_6 - 0.06)$$

$$(q1 = 0.03^{\circ}, q2 = 0.06^{\circ}, q3 = 0^{\circ}, q4 = 0^{\circ}, q5 = 0^{\circ}, q6 = 0.09^{\circ})$$

. . . .

In fact, that's what your program does!


Your program creates sequences of joint angles:



Make a list, a, of 500 evenly spaced numbers between $+1^{\circ}$ and -180° .

Make a list, b, of 500 evenly spaced numbers between $+1^{\circ}$ and $+180^{\circ}$.

Make a list, c, of 500 evenly spaced numbers between +1° and +90°.

 Make a list, d, of 500 evenly spaced numbers between +1° and +450°.

• Make a list, e, of 500 zeros (0°).

Your program creates sequences of joint angles:



Carry the joint J1 through the angles in d. Carry J2 through the angles in b. Carry J3 through the angles in a, etc.

Your program creates sequences of joint angles:



Carry the joint J1 through the angles in d. Carry J2 through the angles in b. Carry J3 through the angles in a, etc.

56. Close the robot window if you haven't already.



57. Change your program to make the robot stand completely still in its zero pose (where all joint angles are at 0° rotation).



58. Make a program that makes the robot wave "Hi."



Change this to:

wave_right = np.asmatrix(np.linspace(-45, 45, 25))
wave_left = np.asmatrix(np.linspace(45, -45, 25))
wave = np.concatenate((wave_right, wave_left), axis=1)
wave_2x = np.concatenate((wave, wave), axis=1)
wave_4x = np.concatenate((wave_2x, wave_2x), axis=1)
q5 = np.transpose(np.concatenate((wave_4x, wave_4x, wave_2x), axis=1))

Run the program to see if it worked! Then close the robot window to stop the program.

59. Make a program that makes the robot spin its gripper around and around.



Run the program to see if it worked! Then close the robot window to stop the program.

60. Make a program that makes the robot spin around, pointing in various directions. This is reminiscent of Oprah Winfrey's famous quote, "You get a car! You get a car! Everybody gets a car!"



Run the program to see if it worked! Then close the robot window to stop the program. Try coming up with your own robot movements!

Can you make the robot dance?

Can you make it do a softball pitch?