PROBLEM
Students
- Hardware requirements
- Background knowledge

Instructors
- Shareable workspaces
- Reproducibility
SOLUTION
RoboStack
● Cross-platform packages for ROS 1 and 2

JupyterLab Extensions
● JupyROS
● JupyterLab-ROS
● JupyterLab-Blockly
● JupyterLab-URDF

JupyterHub
● Docker
GETTING STARTED
Get the mamba package manager

- Install **mambaforge** to get started
  
  https://github.com/conda-forge/miniforge#mambaforge

- Or try **micromamba**
  
  curl micro.mamba.pm/install.sh | bash
ROS1 ENVIRONMENT

$ mamba create -n ros1_env ros-noetic-desktop -c robostack
...
$ mamba activate ros1_env

Install extensions:

$ mamba install jupyros -c conda-forge
deproyerlab-ros -c robostack
jupyterlab-blockly -c conda-forge
jupyterlab-urdf -c conda-forge
ROS2 ENVIRONMENT

```bash
$ mamba create -n ros2_env ros-humble-desktop -c robostack-humble
...
$ mamba activate ros2_env
```

Install extensions:

```bash
$ mamba install jupyros -c conda-forge
```
import rospy
import jupyter
from std_msgs.msg import String
from geometry_msgs.msg import Pose

rospy.init_node('subpub_node')

jupyter.subscribe('/pose_stream', Pose, lambda msg: print(msg))

Step

position:
x: 1.0
y: 2.0
z: 3.0

orientation:
x: 9.0
y: 8.0
z: 7.0
w: 0.0

jupyter.publish('/pose_stream', Pose)

Send Message  □  Latch Message

```
from std_msgs.msg import String
from geometry_msgs.msg import Pose

rospy.init_node('subpub_node')

jupyter.subscribe('/pose_stream', Pose, lambda msg: print(msg))

position:
- x: 1.0
- y: 2.0
- z: 3.0

orientation:
- x: 9.0
- y: 8.0
- z: 7.0
- w: 0.0

jupyter.publish('/pose_stream', Pose)
```
JUPYROS TURTLES

ROS: tf2 Turtlesim

Requirements:

- ros-noetic-tf2
- ros-noetic-turtle-tf2

```python
import jupyter
import rospy
import math
from turtlesim.msg import Pose
from time import time
from jupyteros.rosl import TurtleSim
rospy.init_node('turtlearn')

turtlesim = TurtleSim(background_color='{}D863D9')
# The canvas default size is 1600 x 1600 starting from top-left
turtlesim.spawn(name='turtle2', pose=('x': 630,
'y': 1260,
'theta': math.radians(90))

turtlesim.turtles['turtle1'].path_color = 'bFAFF'
turtlesim.turtles['turtle2'].path_color = 'bF1C32'
display(turtlesim.canvas)
```
jupyterlab-urdf.readthedocs.io/en/latest/lite
JUPYTER BLOCKLY
def _randomize (randomized):
    randomize = Button(description='Randomize')

    for key, value in joint_state_dict.items():
Jupyter Hub: ROS in the Cloud

- Already widely adopted by universities
- Launch Jupyter Notebooks & software environments in the cloud / cluster
- SSO with uni user accounts
- ROS pre-installed and hassle free experience
- Completely free & open source

Pilot project with RWTH Aachen to teach ROS
OUTLOOK

How to help
- Missing a package? Open a PR
- Help us maintain all of ROS in RoboStack

Future
- WebAssembly ROS2 in JupyterLite
- Feature parity of Jupyter extensions between ROS 1 and ROS 2
- Integrate webviz / Foxglove into JupyterLab
THANKS!

Questions?
gitter.im/RoboStack/Lobby
@RoboStack

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, infographics & images by Freepik and illustrations by Stories