

OAK RIDGE COMPUTER SCIENCE GIRLS



SMART ROBOTICS CAMP

ORCSGirls

ORCSGirls

RULES

- Treat each other, the volunteers and the facilities with respect.
- Raise your hand if you have a question or want to answer.
- Only visit the websites needed for the class project.
- Always follow instructions by volunteers.
- Wait in the room until you are picked up at the end of the day.
- **Be gentle with the robotic arms, connectors, and monitors!**
- Help each other.
- Enjoy!



Introductions

Welcome to our camp

Instructors and Volunteers

Thomas Proffen

Amelie Nagle

Kristen Moor

Yvonne Dalschen



Teaming up



- Find the girl with the matching symbol on the badge.
- You are a team for duration of the camp.
- Interview your partner, so you can introduce her to the group.



Schedule

Wednesday	Thursday	Friday
Driver / Navigator Setting up JetMax	More coordinates and inverse kinematics	Using AI
Control via webpage	Python introduction Linux introduction	Calibration
Coordinates Measure and move activity Block stacking action recording	Moving JetMax using Python Measuring and block stacking using Python	Block Stacking Demo
		Using AI in your Python Code



This is an *Advanced* Camp

Things to remember

- You got this!
- Do not give up if things seems hard in the beginning.
- Ask questions.
- All of us ‘google’ things all the time.
- Nobody writes code that works right away.
- Debugging can be frustrating, but it is super sweet if things work in the end.
- We have a **flexible schedule**.
- Have fun!





Let's pair up

We keep these for the entire camp.

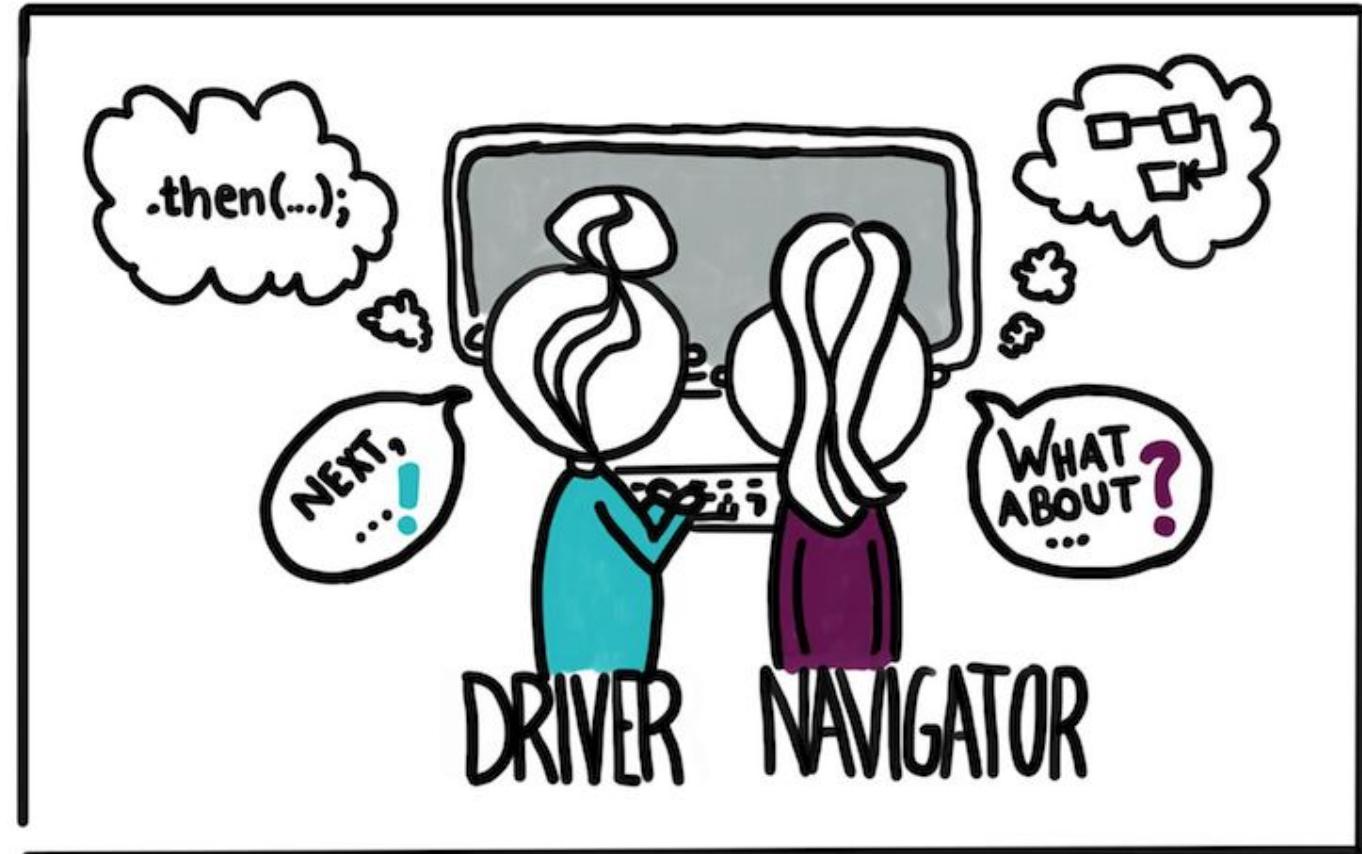
Pair Programming

Two girls

One laptop

One robotic arm

Unstoppable



Driver

- **Write the code or run commands according to the navigator's specification**
- **Do not keep going just because you know what is next.**
- Listen intently to the navigators instructions
- Ask questions wherever there is a lack of clarity
- **Offer alternative solutions if you disagree with the navigator**
- If there is disagreement, defer to navigator. If idea fails, get to failure quickly and move on
- **Own the computer / keyboard**
- **Trust the navigator** - ultimately the navigator has the final say in what is done
- You are writing the code, clicking the buttons and run commands



Navigator

- **Dictates the code that is to be written or commands to be run - the 'what'**
- Clearly communicates what code to write or what to enter
- **Explains 'why' they have chosen the particular solution to this problem**
- Check for syntax / type errors as the Driver drives
- **Be the safety net for the driver**
- Make sure that the driver sticks to the small task at hand
- **Ongoing code review**
- Pay attention



Both

Working together is the key!

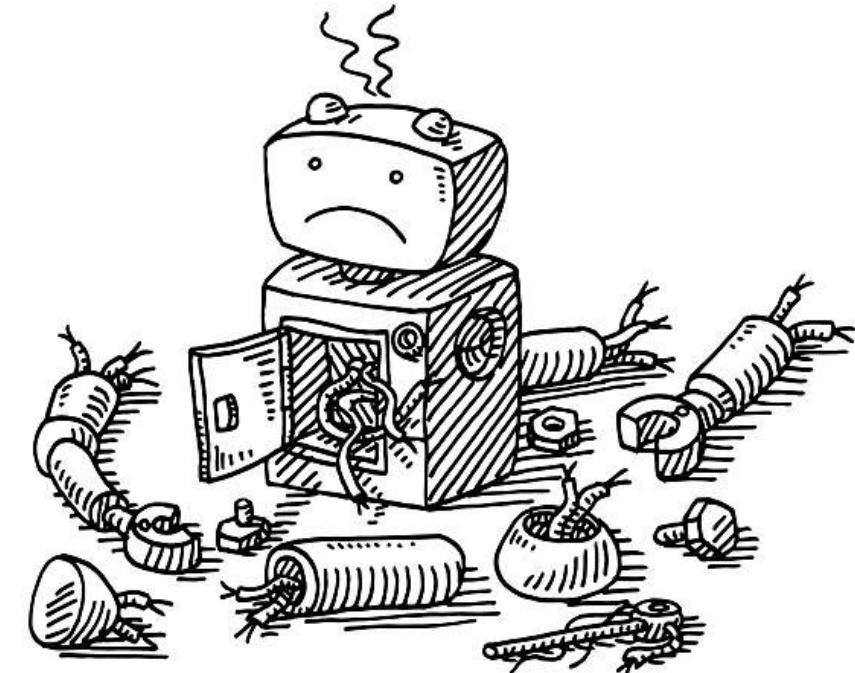
- **Actively take part in programming and operating the arm**
- Aim for optimal flow - avoid trying to be 'right'
- Embrace your role
- Intervene if your pair is quiet
- Know when to swap roles or take a break
- **Communicate, communicate, COMMUNICATE!**
- **Don't hog the keyboard**
- **High-five every time something works high**
- **Swap roles frequently**



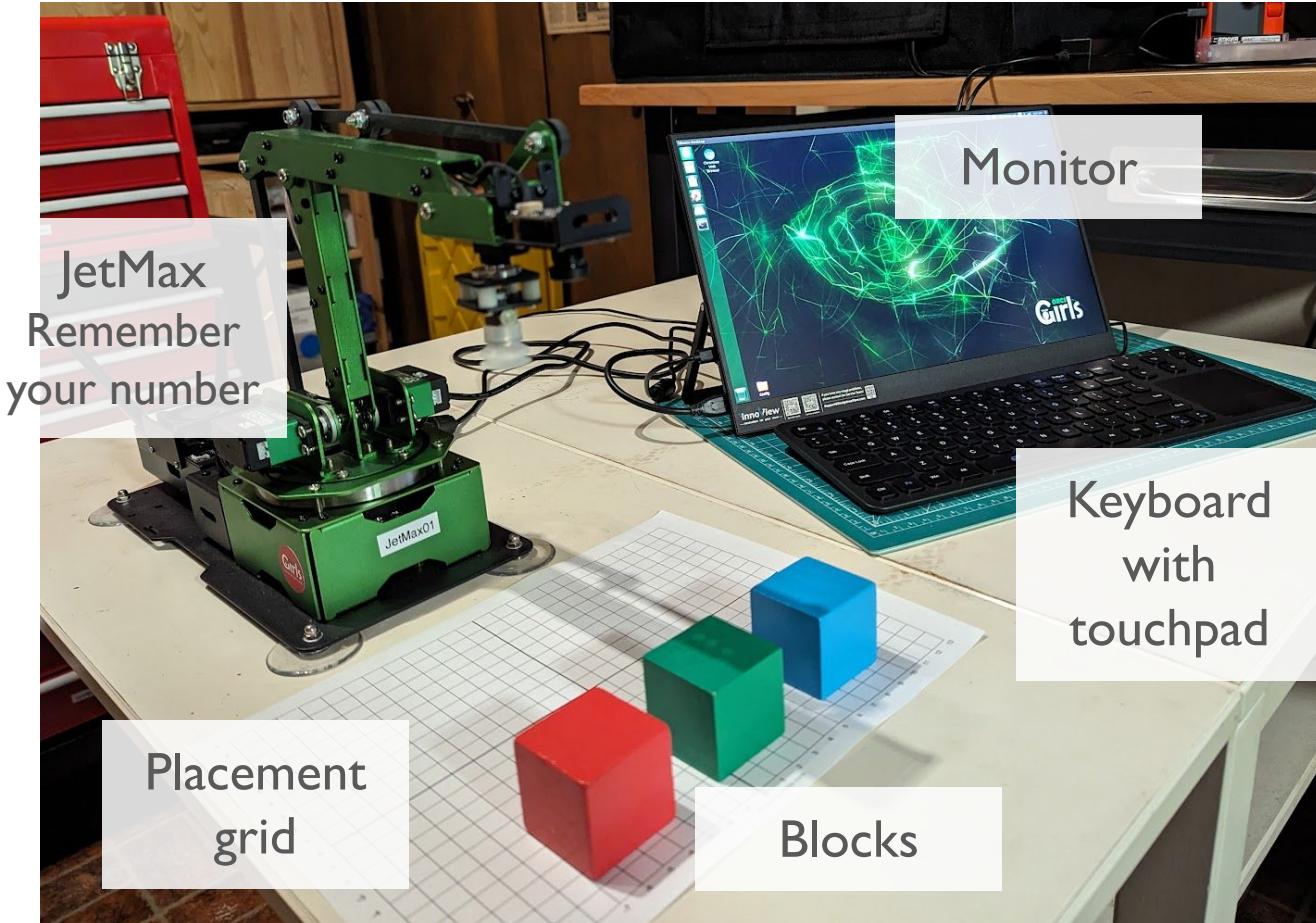
Robot rules

Take turns - communicate - stay engaged
- don't fight over who does what!

- Only pickup robotic arm by its base
- Be gentle when plugging in cables
- Do not move the arm manually
- Use proper shutdown before powering off
- Use slow speed for moving
- Be patient, commands might take a while
- Double check your coordinates are right before moving, so the arm does not collide with something
- Keep your work area clean



Setting up the JetMax



Clear a workspace

Decide who picks up what

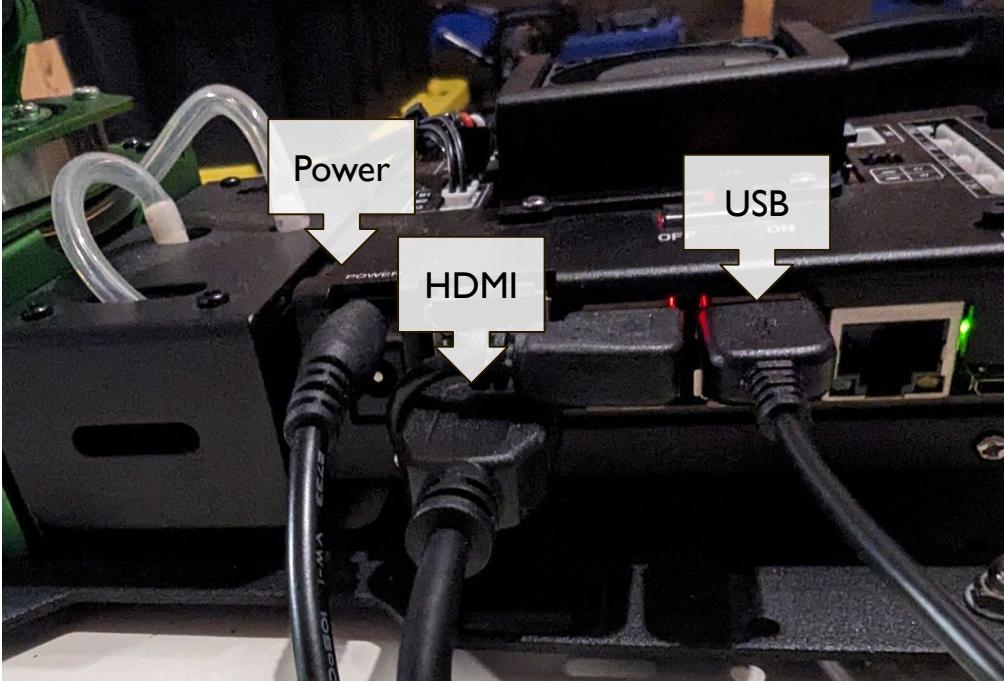
Volunteers will hand our materials

You need

- JetMax robot arm
- Monitor
- Keyboard
- Placement grid
- Three color blocks
- Power supply JetMax
- Power supply monitor
- HDMI cable



Connecting stuff ..



Make sure JetMax switch is turned off

Connect power (careful, this might be hard to get in, ask for help if needed)

Connect HDMI cable from monitor

Connect USB connector from keyboard

Be gentle with the connectors

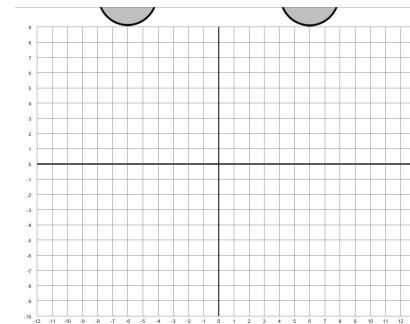


Connect USB-C from power supply

Connect HDMI cable going to JetMax

Place grid in front of JetMax

Match front suction feet with marks on sheet.

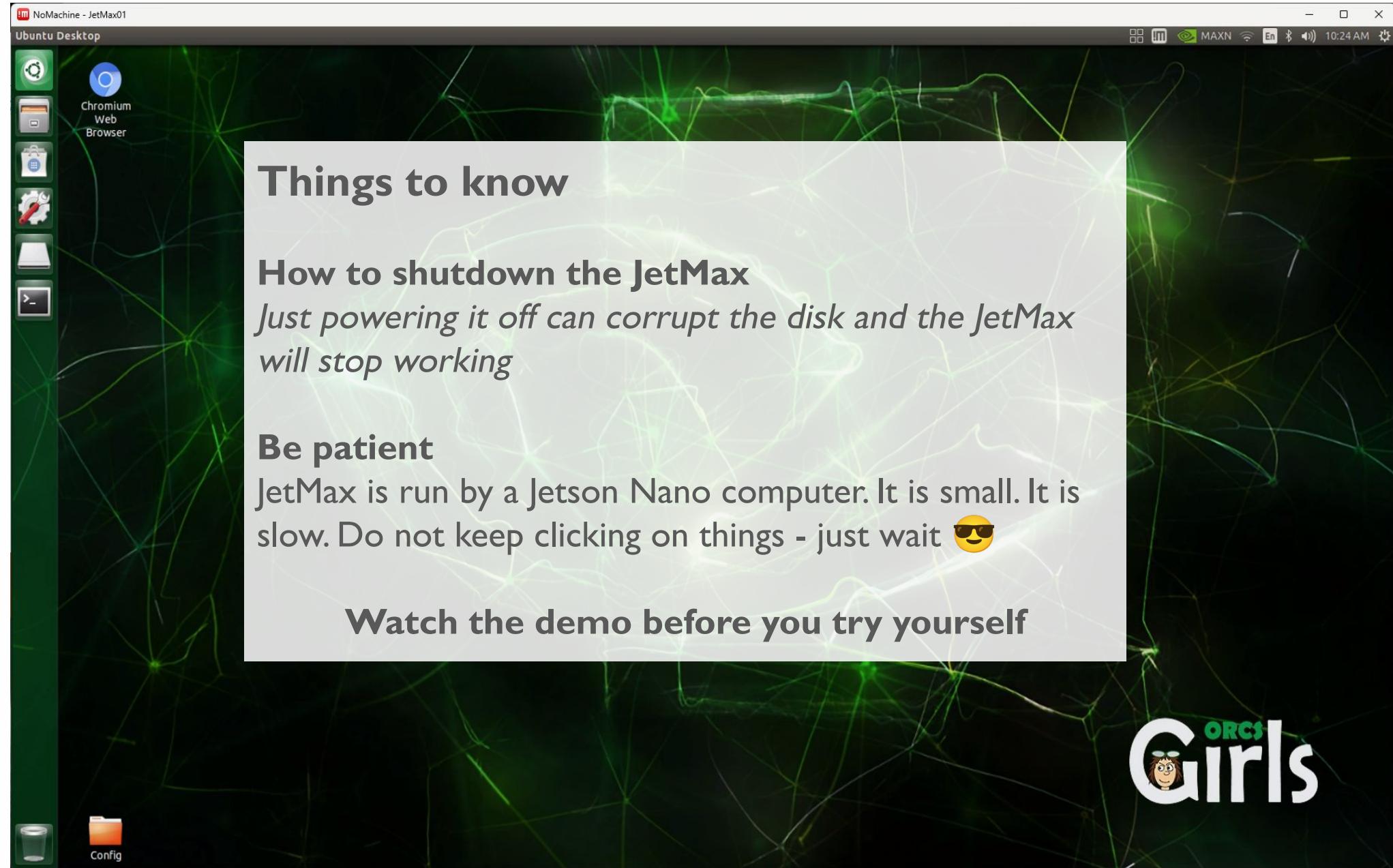


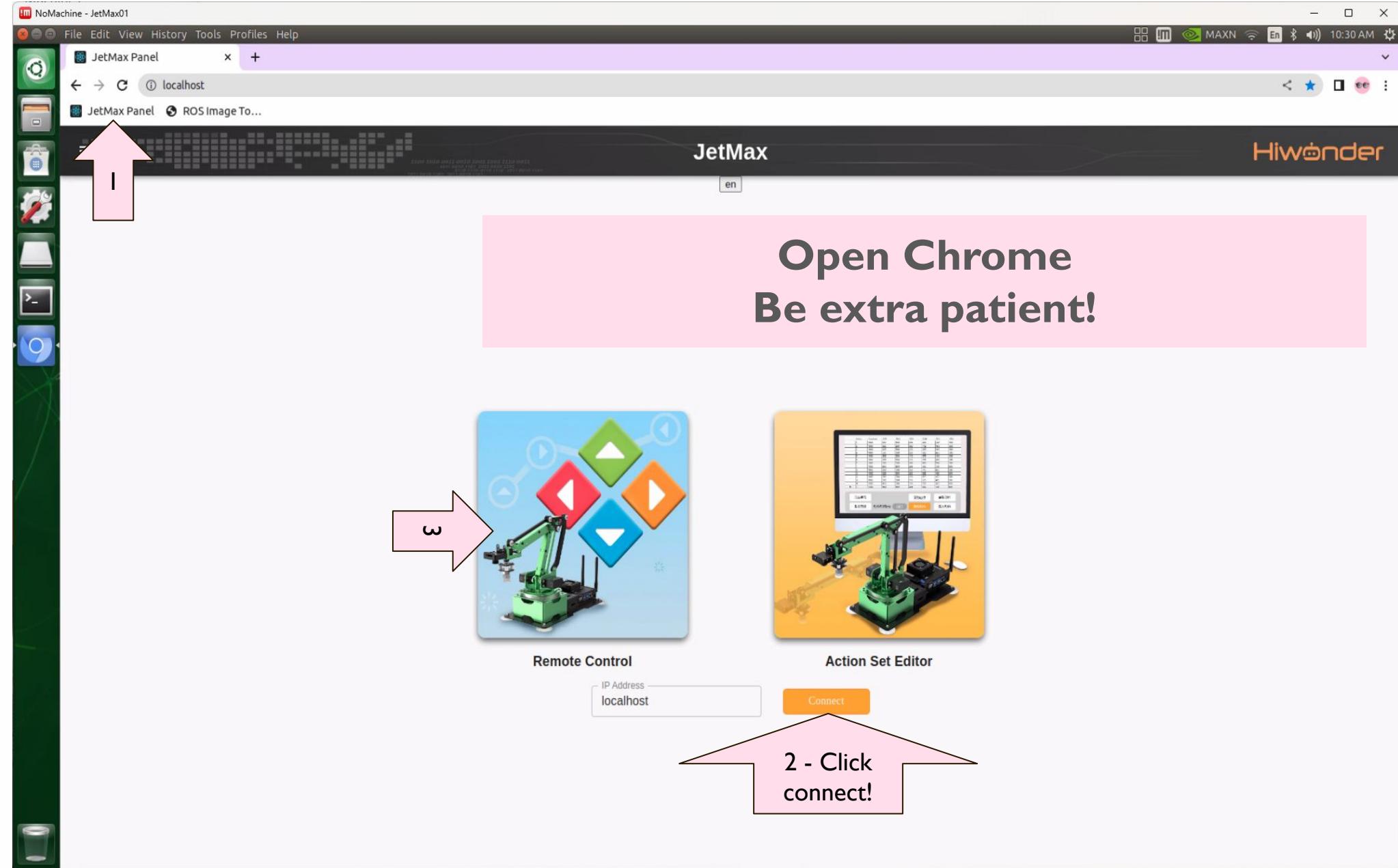
Turn the JetMax and monitor on

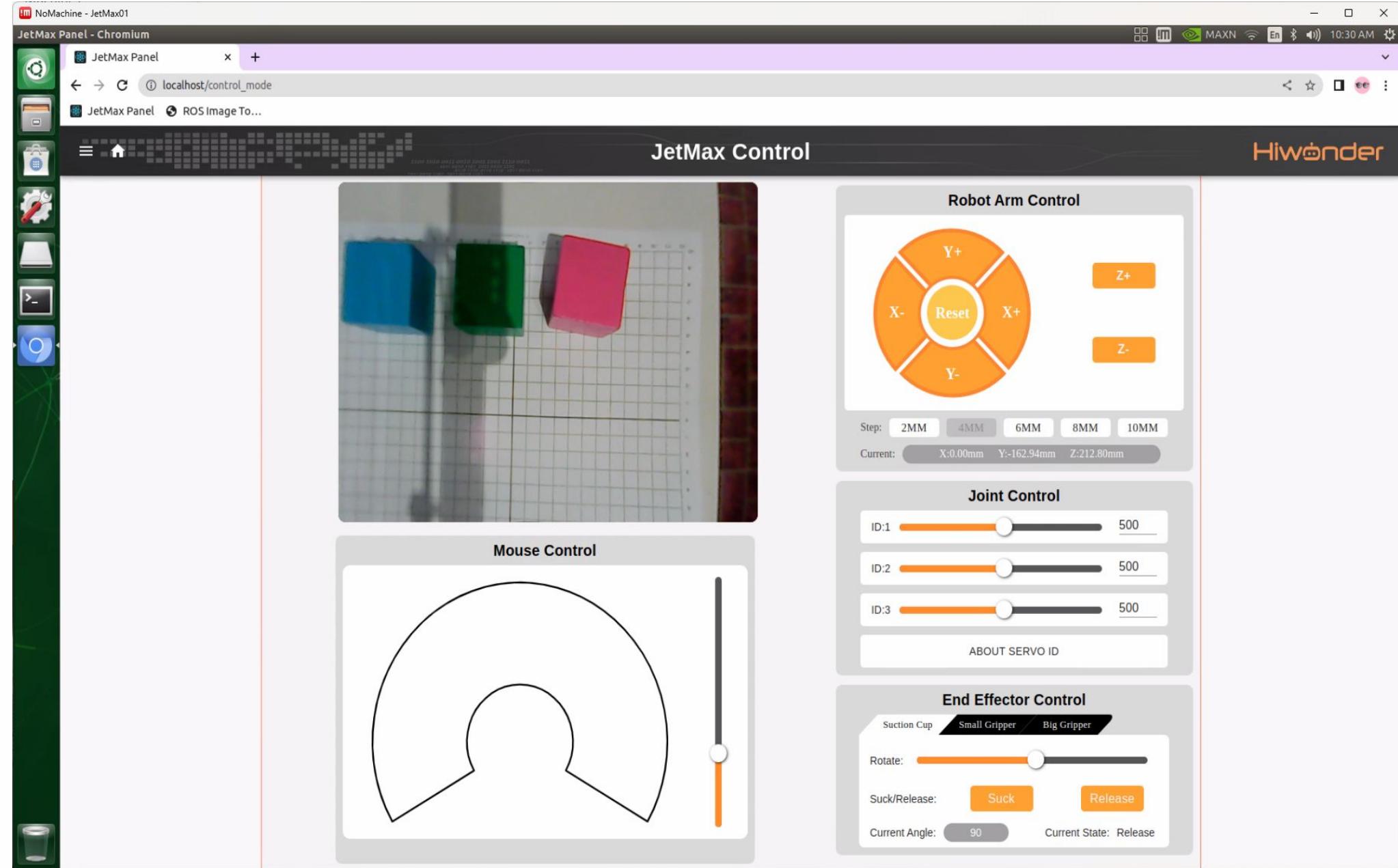
Wait for the beep

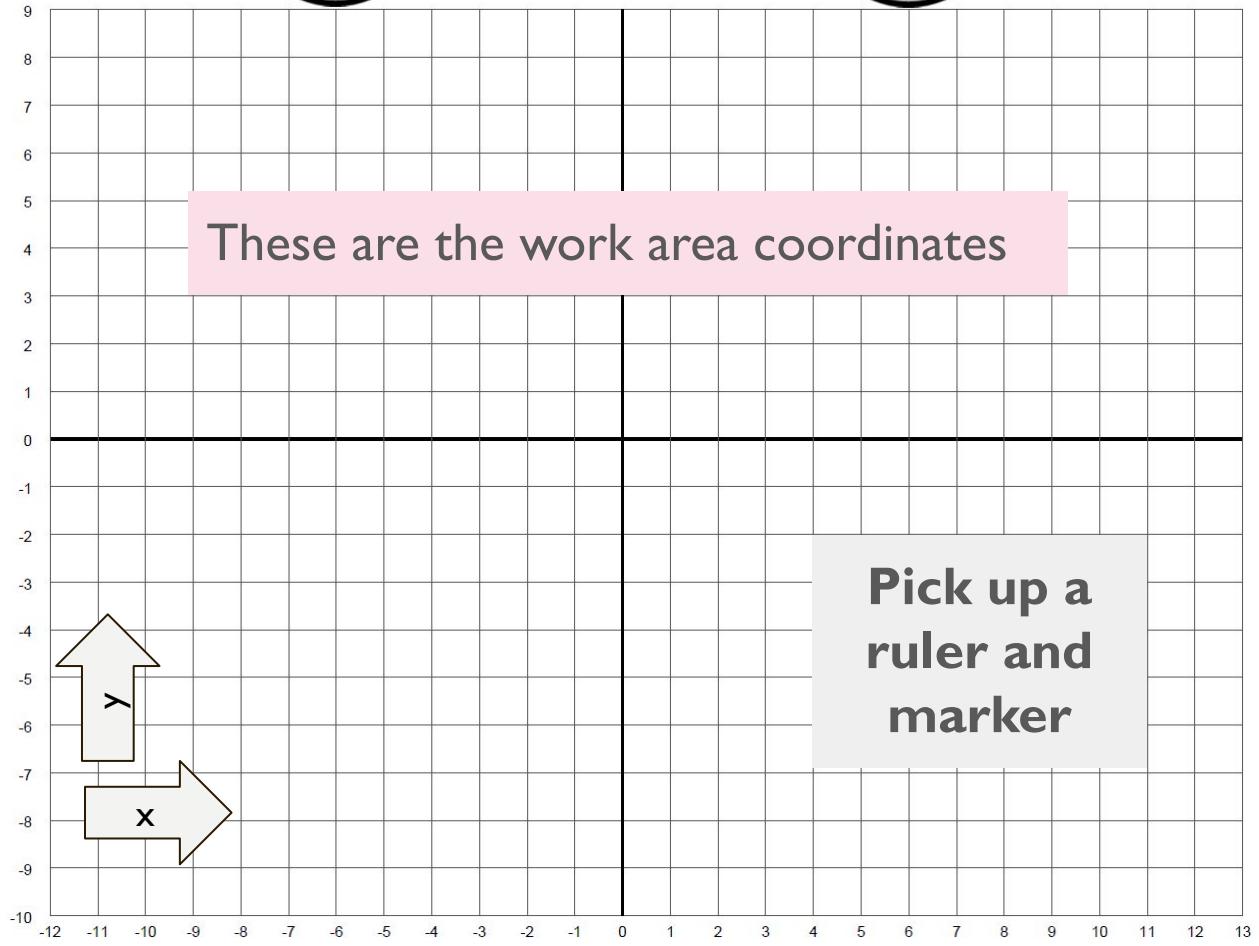
WAIT
We will start together











These are the robot coordinates

Measure and Move Activity

How to convert from robot to work area coordinates?

Where is (0,0,0) on the robot arm?

Challenge: put a block on the grid and determine robot coordinates

Check by driving there

The Action Set Editor interface features a top bar with tabs: POSITION, X, Y, Z, PWM1, PWM2, and SUCKER. Below the tabs is a large empty workspace. At the bottom are several control buttons: Duration (set to 1 s), AutoRepeat (unchecked), ActionSet (set to test), and a row of orange buttons labeled Add, Insert, Replace, Delete, Play, Step, Stop, and Clear.

Do not forget to **save** and have a different name for each of you.

Stacking Challenge

Start the Action Set Editor

You can now save positions of the arm and replay them.

Play around

Challenge

- Put three blocks on the grid and mark the position
- Learn the actions to stack them.
- Put blocks back and replay - does it stack



Showtime

Let's check out
who stacks the
best 

That's it for today

- **Shut down the JetMax arm properly**
- Wait for the screen to go blank
- Switch the arm off
- Carefully disconnect cables
- Bring all parts up and hand them to the volunteers
- **Remember your number, so you have the same robotic arm next time**
- See you tomorrow 



OAK RIDGE COMPUTER SCIENCE GIRLS



SMART ROBOTICS CAMP

Day 2

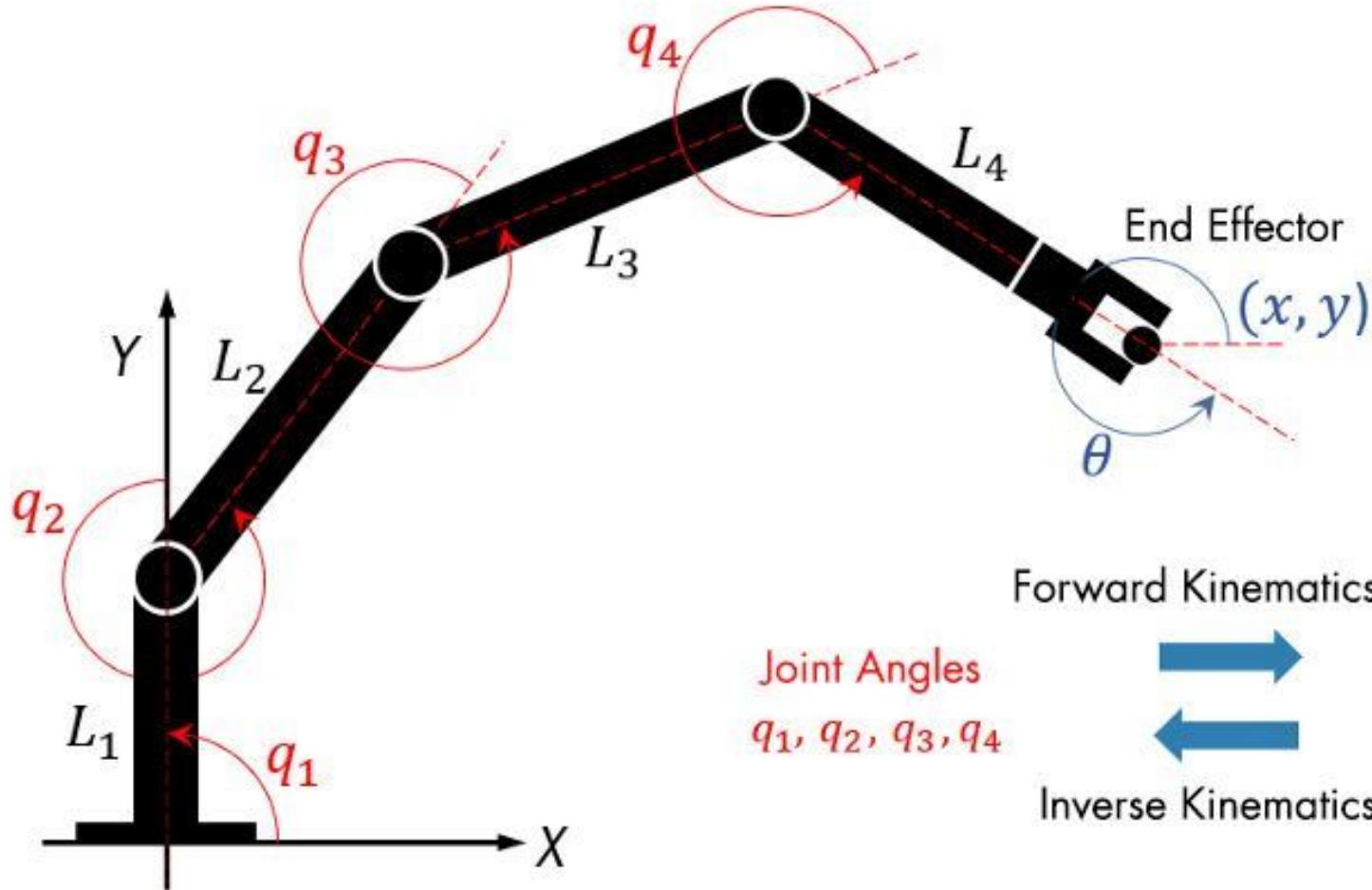
ORCSGirls

ORCSGirls

Review what we learned

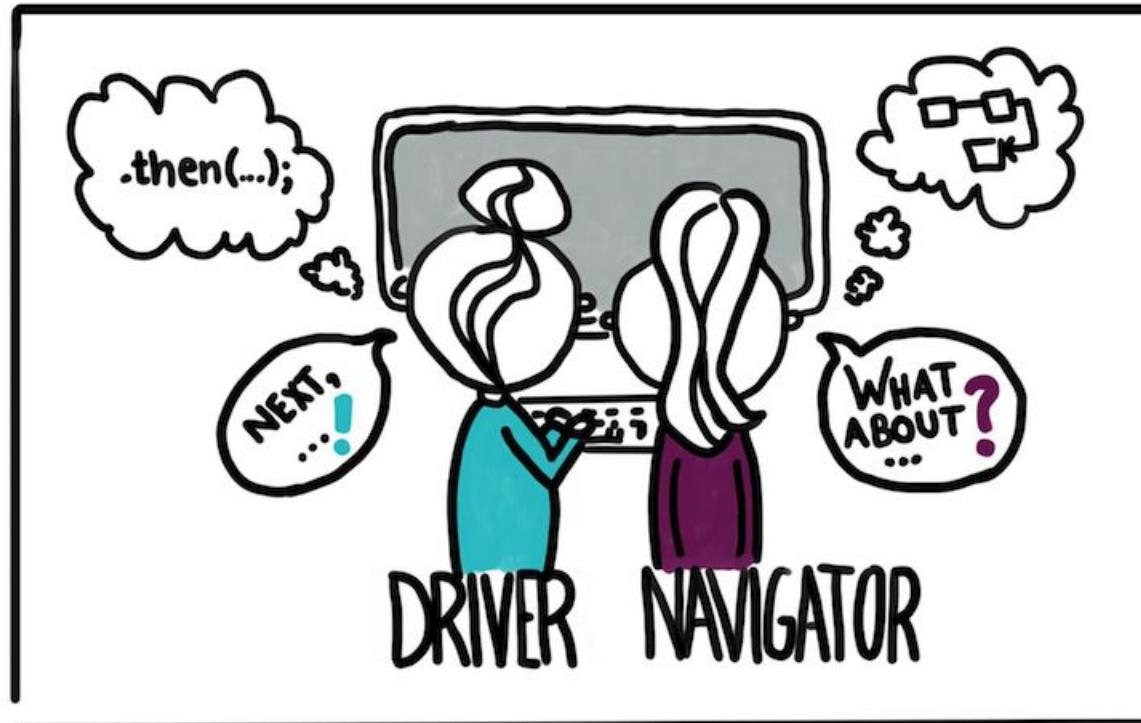


Kinematics

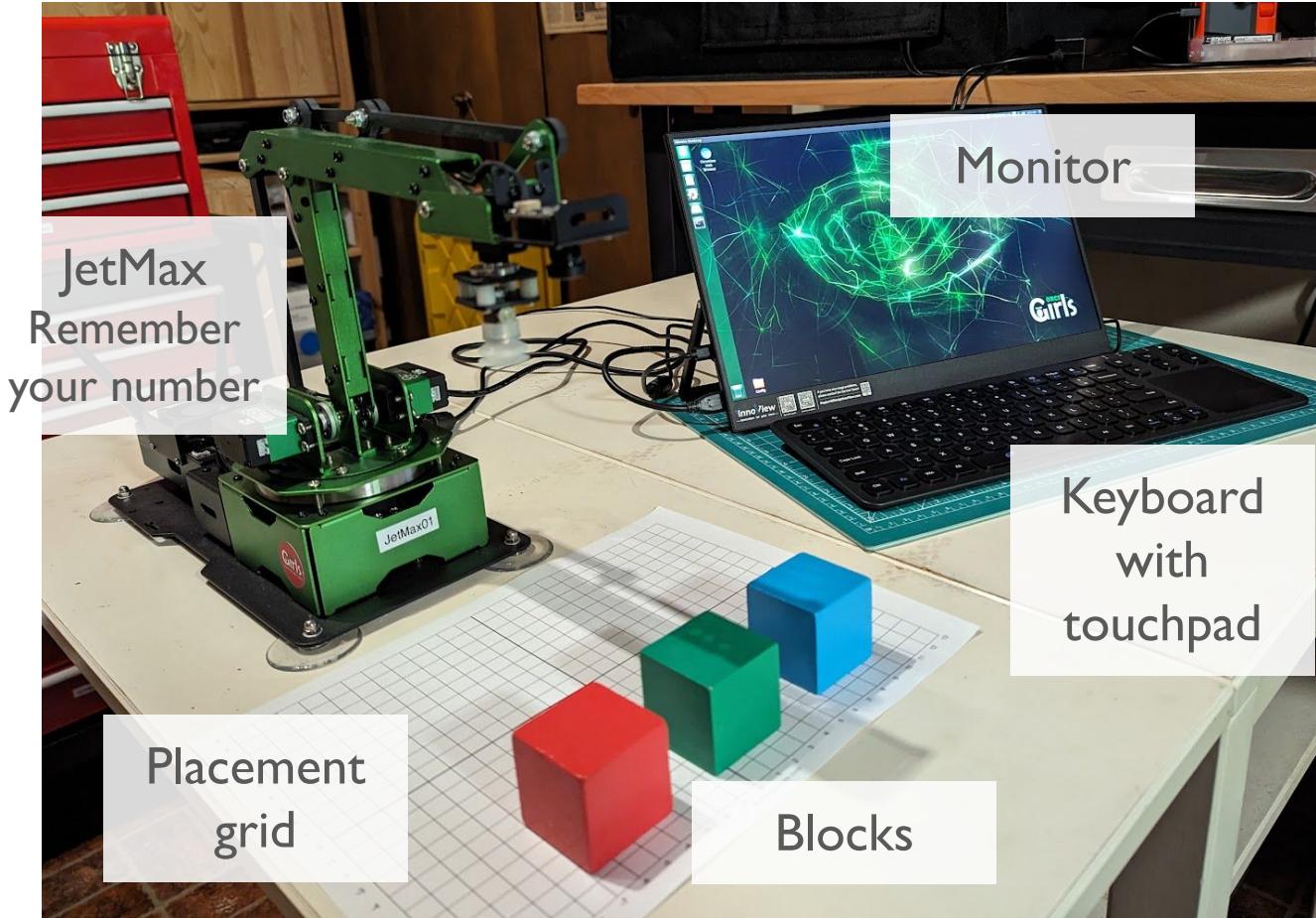


Remember *Driver Navigator* rules

- Make sure everyone gets to perform every task
- Take turns in setting up
- Help each other
- Listen and communicate
- Great teamwork leads to great progress and results
- Valuable skill to learn



Setting up the JetMax



Clear your workspace

Decide who picks up what

Volunteers will hand our materials

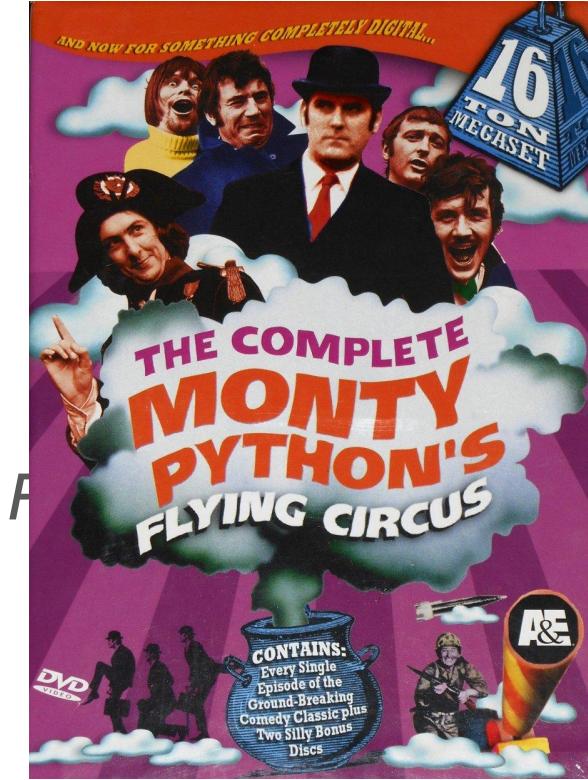
You need

- JetMax robot arm
- **Use same robot number**
- Monitor
- Keyboard
- Placement grid
- Three color blocks
- Power supply JetMax
- Power supply monitor
- HDMI cable



WHAT IS PYTHON?

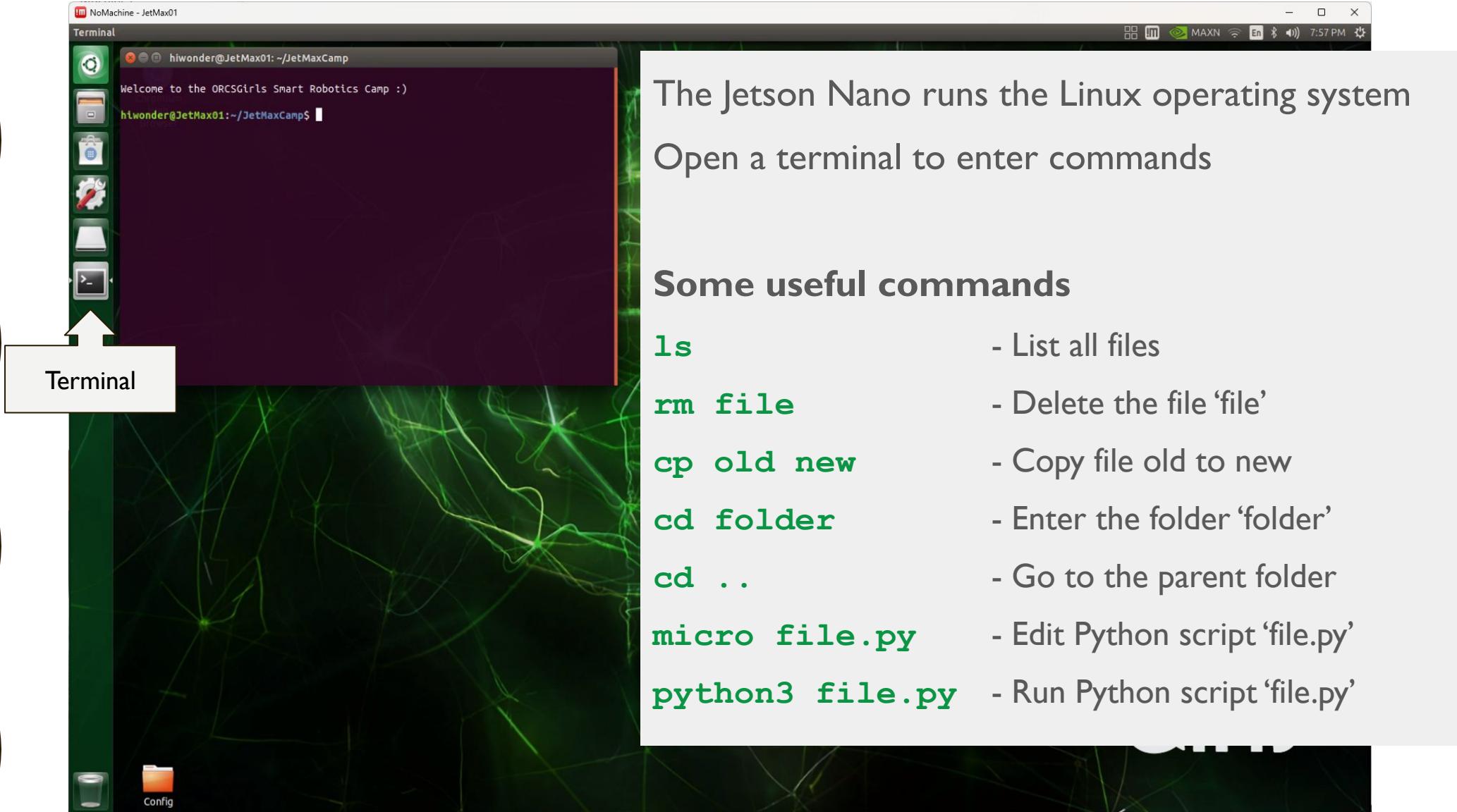
- Python is an interpretive programming language.
- Python was created by Guido van Rossum in 1989.
- Named after his love for the comedy series *Monty Python's Flying Circus* – not the snake.
- Benefits
 - Easy to learn.
 - Free to use and open source.
 - Portable, extensible, and embeddable.
 - Object oriented language.
- Fastest growing programming language.



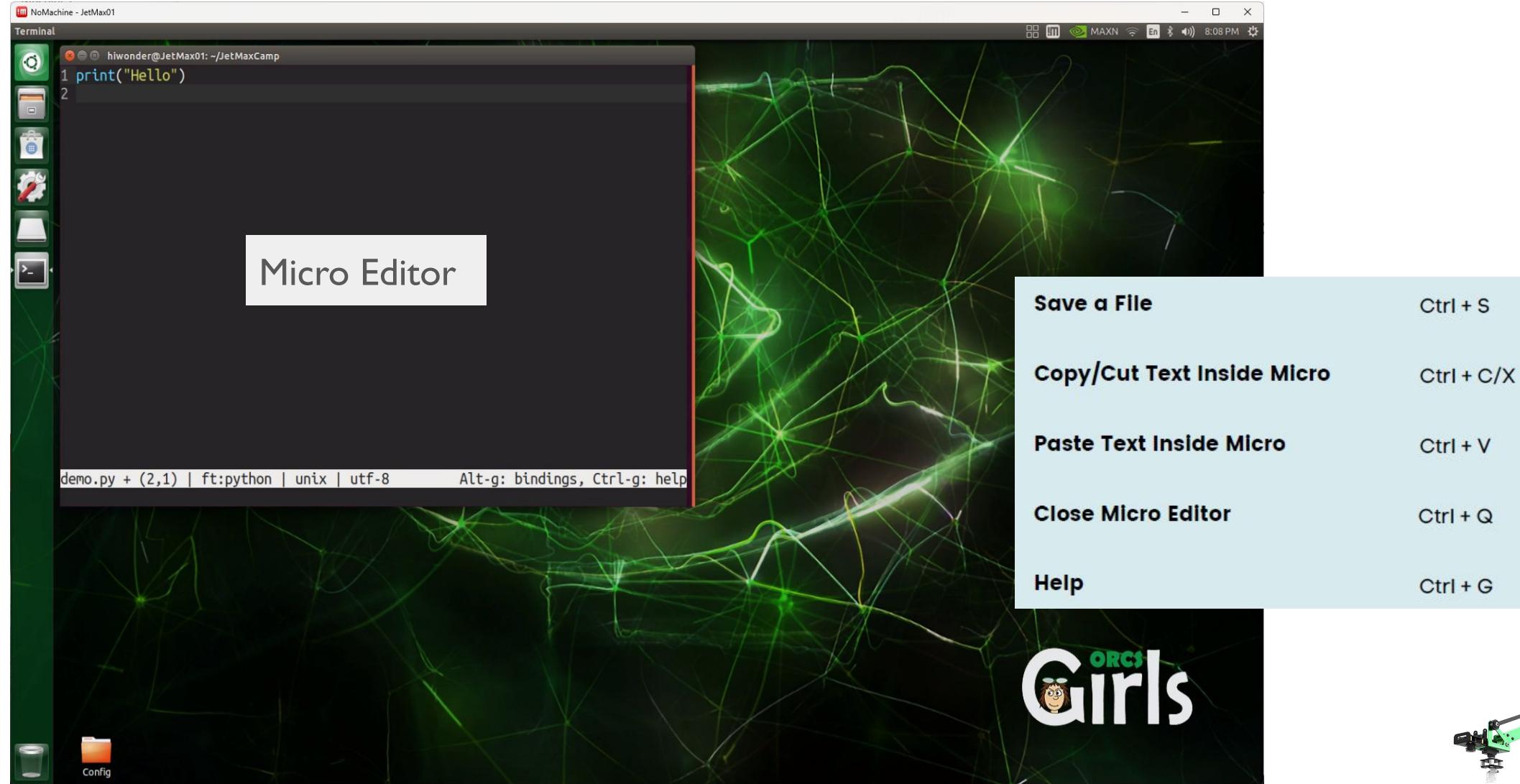
Source: Wikipedia



Using the terminal



Using the terminal



Write our first code

Remember to takes turns who drives and who navigates

- Open a terminal
- Change to the Python folder - `cd Python`
- Create a new file and open in editor - `micro first.py`
- In the editor type - `print("This camp is awesome")`
- Save (control s) and quit the editor (control q)
- Run your code with `python3 first.py`
- Edit the code and make changes - explore Python.
- Brainstorm some simple programs together, e.g. print numbers from 1 to 10 or try the temperature converter from the handout)

Python 3 cheatsheet (the basics)

Interact with the user (input and output)

Print a message
`print('Hello, world!')`

Print multiple values (of different types)
`ndays = 365
print('There are', ndays, 'in a year')`

Asking the user for a string
`name = input('What is your name? ')`

Asking the user for a whole number (an integer)
`num = int(input('Enter a number: '))`

Text (strings)

Single quoted
`'perfect'`

Double quoted
`"credit"`

Multi-line
`'''Hello,
World!'''`

Add (concatenated) strings
`'Hello' + ' World'`

Multiply string by integer
`'Echo...'*4`

Length of a string
`len('Hello')`

Convert string to integer
`int('365')`

Variables

Creating a variable
`celsius = 25`

Using a variable
`celsius*9/5 + 32`

Whole numbers (integers)

Addition and subtraction
`365 * 1 - 2`

Multiplication and division
`25*9/5 + 32`

Powers (2 to the power of 8)
`2**8`

Convert integer to string
`str(365)`

Decide between options

Decide to run a block (or not)
`x = 3
if x == 3:
 print('x is 3!')`

Are two values equal?
`x == 3
△ two equals signs, not one`

Decide between two blocks
`mark = 80
if mark >= 50:
 print('pass')
else:
 print('fail')`

Are two values not equal?
`x != 3
△ two equals signs, not one`

Decide between many blocks
`mark = 80
if mark >= 65:
 print('credit')
elif mark >= 50:
 print('pass')
else:
 print('fail')`

Less than another?
`x < 3`

Greater than another?
`x > 3`

Less than or equal to?
`x <= 3`

Greater than or equal to?
`x >= 3`

`*elif can be used without else`

`*elif can be used many times`

The answer is a Boolean:
`True` or `False`

Repeat a block (a fixed number of times)

Repeat a block 10 times
`for i in range(10):
 print(i)`

Count from 0 to 9
`range(10)`

Sum the numbers 0 to 9
`total = 0
for i in range(10):
 total = total + i
print(total)`

Count from 0 to 10
`range(11)`

Count from 1 down to 1
`range(10, -1)`

Repeat a block over a string
`for c in 'Hello':
 print(c)`

Count 2 at a time to 10
`range(0, 11, 2)`

Keep printing on one line
`for i in 'Hello':
 print(i)`

Count down 2 at a time
`range(10, 0, -2)`

Repeat a block over list (or string) indices

`msg = 'I grok Python!'
for i in range(len(msg)):
 print(i, msg[i])`

Putting it together: Celsius to Fahrenheit converter

Ask the user for a temperature in degrees Celsius
`celsius = int(input('Temp. in Celsius: '))`

Calculate the conversion
`Fahrenheit = celsius*9/5 + 32`

Output the result
`print(Fahrenheit, 'Fahrenheit')`

Use the handout
We will start together



Controlling the robot in Python

```
from myjetmax import *
jetmax = myJetMax()

# This moves to the given coordinates x,y and height z as on the grid.
# Units are mm - here we make a square
# The last number is the duration for the move - here 1 second

jetmax.move_to( 0, 0,30, 1)
jetmax.move_to(80, 0,30, 1)
jetmax.move_to(80,80,30, 1)
jetmax.move_to( 0,80,30, 1)
jetmax.move_to( 0, 0,30, 1)

# This moves relative to the last position
# Here 40 mm up.

jetmax.move_to( 0, 0,40, 1, relative=True)

# This returns the current location of the sucker
(cx,cy,cz) = jetmax.location()

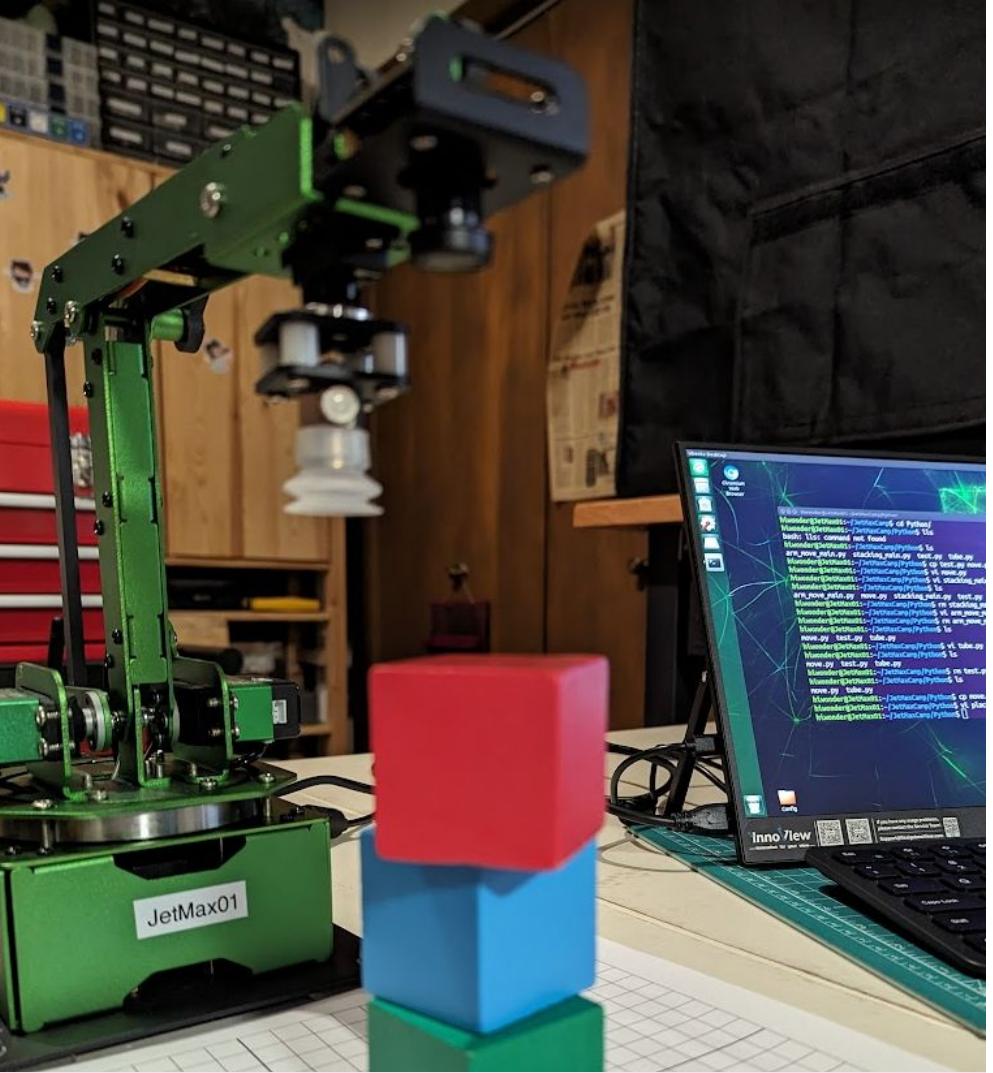
# Set angle of the sucker
jetmax.set_angle(10)

# Turn sucker on and off
jetmax.suck()
time.sleep(1.0)
jetmax.release()
```

JetMax Commands

- Open a terminal
- Go to the Python folder
- Run the **test.py** program
- Edit the program and try different moves
- **Can you pick up a block?**





Make sure the grid mat does not move.

Remember to use Driver Navigator rules
when writing and testing the code!

JetMax Coding Challenge

- Select two locations on the grid and note there x and y
- Measure the height of a block
- Put a block on the start location
- You program should pickup the block and move it to the other location

- Expand to pickup two blocks and stack them

Showtime
Let's check out
all the
awesome code

That's it for today

- Shut down the JetMax arm properly
- Wait for the screen to go blank
- Switch the arm off
- Carefully disconnect cables
- Bring all parts up and hand them to the volunteers
- Remember your number, so you have the same robotic arm next time
- See you tomorrow 



OAK RIDGE COMPUTER SCIENCE GIRLS



SMART ROBOTICS CAMP

Day 3

ORCSGirls

ORCSGirls

Review what we learned



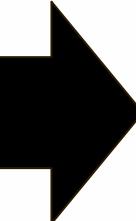
Smart Robotics Camp



TRADITIONAL PROGRAMMING

**Rules for
“Is this picture a pig?”**

Is it pink?
Does it have a tail?
Pointed ears?
Four legs?
And so on...



**Traditional code for “Is this
picture a pig?”**

```
if (isPink(picture) and
    hasTail(picture) and
    hasPointedEars(picture)
    and hasFourLegs(picture)
    and ...)
    print "Yes!"
else
    print "No!"
```



Smart Robotics Camp



Yes, even in Oak Ridge



**HOW DO YOU DECIDE
WHAT THE RULES ARE?**

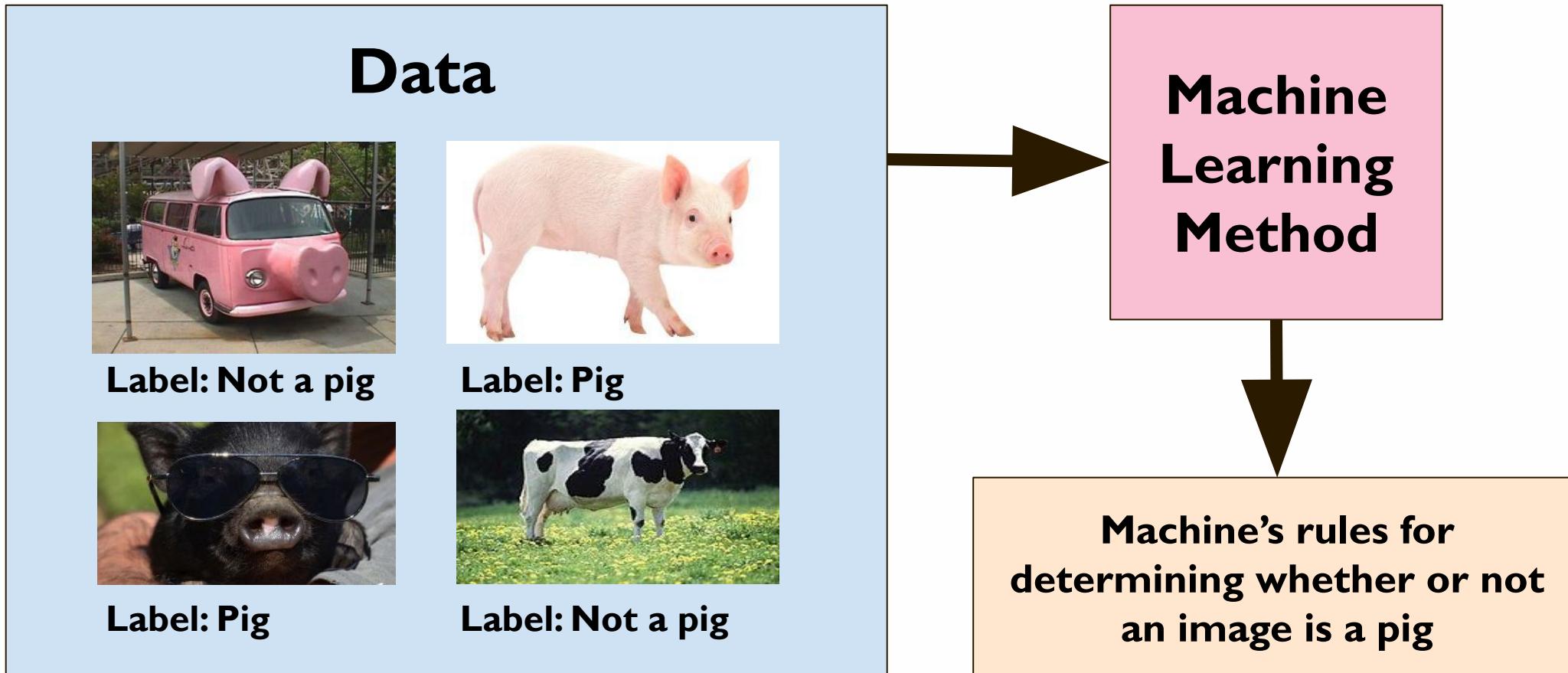


**YOU DON'T!
LET THE COMPUTER DO IT WITH
MACHINE LEARNING!**

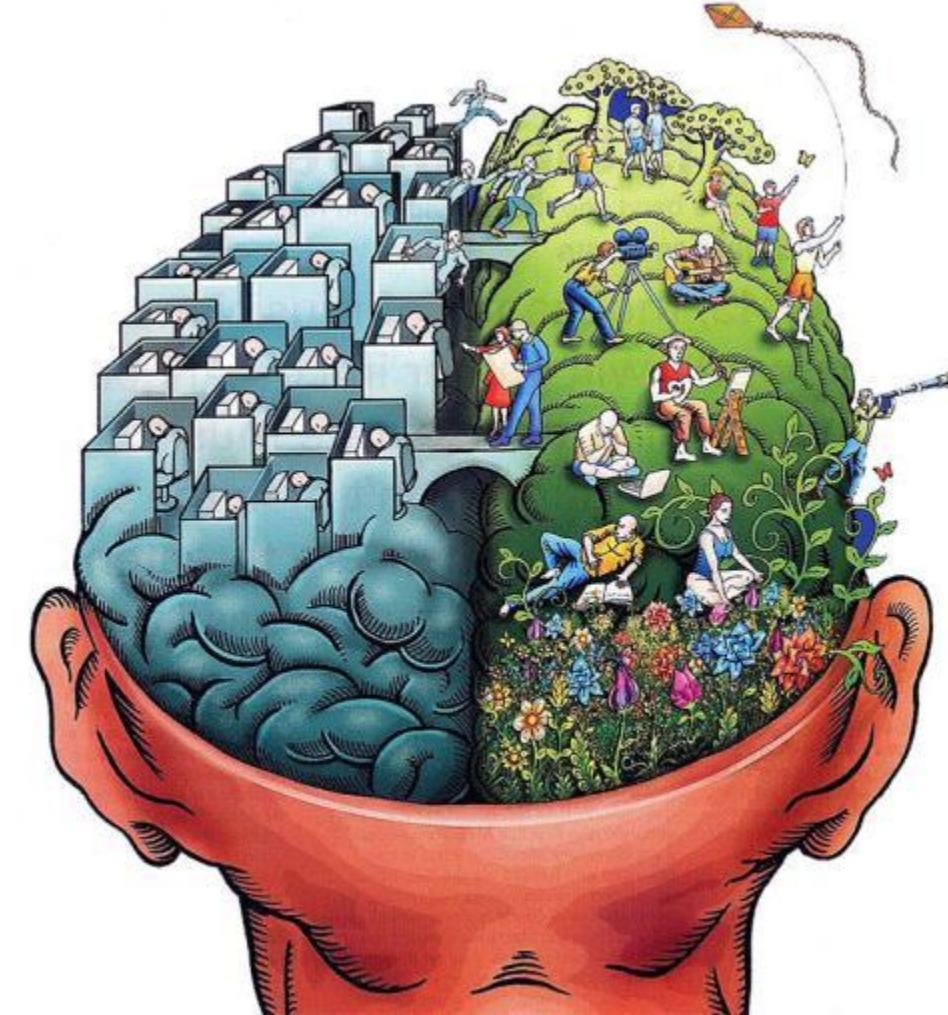


MACHINE LEARNING

A machine learning method takes a bunch of data and “learns” from it!



WHAT'S THE DIFFERENCE BETWEEN MEMORIZING AND LEARNING?



DID IT “LEARN” SOMETHING?



Label: Not a pig



Label: Pig



Label: Pig



Label: Not a pig



Label: Not a pig



Label: Pig

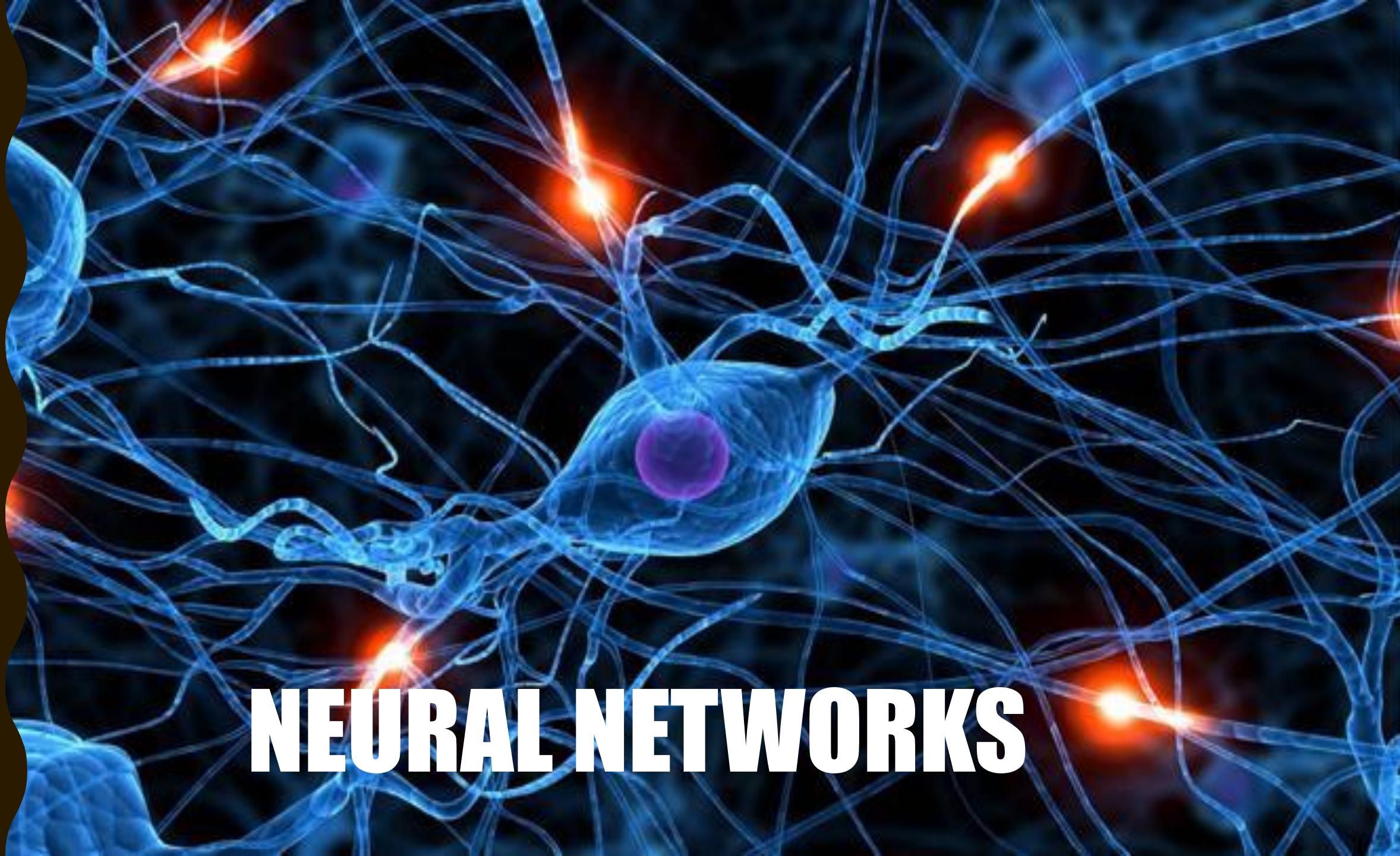
Training Data

The data we give to the machine learning method to learn from

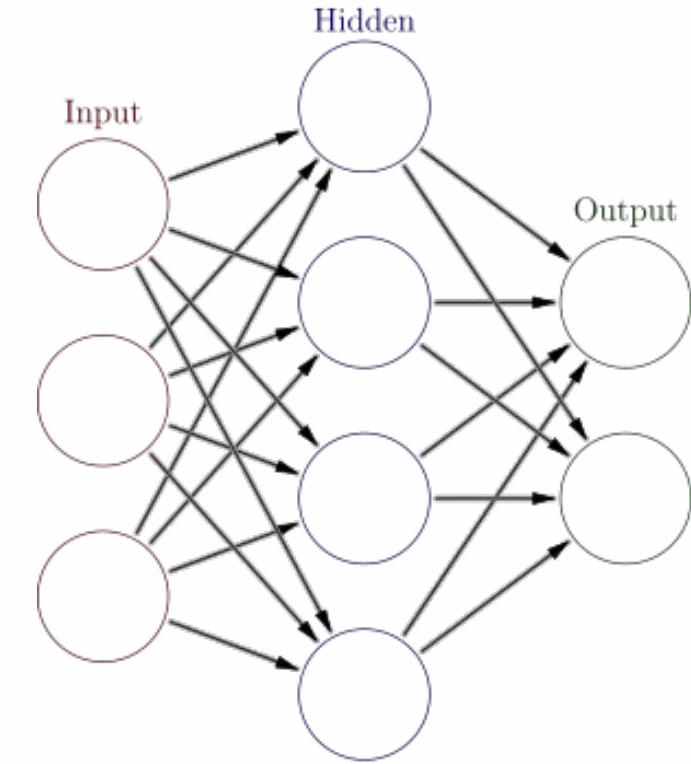
Testing Data

The data we hold out and use to check to see if the method actually learned something!

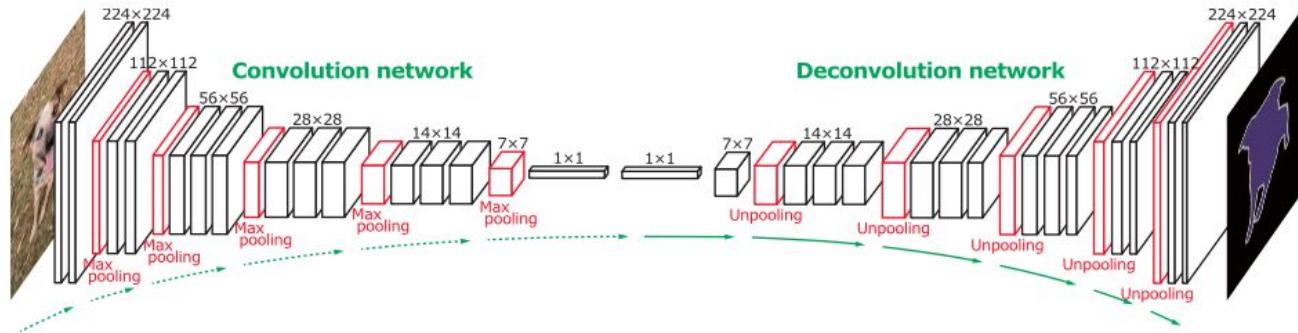
NEURAL NETWORKS



NEURAL NETWORK EXPLANATION



Let's use
this as an
example

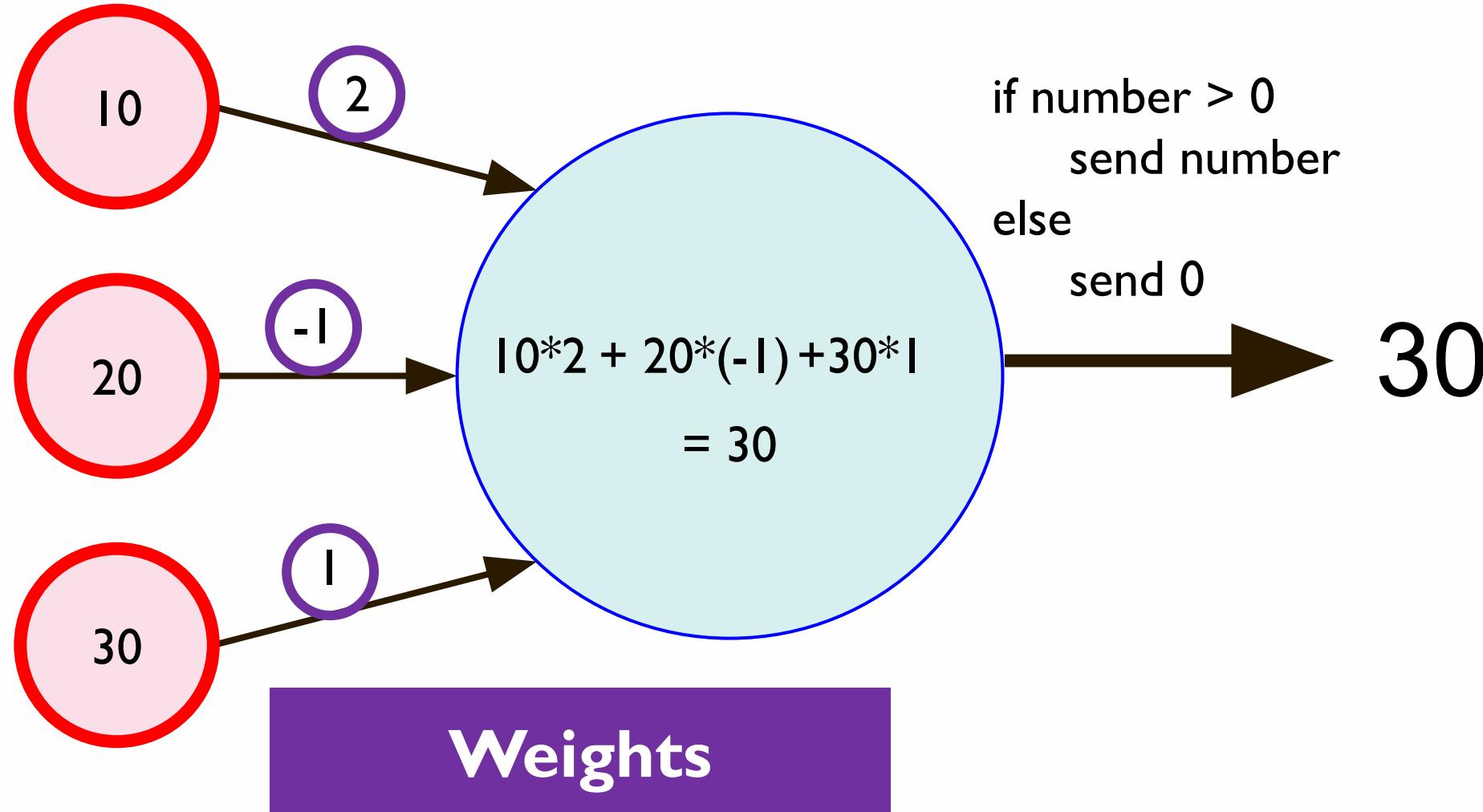


ChatGPT-3

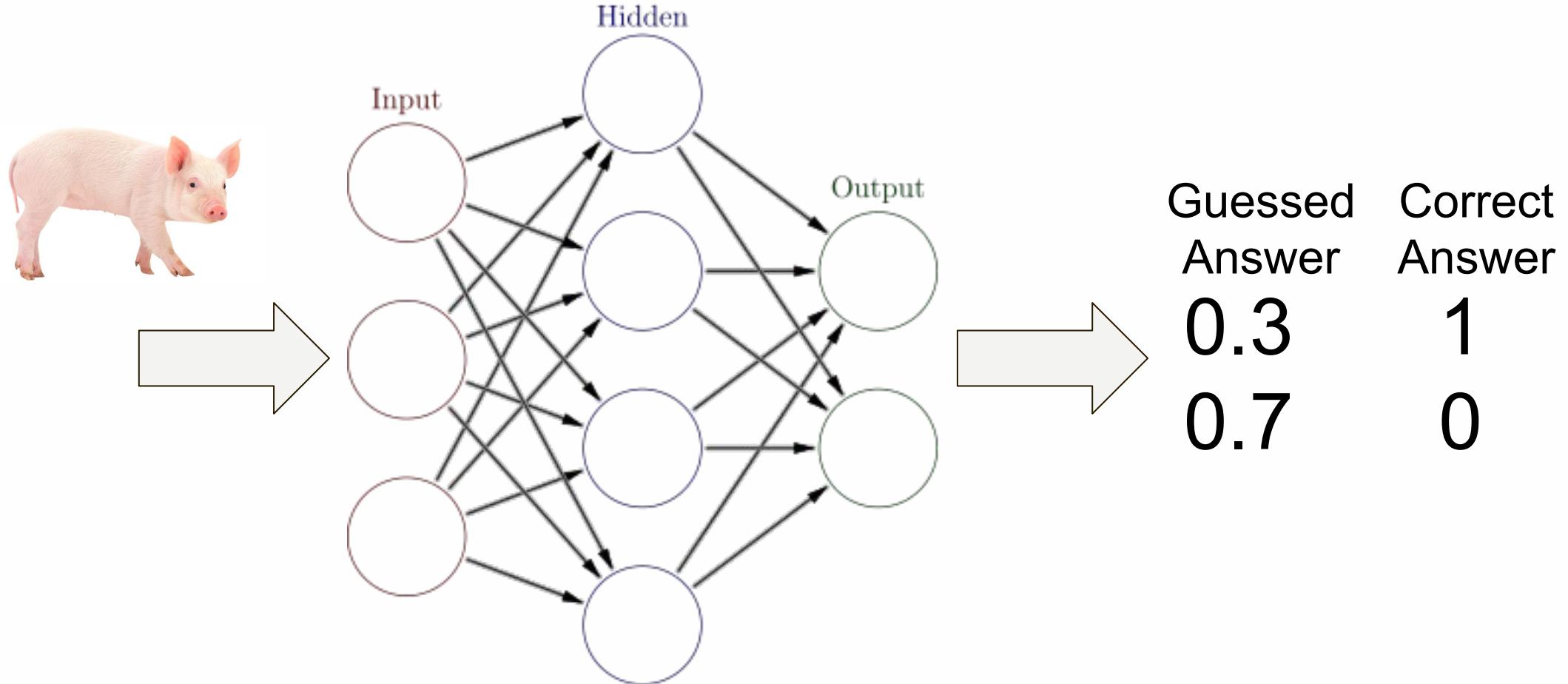
- Number of layers: 96
- Number of attention heads: 96
- Dimensions of its hidden layers: 12288
- Sequence length: 2048
- **Number of parameters: 175 Billion**



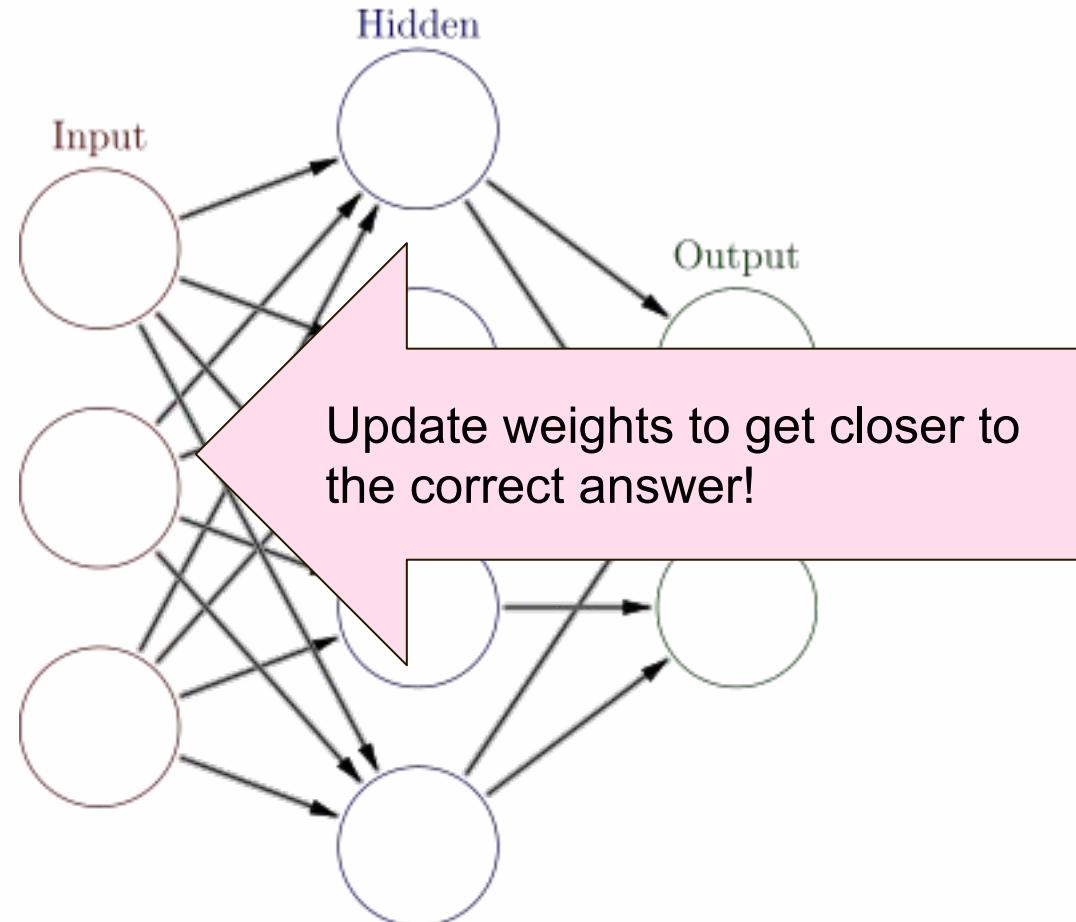
NEURAL NETWORK EXPLANATION



Neural Network Explanation



Neural Network Explanation

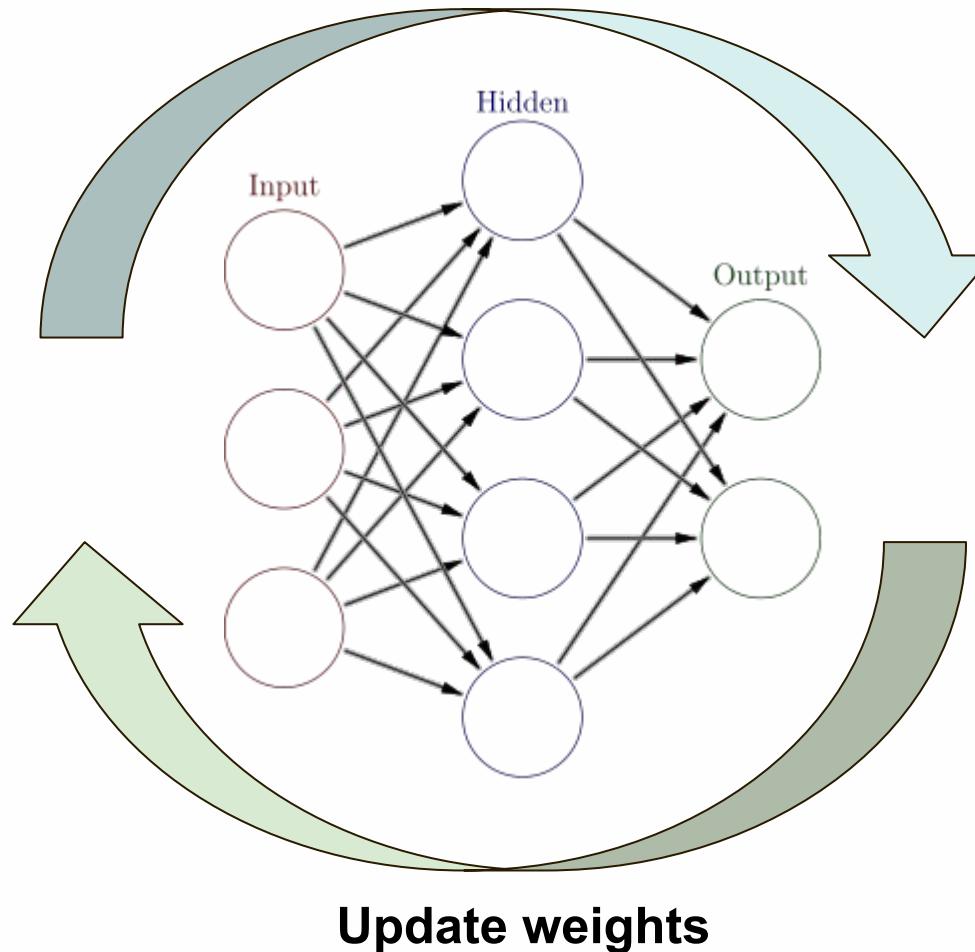


Guessed Answer	Correct Answer
0.3	1
0.7	0



Neural Network Explanation

Run a data example through

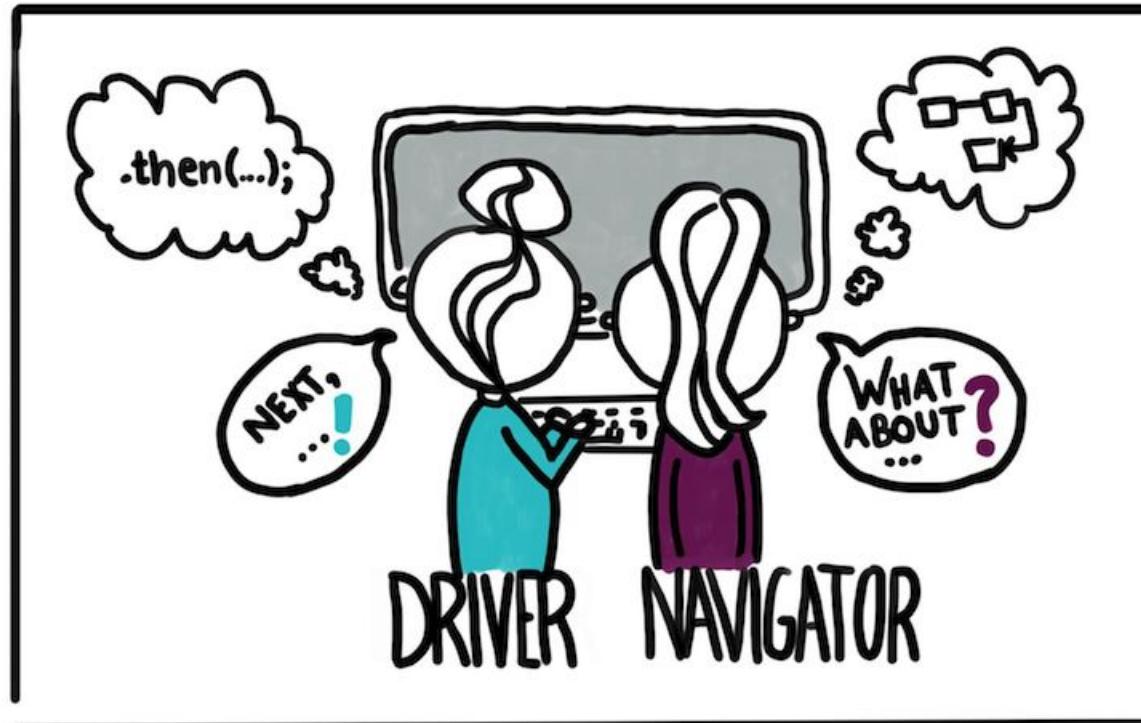


Keep repeating
until your
answers on the
training set are
good enough!

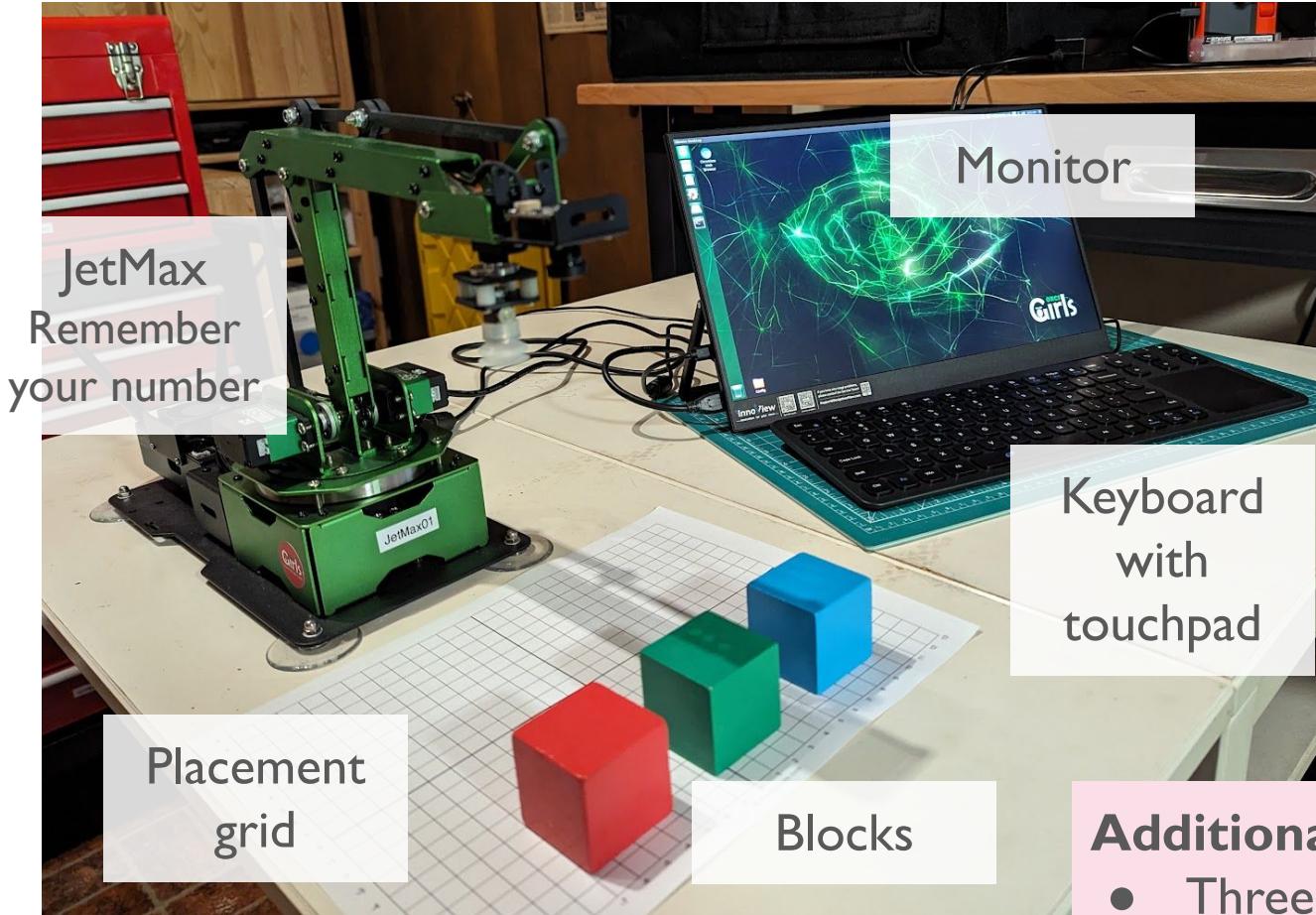


Remember *Driver Navigator* rules

- Make sure everyone gets to perform every task
- Take turns in setting up
- Help each other
- Listen and communicate
- Great teamwork leads to great progress and results
- Valuable skill to learn



Setting up the JetMax



Clear your workspace

Decide who picks up what

Volunteers will hand our materials

You need

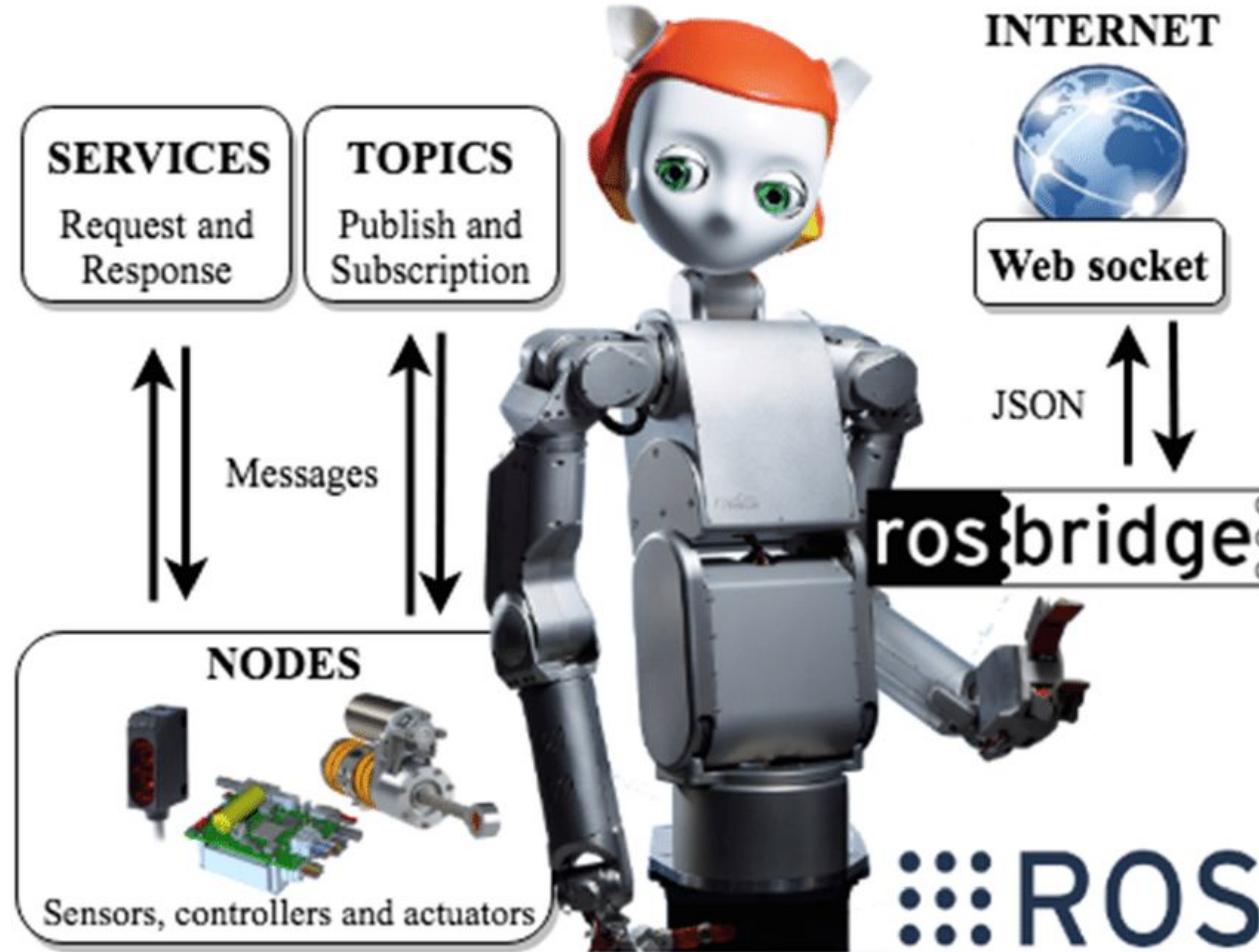
- JetMax robot arm
- **Use same robot number**
- Monitor
- Keyboard
- Placement grid
- Three color blocks
- Power supply JetMax
- Power supply monitor
- HDMI cable

Additional

- Three Apriltag blocks
- Recycling mat and cards



Robot Operating System



Calibrating camera - sucker offset

Open web browser to view camera image

Open a terminal to enter commands

We will do this together

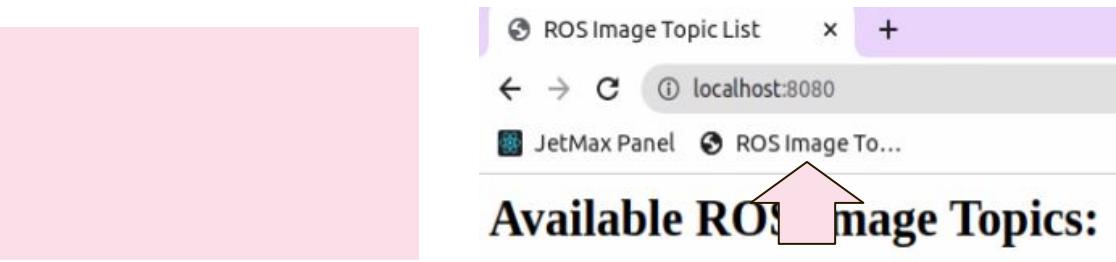
```
rosservice call /camera_cal/enter  
rosservice call /camera_cal/set_running "{data: True}"  
rosservice call /camera_cal/down
```

Position **Apriltag block 1** under middle of nozzle
with the 1 facing the arm

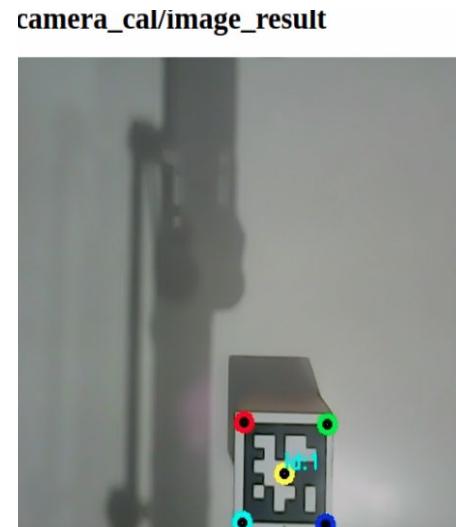
```
rosservice call /camera_cal/up
```

Wait for dots to align on the video feed in the browser

```
rosservice call /camera_cal/save  
rosservice call /camera_cal/exit
```



- Available ROS Image Topics:
- /usb_cam/
 - [image_rect \(Snapshot\)](#)
 - [image_rect_color \(Snapshot\)](#)
 - [image_mono \(Snapshot\)](#)
 - [image_raw \(Snapshot\)](#)
 - [image_color \(Snapshot\)](#)
 - /lab_config_manager/image_result (Snapshot)
 - /alphabetically/image_result (Snapshot)
 - /object_tracking/image_result (Snapshot)
 - /camera_cal/image_result (Snapshot)
 - /palletizing/image_result (Snapshot)
 - /waste_classification/image_result (Snapshot)
 - /color_sorting/image_result (Snapshot)



Block stacking demo

Open web browser to view camera image

Open a terminal to enter commands

Make sure you have space around the arm

```
rosservice call /palletizing/enter  
rosservice call /palletizing/set_running "{data: True}"
```

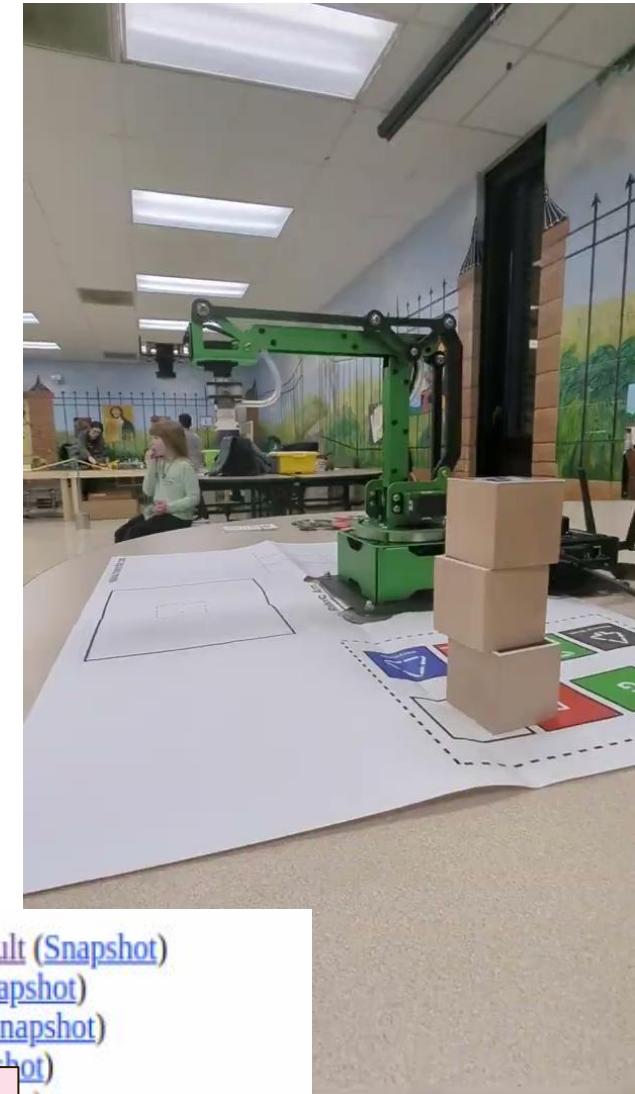
Place **Apriltag blocks** in front of the arm. It will pick them up and stack them. If there is more than one it will stack them in order

Once you are done, exit the demo with

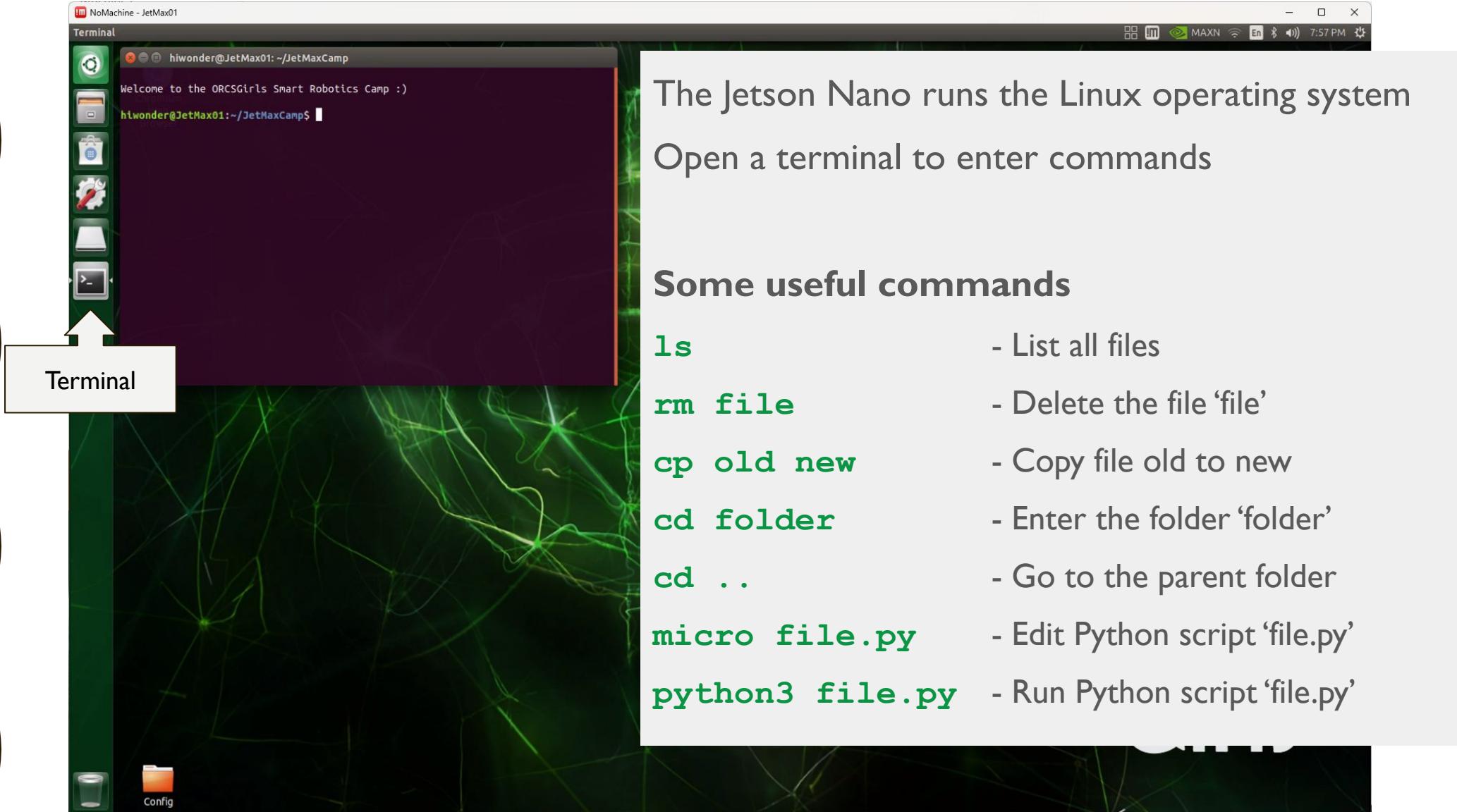
```
rosservice call /palletizing/exit
```

Take turns in starting and stopping the demo and placing blocks

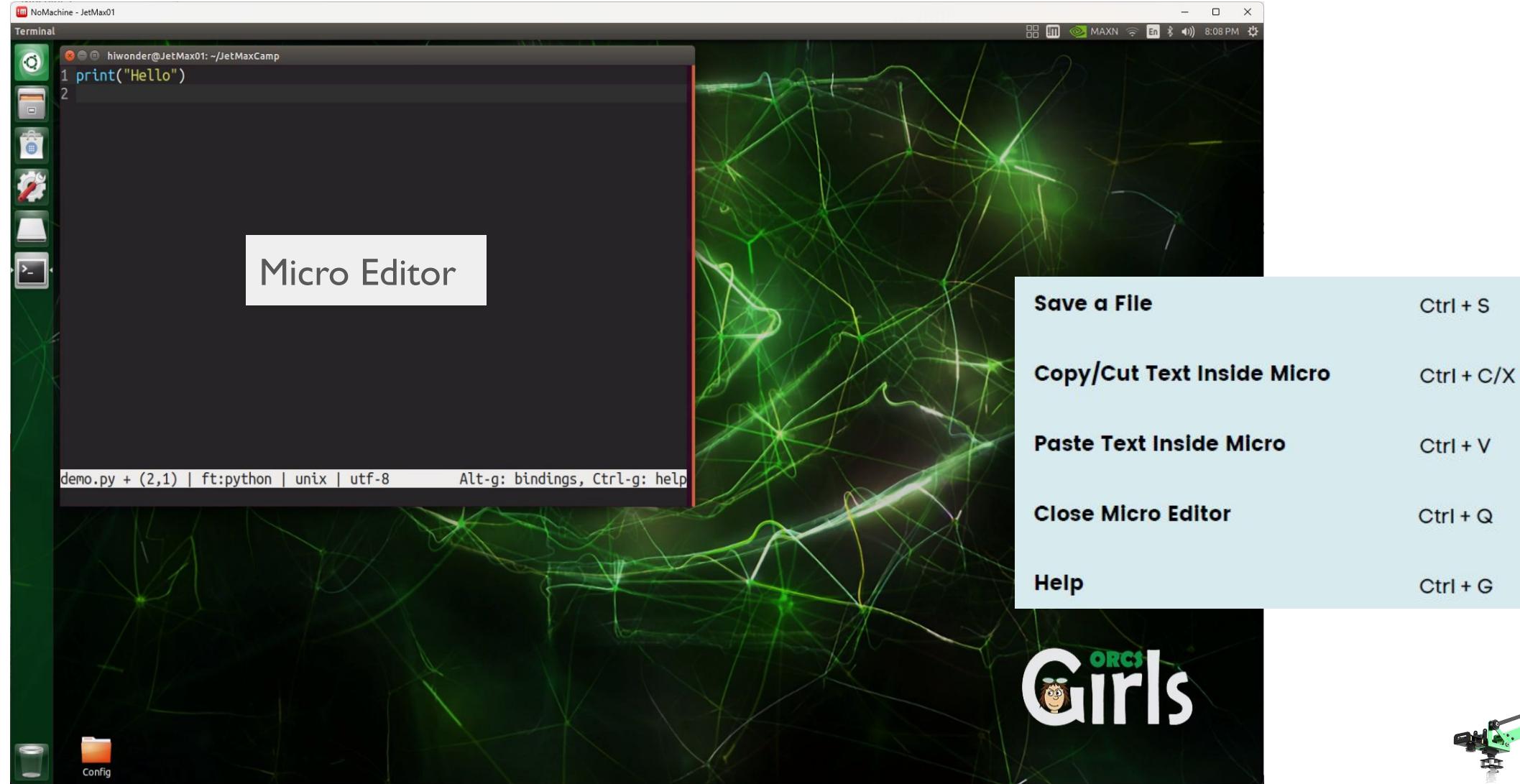
- [/lab_config_manager/image_result \(Snapshot\)](#)
- [/alphabetically/image_result \(Snapshot\)](#)
- [/object_tracking/image_result \(Snapshot\)](#)
- [/camera_cal/image_result \(Snapshot\)](#)
- [/palletizing/image_result \(Snapshot\)](#)
- [/waste_classification/image_result \(Snapshot\)](#)
- [/color_sorting/image_result \(Snapshot\)](#)



Using the terminal



Using the terminal



Controlling the robot in Python

```
import sys
from myjetmax import *

jetmax = myJetMax()
blocks = myColorBlocks()

# Pick up the block and lift and drop :)

(block_x, block_y) = blocks.position
print ("Found block at: ", block_x, block_y)

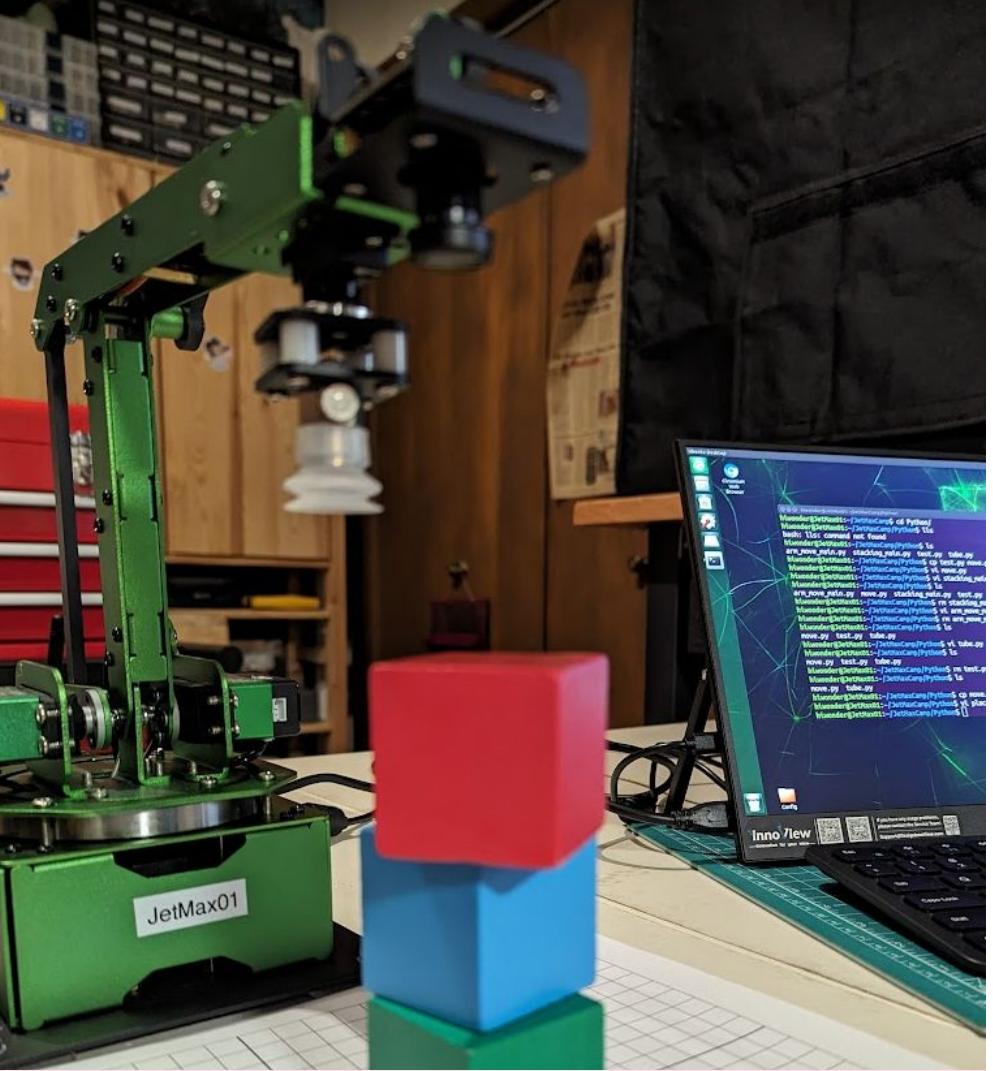
# Moving there and go down
jetmax.move_to(block_x, block_y, 150, 2.0)
jetmax.move_to(block_x, block_y, 20, 2.0)

# Sucker on and move up and drop
jetmax.suck()
jetmax.move_to(block_x, block_y, 40, 2.0)
jetmax.release()
```

JetMax Commands

- Open a terminal
- Go to the Python folder
- First run **python3**
myjetmax/publisher/color_position_publisher.py
- Put a color block within the view
- Run the **pick.py** program
- It should pick up the block
- This is the start for the coding activity





Make sure the grid mat does not move.

Remember to use Driver Navigator rules
when writing and testing the code!

JetMax Coding Challenge

- Select a lay down location
- Put a block in a random spot
- Write code to pick up the block and put it in the location

Extensions

- Use two blocks and stack

Showtime
Let's check out
all the
awesome code

For the last time

- Shut down the JetMax arm properly
- Wait for the screen to go blank
- Switch the arm off
- Carefully disconnect cables
- Bring all parts up and hand them to the volunteers



Group photo and graduation



Smart Robotics Camp



Big thank you to our
volunteers, partners and sponsors.



More classes coming soon ..



contact@orcsgirls.org

ORCSGirls

ORCSGirls