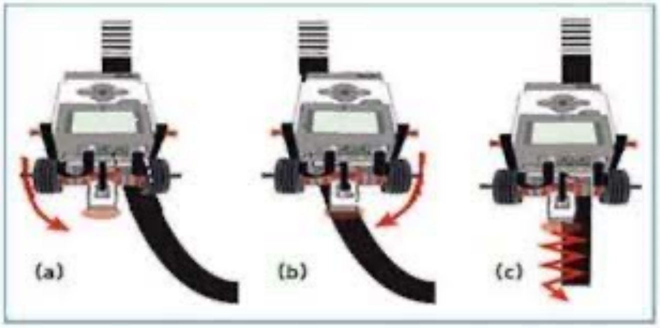
Lego Robotics Camp

Day 5: The Grand Finale

Review

- Yesterday we:
  - Learned about **functions**
  - Practiced using the color sensor to follow lines and detect colors
  - Got more practice with conditionals and loops
  - Designed simple algorithms for line following



Today's Plan

- Today we will:
  - Wrap it all up with one final grand challenge!
  - Prizes will be award to the fastest team **and** the team with the best code (best use of functions, good design, creativity, etc)

Camp Highlights: Day 1

Computational Thinking and Basic Movement

```
#!/usr/bin/env pybricks-micropython
from pybricks.hubs import EV3Brick
from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
InfraredSensor, UltrasonicSensor, GyroSensor)
from pybricks.parameters import Port, Stop, Direction, Button, Color
from pybricks.tools import wait, StopWatch, DataLog
from pybricks.robotics import DriveBase
from pybricks.media.ev3dev import SoundFile, Image, ImageFile

# Create your objects here.
ev3 = EV3Brick()

# Initialize the motors.
left_motor = Motor(Port.B)
right_motor = Motor(Port.C)

# Initialize the drive base.
robot = DriveBase(left_motor, right_motor,
        wheel_diameter=56, axle_track=114)
```

```
# Set eyes
ev3.screen.load_image(Image(ImageFile.NEUTRAL))

# Go forward and backwards for one meter.
robot.straight(500)
ev3.speaker.beep()

robot.straight(-500)
ev3.speaker.beep()

# Turn clockwise by 360 degrees and back again.
robot.turn(360)
ev3.speaker.beep()

robot.turn(-360)
ev3.speaker.say(“hello campers”)
```

Basic Robot Actions

- **Movement**
- `robot.straight(x)` – drive forward for x millimeters
- `robot.drive(x, y)` – drive forward at speed x and turn rate y
- `robot.turn(x)` – turn in place x degrees
- **Light**
- `ev3.light.on(color)` – turns light on to specific color (try Color.RED, Color.GREEN, Color.YELLOW)
- `ev3.light.off()` – turns off the light
- **Sound**
- `ev3.speaker.beep()` – beep speaker once
- `ev3.speaker.say("text")` – speak the text specified
- `ev3.speaker.play_notes(`*`notes`*`)` – plays a sequence of musical notes. For example, try ['C4/4', 'D4/4', 'E4/4', 'F4/4', 'G4/4']. This plays C, D, E, F, G as quarter notes (/4)
- **Screen**
- `ev3.screen.load_image(Image(ImageFile.NEUTRAL))` – sets "neutral" eyes. Also try ANGRY, DIZZY, SLEEPING, EVIL.

Camp Highlights: Day 2

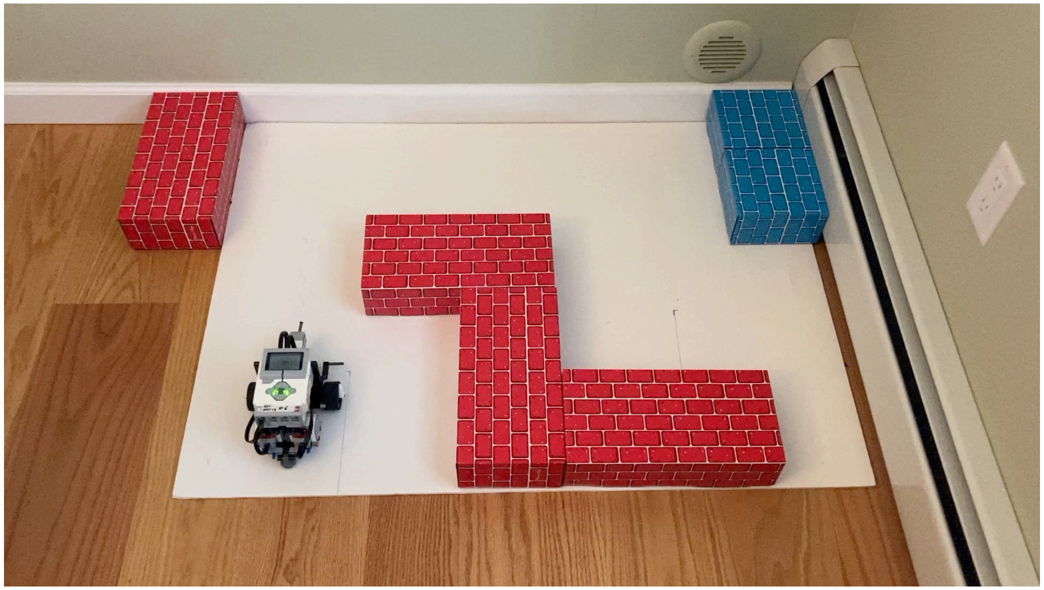
Conditionals and Loops

```
if condition1 is true:
      do action 1
elif condition2 is true:
      do action 2
else:
      do action 3
```

```
while condition1 is true:
      do action1
```

Camp Highlights: Day 3

```
#Initialize the touch sensor
touch_sensor = TouchSensor(Port.S1)
while not touch_sensor.pressed():
       robot.drive(200,0)
       wait(10)
robot.straight(-20)
robot.turn(90)
while not touch_sensor.pressed():
       robot.drive(200,0)
       wait(10)
robot.straight(-20)
robot.turn(90)
while not touch_sensor.pressed():
       robot.drive(200,0)
       wait(10)
robot.straight(-20)
robot.turn(-90)
robot.straight(300)
```



ultrasonic_maze

1. Go forward until you get close to a wall
2. Turn right
3. Go forward until you get close to a wall
4. Turn right
5. Go forward until you get close to a wall
6. Turn left
7. Go forward

```
us_sensor = UltrasonicSensor(Port.S1)
```

To just measure the distance:

```
dist = us_sensor.distance()


while us_sensor.distance() > 50:
    robot.drive(100,0)
    wait(10)
```

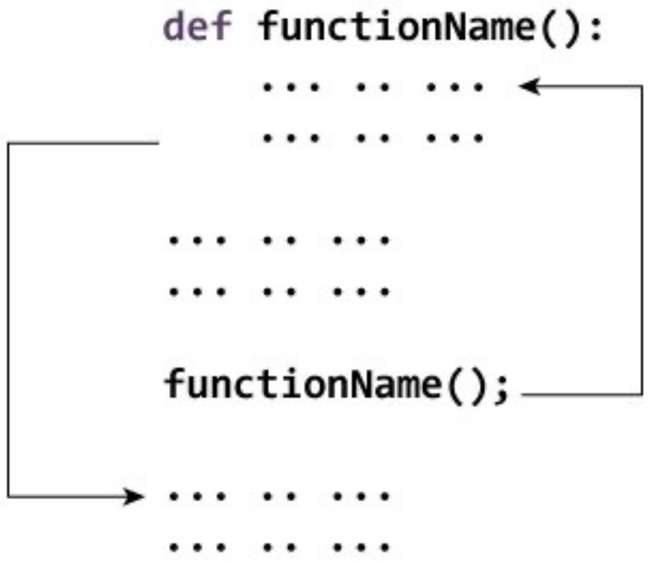
Camp Highlights: Day 4

Functions

- A function is a block of organized, reusable code that is used to perform a single, related action
- Allow for a high degree of code reusing
- Python gives you many built-in functions like print(), etc.
- You can also create your own functions (called *user-defined functions*)

User-defined Functions in Python

```
def function_name(parameters):
    statement(s)

#rest of your program
...

#now you can call your function
function_name(parameters)
```

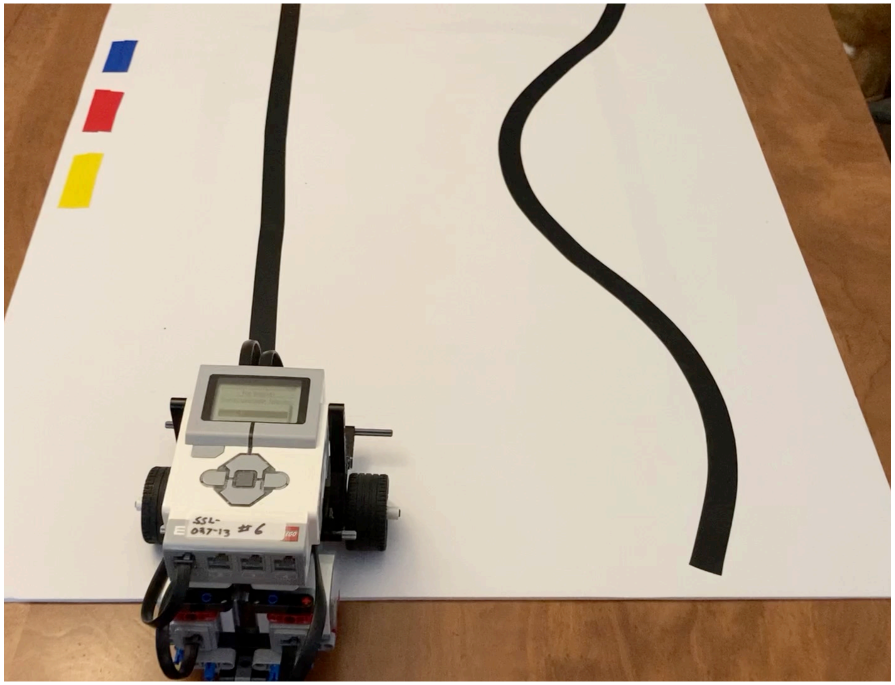


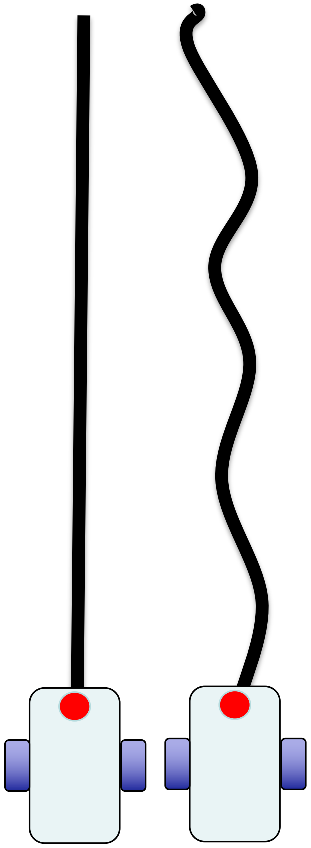
```
def functionName():
    ... .. ...
    ... .. ...

... .. ...
... .. ...

functionName();

... .. ...
... .. ...
```

color_tester

```
color_sensor = ColorSensor(Port.S1)

while True:
    if color_sensor.color() == Color.RED:
        ev3.speaker.say("red")
    elif color_sensor.color() == Color.BLUE:
        ev3.speaker.say("blue")
    elif color_sensor.color() == Color.YELLOW:
        ev3.speaker.say("yellow")
    elif color_sensor.color() == Color.GREEN:
        ev3.speaker.say("green")
    elif color_sensor.color() == Color.BLACK:
        ev3.speaker.say("black")
    elif color_sensor.color() == Color.WHITE:
        ev3.speaker.say("white")
    else:
        ev3.speaker.say("unknown")
    wait(1000)
```

Line Follower

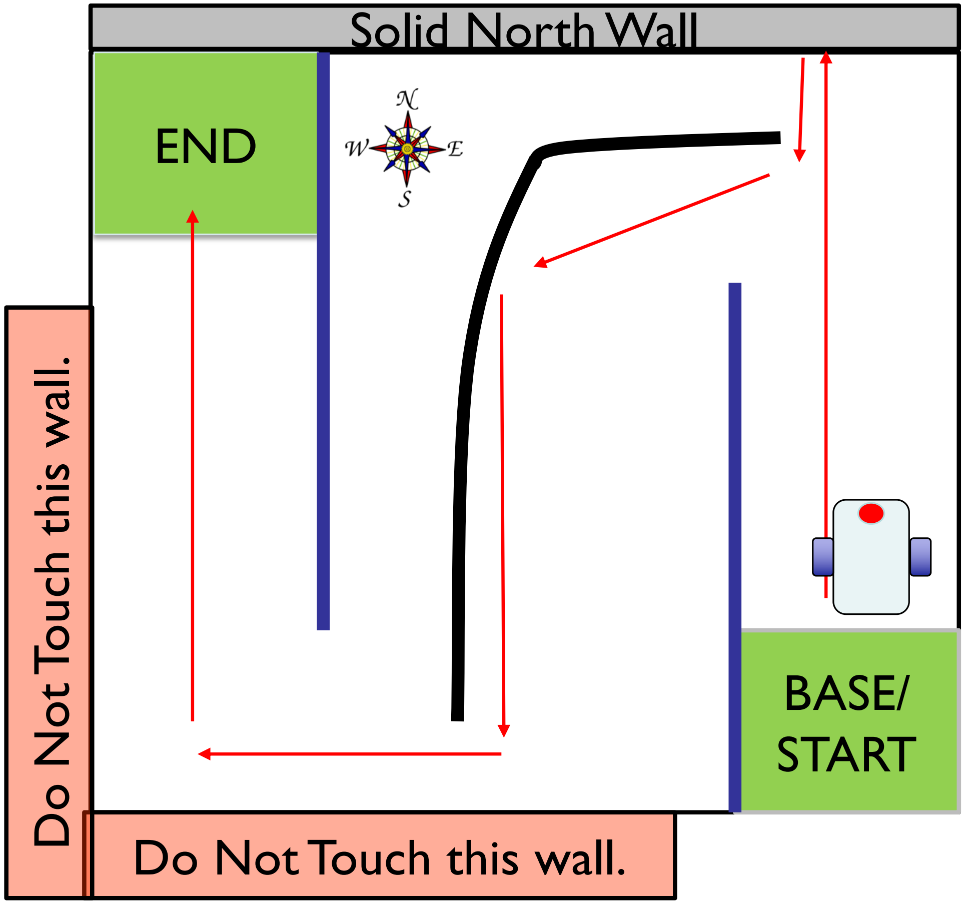




Grand Challenge

- Today you will complete your final challenge!
- You can use as many sensors as you want, but you will likely need at least 2 (probably 3)
- Your goal is to solve the maze as quickly as possible
- Work with a partner
- Plan your code first, then start coding
- Test often
- After lunch we'll break down our robots and clean up
- Good luck!

Grand Challenge



1) Go out of base
2) Go up to the North wall and touch it.
3) Back up turn and navigate through the passage (see black line)
4) Go to the South wall. Don't touch it!
5) Turn and go to the West wall. Don't touch it!
6) Turn and head to the END! You may stop anywhere in the region indicated.

USE 2-3 SENSORS! Don't cross blue lines.

Good luck!

- May the best robot win! ☺