Understanding the `is_near()` Function - A Guide for 7th Graders

What is the `is_near` Function?

Think of `is_near` as a question you can ask your robot: "Is there something close to me?"

```
def is_near(distance_threshold=100):
    distance = distance_sensor.distance(port.B)

    if distance == -1:
        print("Warning: Distance sensor not detecting anything")
        return False

    print ("Distance {:5.2f} cm {:6.2f} inches ".format(distance / 10, distance /
25.4))
    return distance < distance_threshold
```

This function returns either `True` (yes, something is near) or `False` (no, nothing is close).

Using Functions WITH Parentheses vs WITHOUT Parentheses

WITH Parentheses `is_near()` - "Ask the Question Now!"

When you write `is_near()` with parentheses, you're telling Python: **"Run this function right now and give me the answer!"**

```
# This CALLS the function immediately and gets True or False
if is_near():
    print("Something is close!")
    motor_pair.stop(motor_pair.PAIR_1)

# This also CALLS the function right now
answer = is_near()  # answer will be True or False
```

WITHOUT Parentheses `is_near` - "Here's the Recipe!"

When you write `is_near` without parentheses, you're giving Python the **recipe** for the function, not running it.

```
# This gives Python the "recipe" but doesn't run it yet
my_function = is_near
# Later, Python can use this recipe: my_function()
```

Real-World Analogy 🏠

Think of it like this:

- `is_near` without parentheses = giving someone a recipe card for cookies
- `is_near()` with parentheses = actually baking the cookies and getting fresh cookies

Wait Until vs Repeat Until Commands

1. **Wait Until** - Stop and Wait for Something to Happen

```
# Wait until something gets close
await runloop.until(is_near)  # NO parentheses - give the recipe!

# This means: "Keep checking is_near() over and over until it returns True"
```

Why no parentheses here? Because `runloop.until()` needs the recipe so it can:

1. Call `is_near()`
2. Check if it's True
3. If False, wait a moment and try again
4. Keep repeating until `is_near()` returns True

2. **Repeat Until** - Keep Doing Something Until a Condition is Met

```
# GOOD: Keep moving until something gets close
while not is_near():  # WITH parentheses - check right now!
    motor_pair.move(motor_pair.PAIR_1, 0)  # keep moving forward

# Stop when we exit the loop
motor_pair.stop(motor_pair.PAIR_1)
```

Why parentheses here? Because `while` needs to check the condition immediately each time through the loop.

What About Lambda Functions? 🤔

A **lambda** is like a mini-function that you create on the spot. It's useful when you need to modify how a function works.

Example 1: Different Distance Thresholds

```
# Wait until something is VERY close (5cm)
await runloop.until(lambda: is_near(50))  # 50mm = 5cm

# Wait until something is far away (20cm)
await runloop.until(lambda: is_near(200)) # 200mm = 20cm
```

Example 2: Opposite Conditions

```
# Wait until nothing is near (robot moved away from object)
await runloop.until(lambda: not is_near())

# Wait until EITHER pressed OR something is near
await runloop.until(lambda: is_pressed() or is_near())
```

Lambda Analogy

Think of lambda like customizing a pizza order:

- `is_near` = basic pizza recipe
- `lambda: is_near(50)` = "make that pizza but with extra cheese (50mm threshold)"
- `lambda: not is_near()` = "make the opposite of pizza (wait until nothing is near)"

Common Student Mistakes

Mistake 1: Wrong Parentheses with Wait Until

```
# WRONG - calls function once and uses that single result
await runloop.until(is_near())  # BAD!

# RIGHT - gives the recipe so it can be called repeatedly
await runloop.until(is_near)    # GOOD!
```

Mistake 2: Forgetting Parentheses in While Loops

```
# WRONG - this doesn't call the function
while not is_near:  # BAD!

# RIGHT - this calls the function each time
while not is_near():  # GOOD!
```

Quick Reference Card

Command Type	Code Example	Parentheses?	Why?
Wait Until	`await runloop.until(is_near)`	NO	Give recipe to be called repeatedly
Repeat Until	`while not is_near():`	YES	Check condition each loop

Command Type	Code Example	Parentheses?	Why?
If Statement	`if is_near():`	YES	Check condition right now
Lambda Custom	`lambda: is_near(50)`	Inside lambda	Create custom version

Practice Challenges 🎯

Try writing code for these scenarios:

1. **Wait until pressed, then move until near something**
2. **Keep moving in a square until you detect red color**
3. **Wait until something is very close (3cm) using lambda**
4. **Move forward until either pressed OR something is near**

Remember: Functions are tools - sometimes you use the tool (`is_near()`), sometimes you just hand someone the tool (`is_near`) so they can use it when needed!