

# Lab 2: Turtle Pirate Adventure Part 1



- 
1.  **Worth:** 0%
  2.  **Due:** TBD (September 18 - 30)
  3.  **Late submissions:** N/A
  4.  **Submission:** None, progress made in class.

This in-class activity **is not graded**, but:

- doing it will teach you skills needed for Assignment 2
- there will be a Quiz on the content we learn in this activity

## Objectives

---

We will learn:

- how to create and use variables
- how to convert variable data types
- how to perform mathematical operations to solve useful problems
- how to use python `input/output` functions
- how to use format-strings

## Background

---

You (a programmer) have been hired by Captain Threepwood, a mighty pirate, to bury their pirate treasure at a location known only to you.

To keep this secret safe, you do not draw a map, but instead you keep track of the coordinates of the buried treasure by storing their numbers in variables in your code (`x_treasure` , `y_treasure` ), and writing a program that instructs pirates what steps to

take in order to get to the treasure from any given starting location.

This crew of pirates crew is not very original; they always take steps in a *clockwise* order of cardinal directions:

- North
- then East
- finally West
- then South

Pirates locate their treasure by going from some starting position and then taking a series of steps in various directions. Example: start at (3, 4) on the map, take 5 steps North, then 3 steps East, etc...



The only unit of measurement they seem to care about is “steps”. Be aware that Pirates do not like to read more than 3 decimal places.

## Instructions

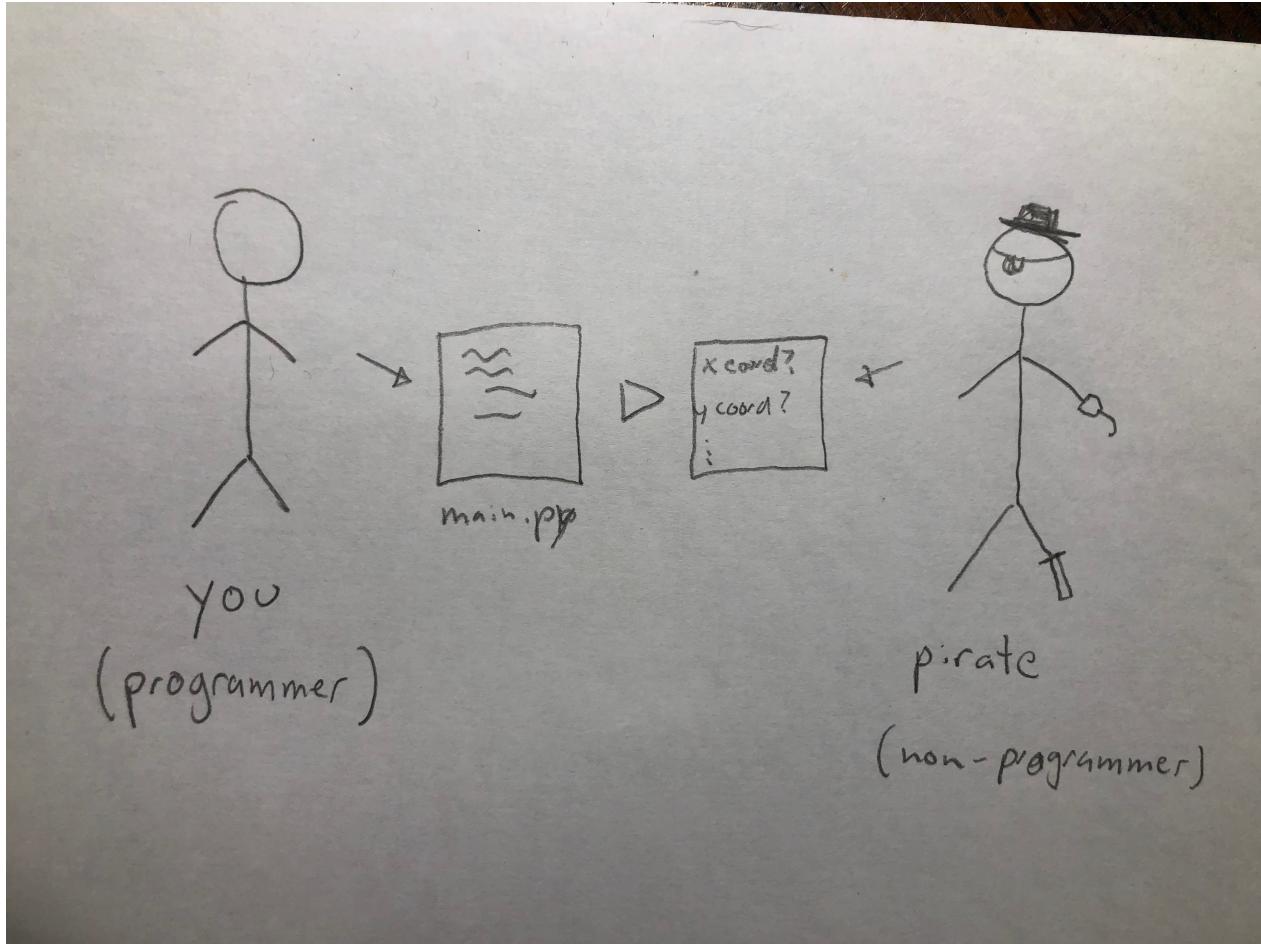
---

Write a python program that:

- Allows the pirates tell you where they buried their treasure as starting point and a

sequence of steps (the user input).

- Tells the pirates (the program output):
  - what are the x,y-coordinates of their treasure
  - how far they've walked
  - what is the straight line distance from the start to the treasure



## Part 1: Treasure Hunting

Write a python program that:

1. Defines a hiding place for the treasure (choose a random `x_treasure` , `y_treasure` that only you will know, the map goes from [ -300 , +300 ] on each direction)
2. When Captain Toothache's pirates arrive on the island (the player of the game starts the game) they will be asked what position on the map that they want to start
3. The program will ask them how many steps they want to take before turning right by 90°. Keep track of the pirate's position as they move.

4. Since you, the programmer, have not been paid enough by Captain Threepwood, you will inform Captain Toothache's pirate how far they are from the treasure everytime they have walked a certain distance. To do this, you must calculate the distance that the pirates are away from the treasure, and this distance will be printed to the console ( print )
5. Captain Toothache's pirates will have EXACTLY 8 opportunities to move.

Your program must respect the **variable naming conventions** seen in class.

Your program must use these **assignment operators**: = , += , -= , etc.

## Step 1: Get the user to tell you their starting position

Use the `turtle.numinput` to ask the pirates for:

- the starting x-coordinate position. you can assume that it's a whole number (`x_pirate`).
- b. the starting y-coordinate position. You can assume that it's a whole number (`y_pirate`).



- Calculate the distance between the pirate and the treasure:

- $distance = \sqrt{(x\_pirate - x\_treasure)^2 + (y\_pirate - y\_treasure)^2}$

- Print the distance to the terminal window

You are currently 24.32 units away from the treasure

*Use formatted strings!!! (no more than 2 decimal points)*

## Move the turtle

- Note that when the game starts, the turtle is facing west.
- Move the turtle to its new starting position based on the input from the user
- Make sure the turtle is facing north when you are finished.

## Step 2: Moving

Using the `turtle.numinput` function, prompt the user for the distance that they want to move.

**After moving the turtle,**

- have the turtle turn right by 90 degrees
- Update the current position of the pirate (`x_pirate` , `y_pirate`) after each movement
  - use the appropriate assignment operator `+=` or `-=`
- Calculate the distance between the pirate and the treasure:
  - $distance = \sqrt{(x\_pirate - x\_treasure)^2 + (y\_pirate - y\_treasure)^2}$
- Print the distance to the terminal window

You are currently 24.32 units away from the treasure

*Use formatted strings!!! (no more than 2 decimal points)*

## Step 3: Final result.

In the end, display how many steps the pirates traveled in total? How far are they from the treasure at the end?

For example:

You are out of moves.  
You have walked 14 steps.  
You are 2.82 steps from the treasure.

## Part 2: Did the pirate find the treasure?

---

In this part, you must use a simple `if` statement to verify if the pirates final coordinates correspond to the coordinates of the treasure. If it's the case, display "Hooray! You found the treasure"

For example:

```
You have walked 16 steps.  
You are 0.001 steps from the treasure.  
Hooray! You found the treasure!
```

Hint: You must use the comparison operator `==` and the logical operator `and`. Here is an example:

*I am checking if `day_of_week` is equal to "Saturday" and that the `month` is equal to "July", if it's the case I am printing "Hooray! It's a Saturday of July"*

```
if day_of_week == "Saturday" and month == "July":  
    print("Hooray! It's a Saturday of July")
```

## Step 1: Did the pirates find your treasure?

- After the pirate has completed all their walking, did they find the treasure?
  - If yes, print `CONGRATULATIONS, You found the treasure` on the terminal
  - If no, print `No treasure found, but try again sometime soon`

## Part 3: Testing with a Pirate

Pirates can be morally corrupt and might try to break your program! You should *test* your program with a few examples to make sure your algorithm is correct. You can assume that the user will only type integer numbers into your program.

**It's always best to try it out on real pirates. Swap seats with the person next to you and try running each others' program! You are now the pirate and playing someone else's game. Your goal should be to find a bug in their program** 

## Emoji Licenses

---

Pirate Emoji:

<https://moremoji.allezsoyez.com/>

<a href="https://moremoji.allezsoyez.com/">https://moremoji.allezsoyez.com/</a>	
Author	alleZSoyez, derivative mork of <a href="#">Twemoji</a>

Treasure Emoji:

[png image from pngtree.com/](#)