
The Joy of Programming
Haven't started EX07 yet? Complete "Upgrade Trailhead" in Step 1 of setup to follow along today.
COMP110 - CL23
2024/04/30

Code Writing Practice

- Write a class with the following characteristics:
- The class' name is Staff.
- Every Staff object has two attributes: name (string) and is_cs (bool).
- You should be able to construct a Staff object with a constructor that has parameters to initialize each attribute
- You should implement any methods necessary to implement the following behavior:

```
>>> prof: Staff = Staff("Kris", True)
>>> print(prof.greet())
Hello, I'm Kris in CS
>>> dr: Staff = Staff("Mara", False)
>>> print(dr.greet())
Hello, I'm Mara NOT in CS
```

- Write a class with the following characteristics:
- The class' name is Staff.
- Every Staff object has two attributes: name (string) and is_cs (bool).
- You should be able to construct a Staff object with a constructor that has parameters to initialize each attribute
- You should implement any methods necessary to implement the following behavior:

```
>>> prof: Staff = Staff("Kris",
True)
>>> print(prof.greet())
Hello, I'm Kris in CS
>>> dr: Staff = Staff("Mara", False)
>>> print(dr.greet())
Hello, I'm Mara NOT in CS
```

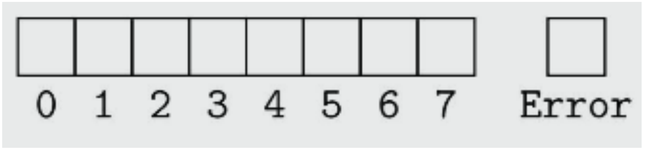
Question 5: Loops In this series of questions, you will trace code that modifies a boolean `list a`. You will respond beneath each code listing by *completely shading in the squares of items whose value is assigned* ***`True`***. If an error occurs during the evaluation of the loop, fill in the `Error` box and stop evaluating. If any item's value was assigned `True` prior to the error, keep its value shaded in.

You can assume `a` is initialized with *8* ***`False`*** *elements*, as shown below, and that each question is independent of the next.

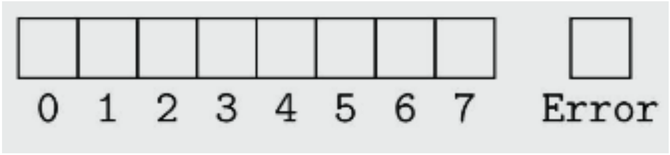
You can assume `a` is initialized with *8* ***`False`*** *elements*, as shown below, and that each question is independent of the next.

```
f: bool = False
a: list[bool] = [f, f, f, f, f, f, f, f]
```

```
4    i: int = 0
5    while i < len(a):
6        if i % 2 == 1 and i >= 3:
7            a[i] = True
8        i += 1
```




```
4    i: int = 0
5    while i <= 8:
6        if i % 2 == 0:
7            a[i] = True
8        i += 1
```



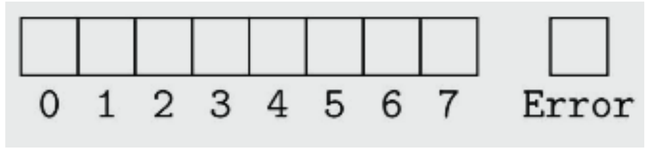
Question 5: Loops In this series of questions, you will trace code that modifies a boolean `list a`. You will respond beneath each code listing by *completely shading in the squares of items whose value is assigned* ***`True`***. If an error occurs during the evaluation of the loop, fill in the `Error` box and stop evaluating. If any item's value was assigned `True` prior to the error, keep its value shaded in.

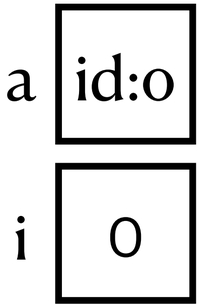
You can assume `a` is initialized with *`8 False elements`*, as shown below, and that each question is independent of the next.

You can assume `a` is initialized with *`8 False elements`*, as shown below, and that each question is independent of the next.

```
f: bool = False
a: list[bool] = [f, f, f, f, f, f, f, f]
```

```
4   i: int = 0
5   while i < len(a):
6       if i % 2 == 1 and i >= 3:
7           a[i] = True
8       i += 1
```





id:0

list[bool]	
0	False
1	False
2	False
3	False
4	False
5	False
6	False
7	False

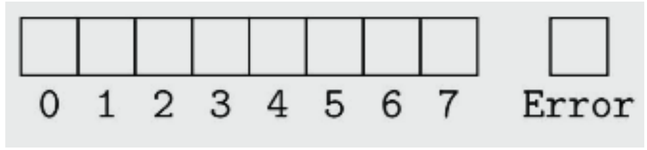
Question 5: Loops In this series of questions, you will trace code that modifies a boolean `list a`. You will respond beneath each code listing by *completely shading in the squares of items whose value is assigned* ***`True`***. If an error occurs during the evaluation of the loop, fill in the `Error` box and stop evaluating. If any item's value was assigned `True` prior to the error, keep its value shaded in.

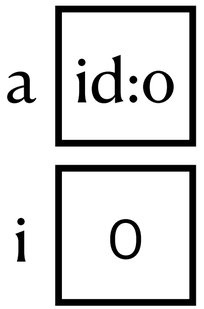
You can assume `a` is initialized with *`8 False elements`*, as shown below, and that each question is independent of the next.

You can assume `a` is initialized with *`8 False elements`*, as shown below, and that each question is independent of the next.

```
f: bool = False
a: list[bool] = [f, f, f, f, f, f, f, f]
```

```
4   i: int = 0
5   while i <= 8:
6       if i % 2 == 0:
7           a[i] = True
8       i += 1
```





id:0

list[bool]	
0	False
1	False
2	False
3	False
4	False
5	False
6	False
7	False

Code-Along: Turtle Graphics

1. *This will only work if you've started EX07 and updated Trailhead to 0.2.0. See EX07, Step 1, if not.*
2. In VSCode, create a directory in lectures named **cl23** and blank file in the directory named **`turtle.py`**
3. Open a browser to this link, copy and paste its contents into turtle.py:

   -go.unc.edu/turtle

4. Create a blank, new file named **`flower.py`** and type in the contents right

```
"""Turtle Art!"""

from .turtle import Turtle
from math import pi

__template__ = "https://24s.comp110.com/static/turtle"


def main() -> Turtle:
    t: Turtle = Turtle()
    t.setSpeed(0.25)

    t.left(pi / 2.0)
    t.forward(150)

    t.left(pi / 2.0)
    t.forward(148)

    return t
```

Practice Looping: Draw a Spiral

- Write a while loop (don't forget a counter variable!) that, inside of the loop:
  - Turns the Turtle t left by pi / 2.0
  - Moves the Turtle t forward by 150, 148, 146, and so on, until not moving forward
  - Update your variable so that it moves toward the loop's terminating condition

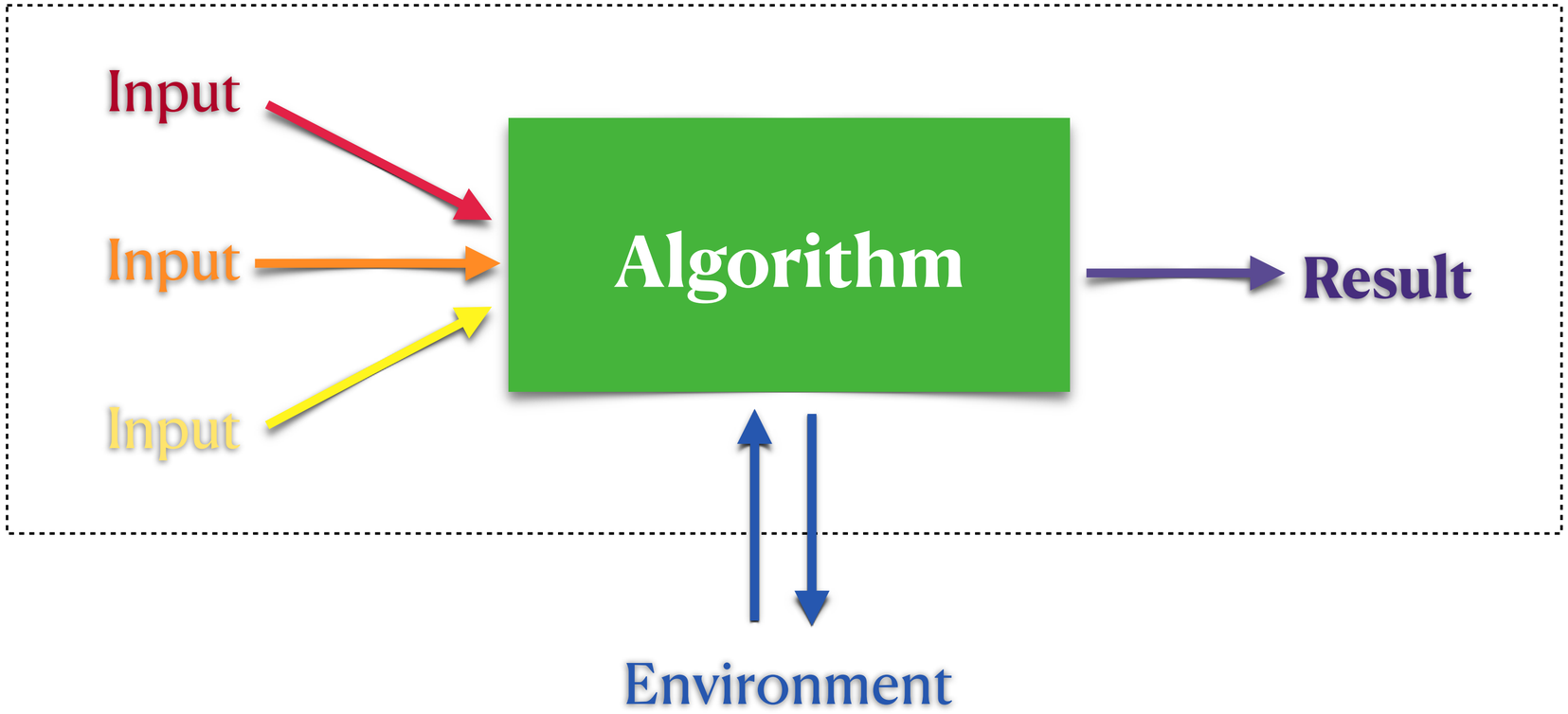
- You should see a spiral being drawn once correct!
- Try increasing the speed to 10 or 100 once you have it working. Additionally, try playing with the angle left the turtle is playing for different spirals.

Code-Along #2: Turtle Graphics

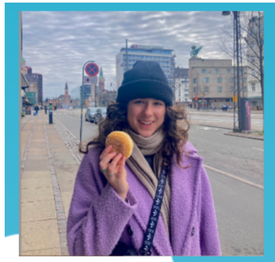
1. In the lecture cl23 directory, Create a blank, new file named **`happy_trees.py`** and type in the contents right
2. Once you have it, try clicking around your canvas and planting some sad, little trees.

```
"""Some happy, little trees!"""

from .turtle import Turtle
from math import pi
from random import random

__template__ = "https://24s.comp110.com/static/turtle"

DEGREE: float = -pi / 180.0


def main() -> None: ...


def click(x: float, y: float) -> Turtle:
    t: Turtle = Turtle()
    t.moveTo(x, y)
    t.turnTo(90 * DEGREE)
    t.forward(100)
    return t
```

The Fundamental Pattern

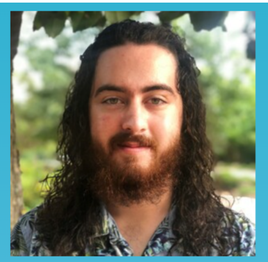


Special Thanks To...

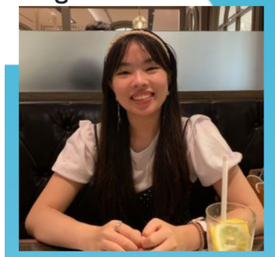

Abigail Kessel


Alicia Bao


Bailey DeSouza


Benjamin Eldridge


Bernie Chen


Catherine Huang


Chloe Lin


David Karash


Dika Manne


Gabriela Barros


Jessica Bring


Madeleine Genova


Madi Drummonds


Matt Kolsch


Meghan Sun


Olivia Wen


Ryan Krasinski


Sadie Amato


Somer Lillard

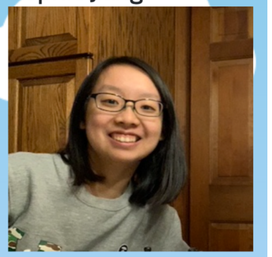

Sophie Jiang


Sritan Vemuru


Sunny Wang


Vivian Deng


Vrinda Desai


Yueen Ma

Apply to be a UTA

1. csxl.unc.edu
2. Open sidebar and login
3. Academics > Apply


CAMP110
Thank YOU for a Great Semester!