



# Computer Programming Summer Interest Group

**Session 4 / Iteration**

7/9/2025

# Numbers 7:12-...

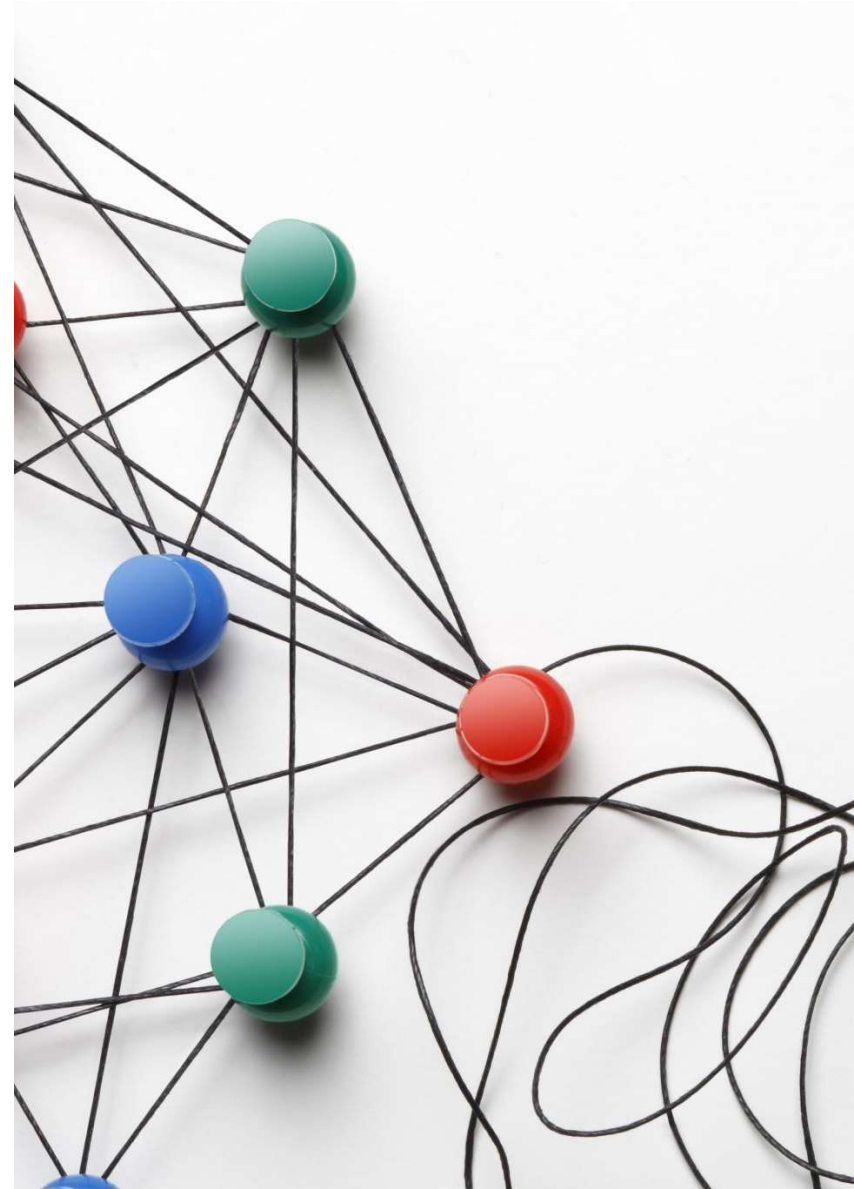
- The one who brought his offering on the first day was Nahshon son of Amminadab of the tribe of Judah. His offering was
  - one silver plate weighing a hundred and thirty shekels and
  - one silver sprinkling bowl weighing seventy shekels, both according to the sanctuary shekel,
  - each filled with the finest flour mixed with olive oil as a grain offering;
  - one gold dish weighing ten shekels, filled with incense;
  - one young bull,
  - one ram
  - and one male lamb a year old for a burnt offering;
  - one male goat for a sin offering
  - and two oxen
  - five rams
  - five male goats
  - and five male lambs a year old to be sacrificed as a fellowship offering.
- This was the offering of Nahshon son of Amminadab.

# Proverbs 26:11

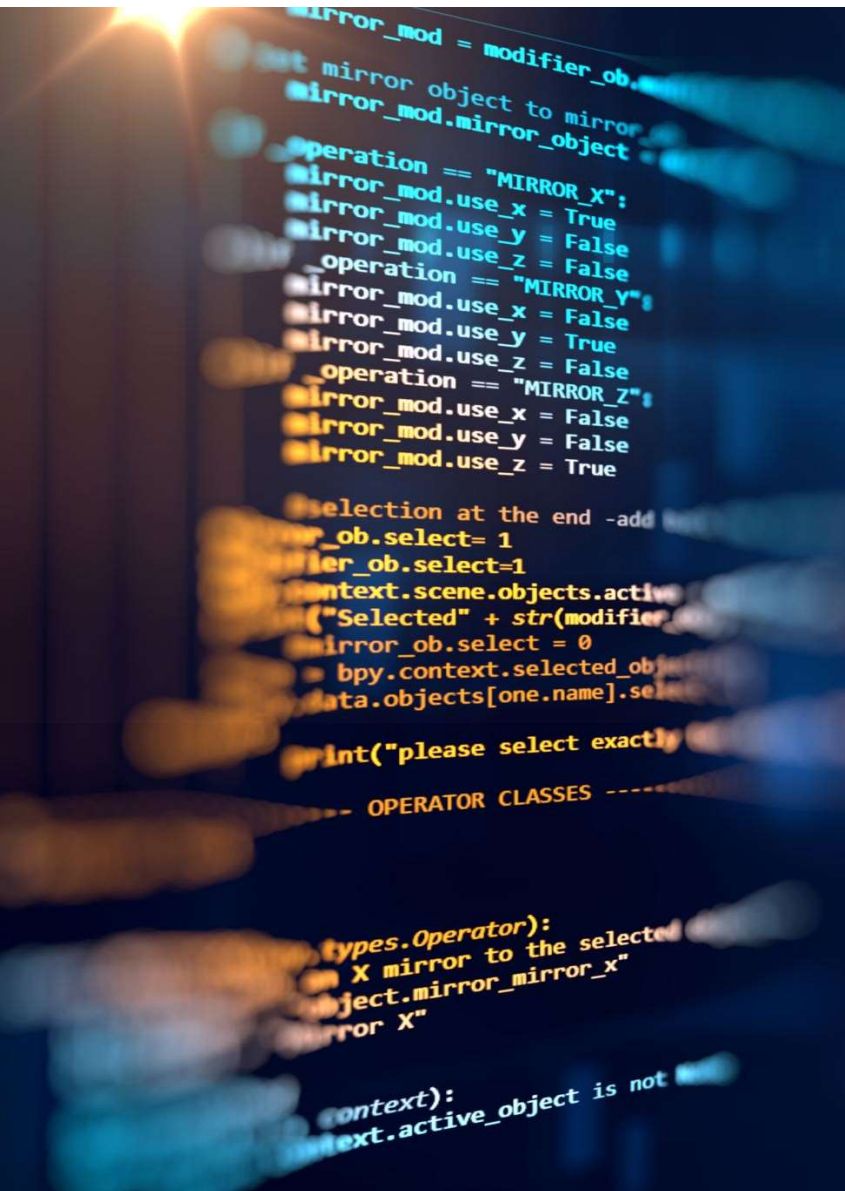
As a dog returns to its vomit, so fools repeat their folly.

# Objectives

- By the end of this session, you will be able to:
  - Understand the concept and purpose of loops in programming
  - Use **for** loops to iterate over sequences like lists, strings, and ranges
  - Use **while** loops to repeat actions while a condition is true
  - Identify common errors such as infinite loops
  - Choose the appropriate loop type for a given task



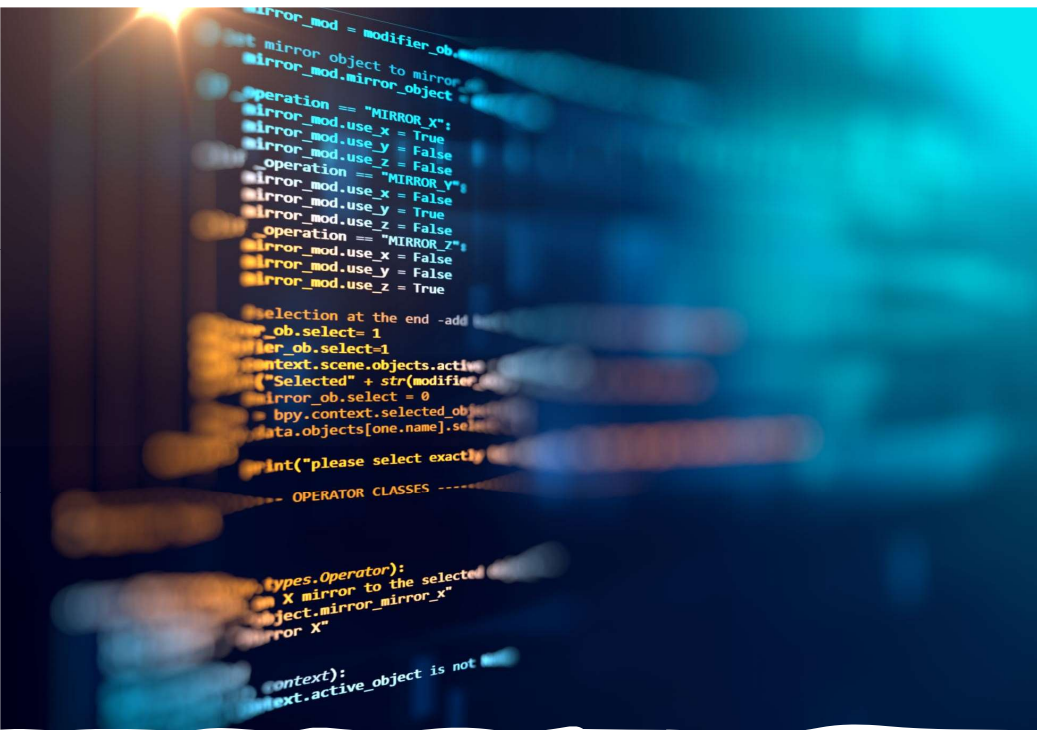




# Sequential Flow

- No decisions





# Conditional Flow

- Make a decision to do ... one thing ... or some other thing



# Iteration Control Flow

- Repeat some activity
  - **For** a collection of values
  - **While** some condition is satisfied

```
mirror object to mirror
mirror_mod.mirror_object

operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
    operation == "MIRROR_Y":
        mirror_mod.use_x = False
        mirror_mod.use_y = True
        mirror_mod.use_z = False
    operation == "MIRROR_Z":
        mirror_mod.use_x = False
        mirror_mod.use_y = False
        mirror_mod.use_z = True

selection at the end - add
ob.select-1
for ob.select-1
context.scene.objects.active
("Selected" + str(modifier
mirror_ob.select = 0
bpy.context.selected_ob
data.objects[one.name].sel
Int("please select exactly
OPERATOR CLASSES -----

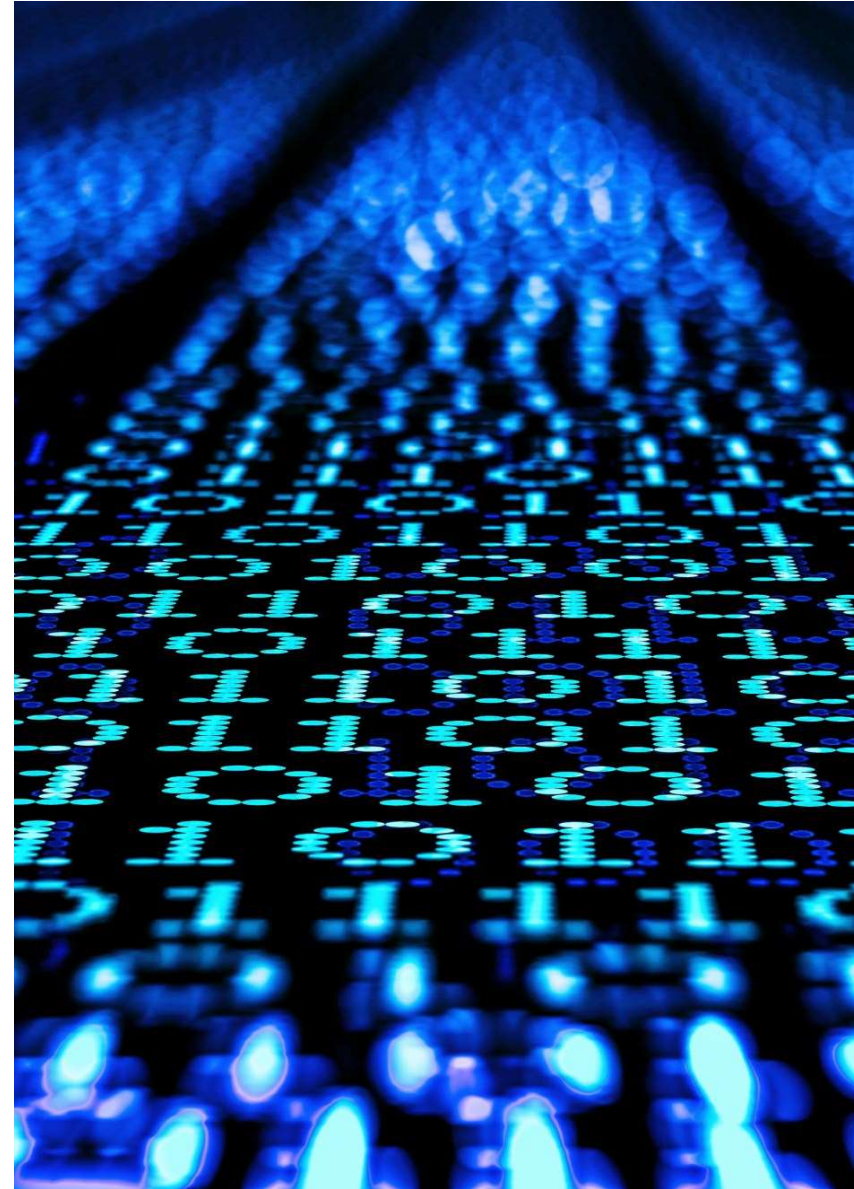
types.Operator):
    X mirror to the selected
    object.mirror_mirror_x"
    mirror X"

def():
    object is not
```



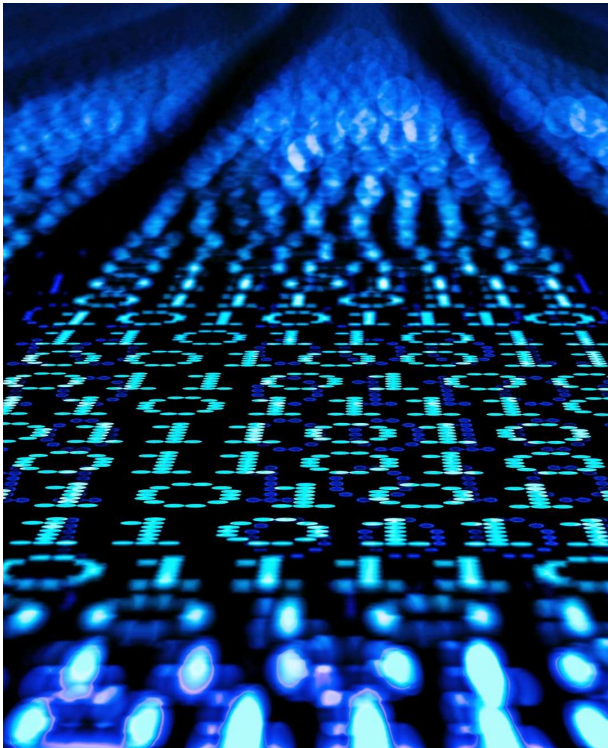
# Loops

- Loops allow a programmer to write code which is executed 0, 1 or many times
- In Python, the number of times the code is executed is determined when the program is executed
- Examples in real life:
  - Wash, Rinse, Repeat (shampoo instructions)
  - Whip the cream until it becomes light and fluffy
- Python supports two types of loops
  - **for** loops – Execute some code “for” each value provided
  - **while** loops – Execute some code “while” some condition is True





# for Loops with ranges



```
for i in range(5):  
    print("Hello")
```

```
for i in range(1, 6):  
    print("Step:", i)
```



# for Loops for value collections

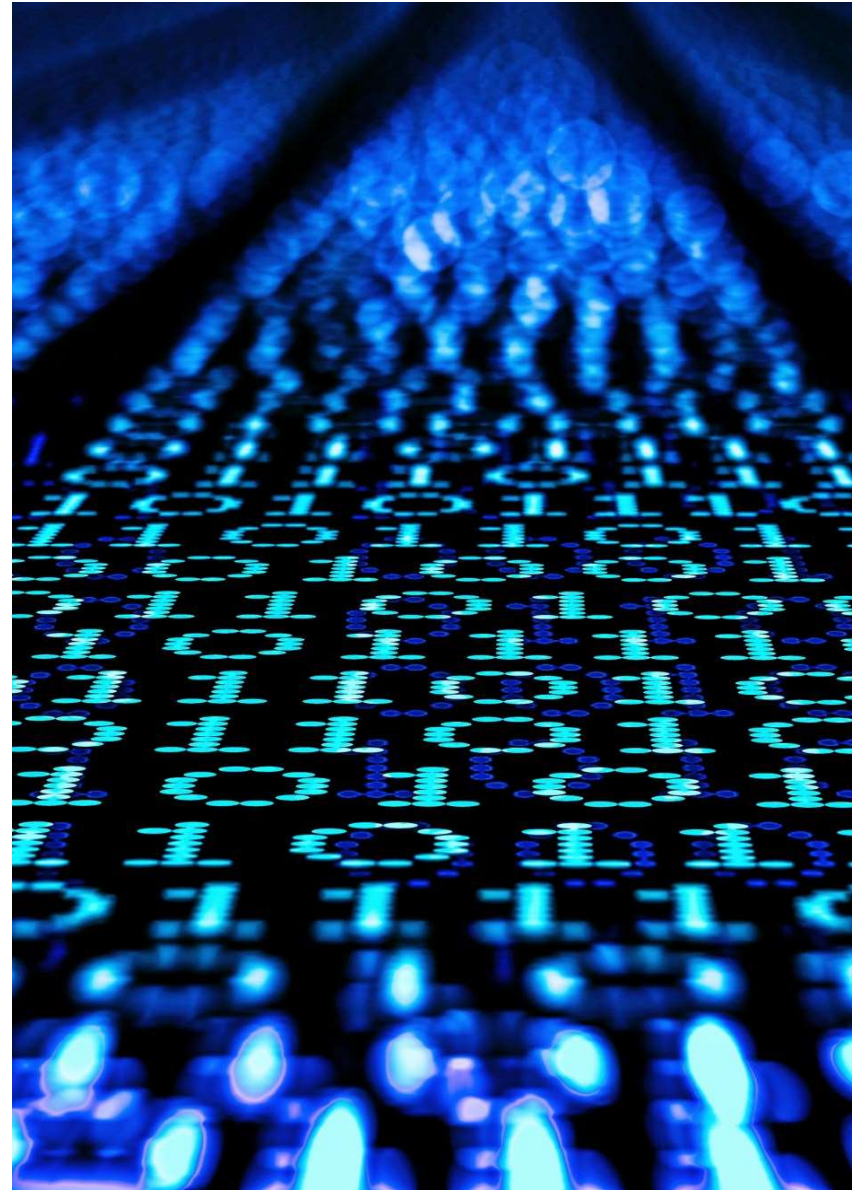
```
for letter in "Python":  
    print(letter)
```

```
fruits = ["apple", "banana", "cherry"]
```

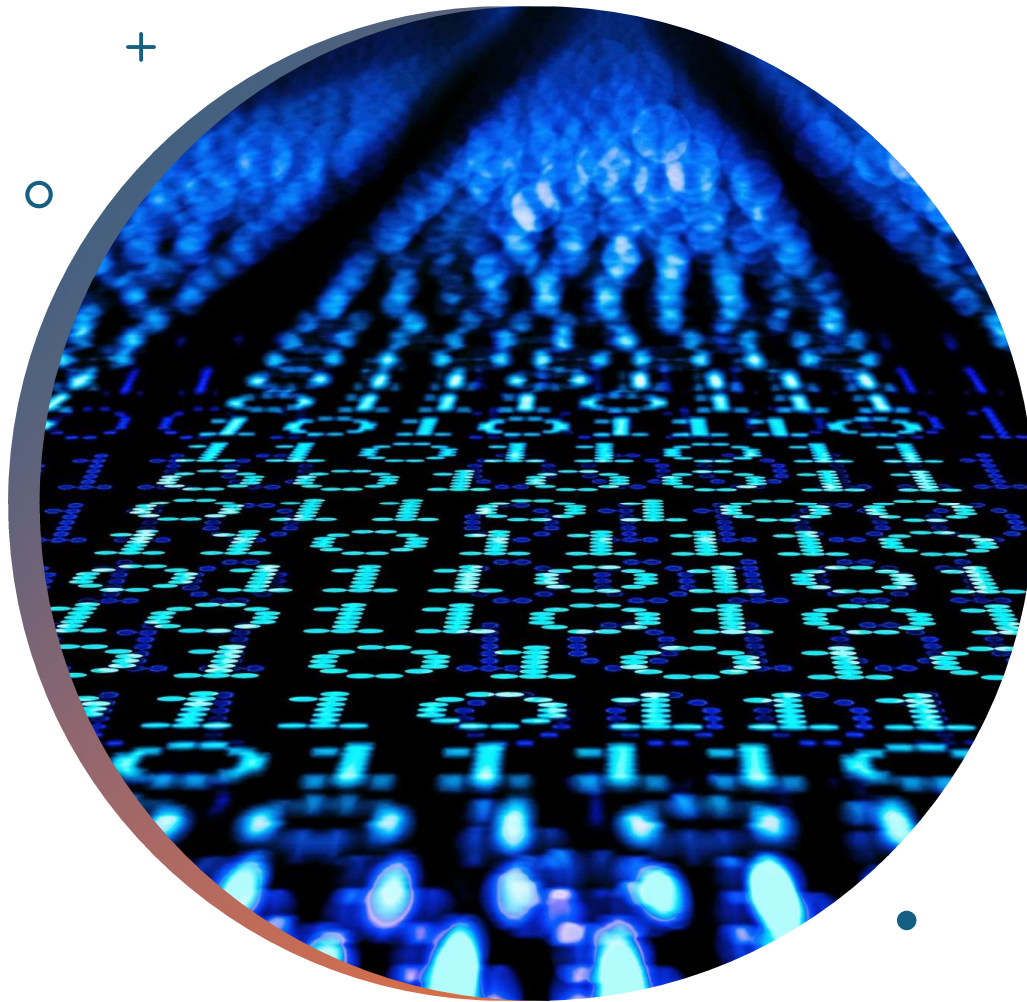
```
for fruit in fruits:  
    print(fruit)
```

# while Loops

```
count = 0  
while count < 5:  
    print(count)  
    count += 1
```





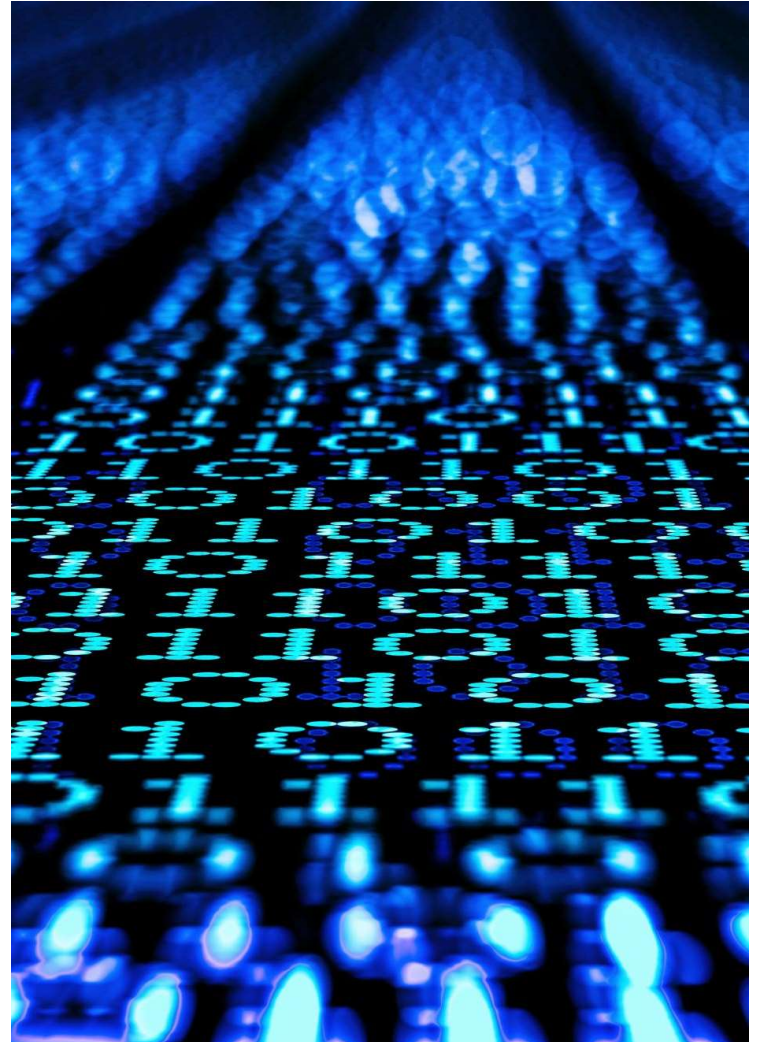


# break and continue

- **break** – Stops execution of the loop
- **continue** – Stops the current iteration of the loop and starts the next one

# break and continue

```
count = 0
while True:
    count += 1
    if count % 2 == 0:
        continue
    print(count)
    if count >= 10:
        break
```





# Activity

- Create an interactive number guessing game:
  - Set an initial secret value to a variable
  - Prompt the user for a guess of the secret value
  - While the guess is not the same as the secret value, tell the user to try again and read another value from the user
- If you get that done, extend the above to include:
  - Count the number of attempts to guess the correct number
  - Tell the user if the number they provided was greater than or less than the secret number
  - Limit the number of tries the user gets to try values
- If you get all that done:
  - Write a program that writes the first 10 square values (i.e., 1, 4, 9, 16, 25, ...)