# Week 4: Iteration –

- While loop vs for loop
- Use of continue and break

# While loop vs for loop

- Which is the best solution?

```
i = 1
while i < 10 :
    print(i)
    i = i + 1
```

---

```
for i in range(1, 10) :
    print(i)
```

# ANSWER - While loop vs for loop

- Both produce the same result.
  - The **while loop** has a condition and it runs until the condition is false (condition controlled).
  - Use the **for loop** (count controlled) when you know how many items the loop should execute.

```
for i in range(1, 10) :
    print(i)
```

# Breaking the loop

- If you need to break out of a loop there are two important keywords: continue and break.
- **Continue**
  - when it gets executed the loop will stop the current iteration where it is without executing anything after the continue statement and go on to the next iteration of the loop.
- **Break**
  - when it gets executed it will break out of the loop and continue executing from the first instruction after the loop.

# break & continue

- **break** What will this example print?

```
count = 0
while True:
    print(count)
    count += 1
    if count >= 5:
        break
```

- **continue** What will this print?

```
for x in range(10):
    if x % 2 == 0:
        continue
    print(x)
```

# ANSWERS - break & continue

- **break** What will this example print? **0,1,2,3,4**

```
count = 0
while True:
    print(count)
    count += 1
    if count >= 5:
        break
```

- **continue** What will this print? **only odd numbers 1,3,5,7,9**

```
for x in range(10):
    if x % 2 == 0:
        continue
    print(x)
```

# Week 4: Iteration –

- while loop vs for loop
  - While loop - **condition controlled**
  - For loop - **count controlled**
- Use of continue:
  - skips the current iteration and goes to the next iteration of the loop
- Use of break:
  - terminates the loop execution.