CSIT110

# Fundamental Programming with Python

Loop Statements (2)

Goh X. Y.

# In this lecture

- While loop
- Reading documentation

# While loop- what does it look like?

```
while (<condition that returns True | False>):
    # block statements when condition is True
```

# The first while-loop example

```
for i in range(0,10):
  print(i)
```

```
i = 0, print(i)
i = 1, print(i)
i = 2, print(i)
i = 3, print(i)
i = 4, print(i)
i = 5, print(i)
i = 6, print(i)
i = 7, print(i)
i = 8, print(i)
i = 9, print(i)
```

```
0
1
2
3
4
5
6
7
8
9
```

initialization statement

```
i = 0
while (i < 10):
  print(i)
  i = i + 1
```

conditional statement

prep statement

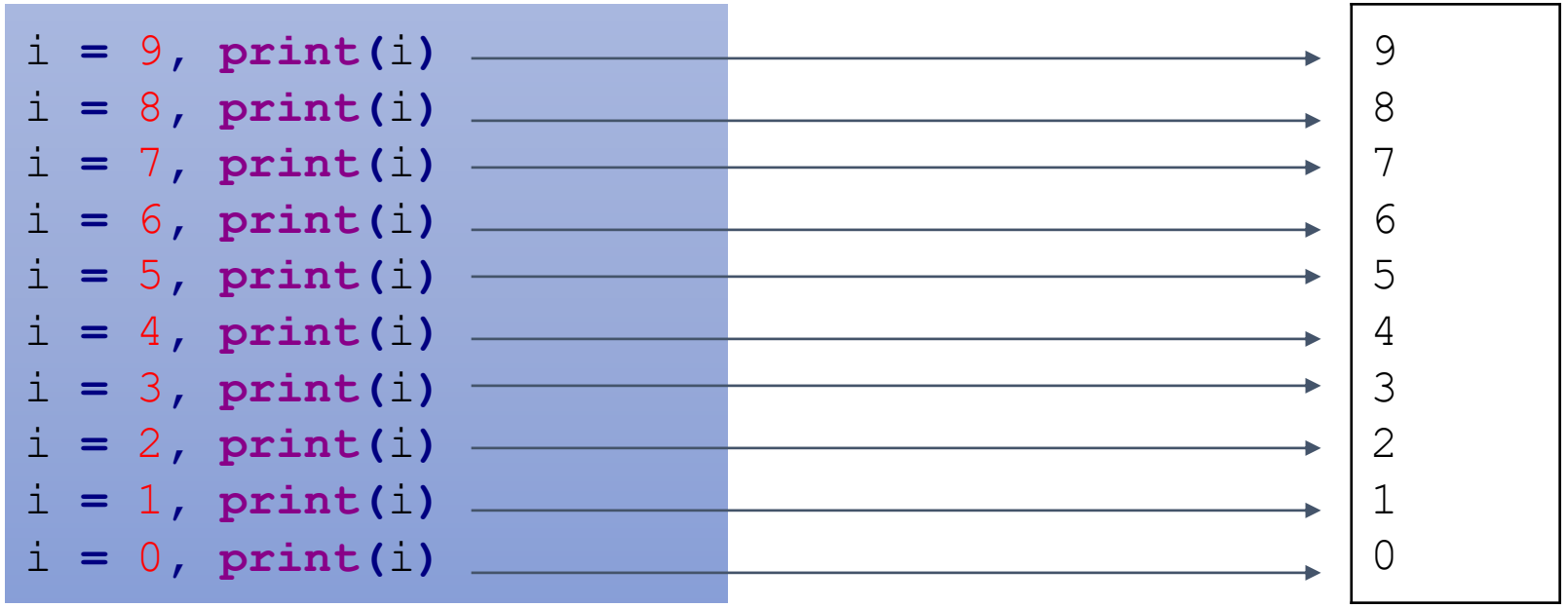
# Going backwards

initialization statement

prep statement

```
i = 9
while (i >= 0):
  print(i)
  i = i - 1
```

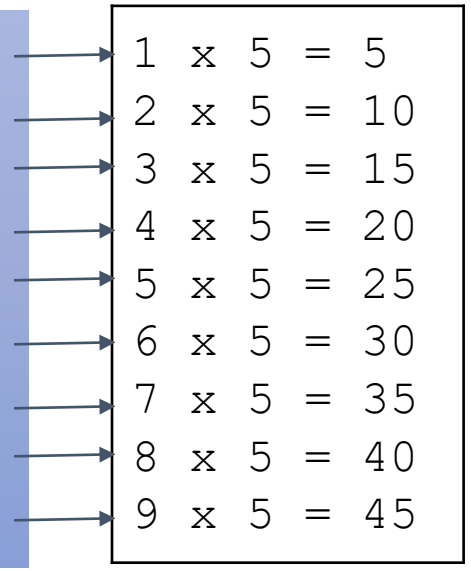
conditional statement



# Times table example

```
for i in range(1,10):
  print(f"{i} x {5} = {5*i}")
```

```
i = 1; print(f"{i} x {5} = {5*i}"))
i = 2; print(f"{i} x {5} = {5*i}"))
i = 3; print(f"{i} x {5} = {5*i}"))
i = 4; print(f"{i} x {5} = {5*i}"))
i = 5; print(f"{i} x {5} = {5*i}"))
i = 6; print(f"{i} x {5} = {5*i}"))
i = 7; print(f"{i} x {5} = {5*i}"))
i = 8; print(f"{i} x {5} = {5*i}"))
i = 9; print(f"{i} x {5} = {5*i}"))
```




```
i = 0
while (i < 10):
  print(f"{i} x {5} = {5*i}"))
  i += 1
```

# Friend of 10 table

```
for i in range(0,11):
  print("{i:>2} + {10-i:>2} = {10:>2}")
```


```
i = 0
while (i <= 10):
  print(f"{i:>2} + {10-i:>2} = {10:>2}")
  i += 1
```

# Questions

What is the output of the following codes?

A

```
i = 0
while (i < 10):
    print(i)
    i = i + 2
```

B

```
i = 0
while (i < 10):
    i = i + 2
    print(i)
```

# Questions

What is the output of the following codes?

C

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

D

```
i = 5
while (i < 10):
    print(i)
    i = i + 1
```

E

```
i = 5
while (i < 10):
    i = i + 1
    print(i)
```

# Questions

What is the output of the following codes?

F

```
i = 0
i = i + 1
while (i < 10):
    print(i)
    i = i + 1
```

G

```
i = 0
while (i < 10):
    print(i)
```

H

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

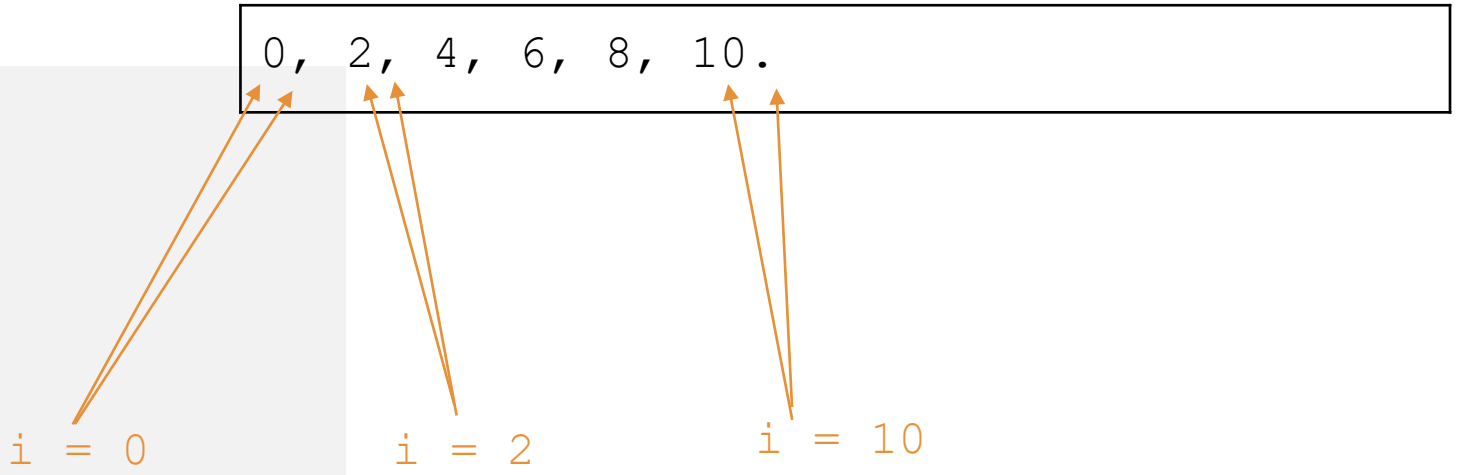
# Even numbers

```
i = 0
while (i <= 10):
  trailing = "cat"

  # display the number
  print(i, end="")

  # display the trailing
  print(trailing, end="")

  # update the even number
  i = i + 2
```



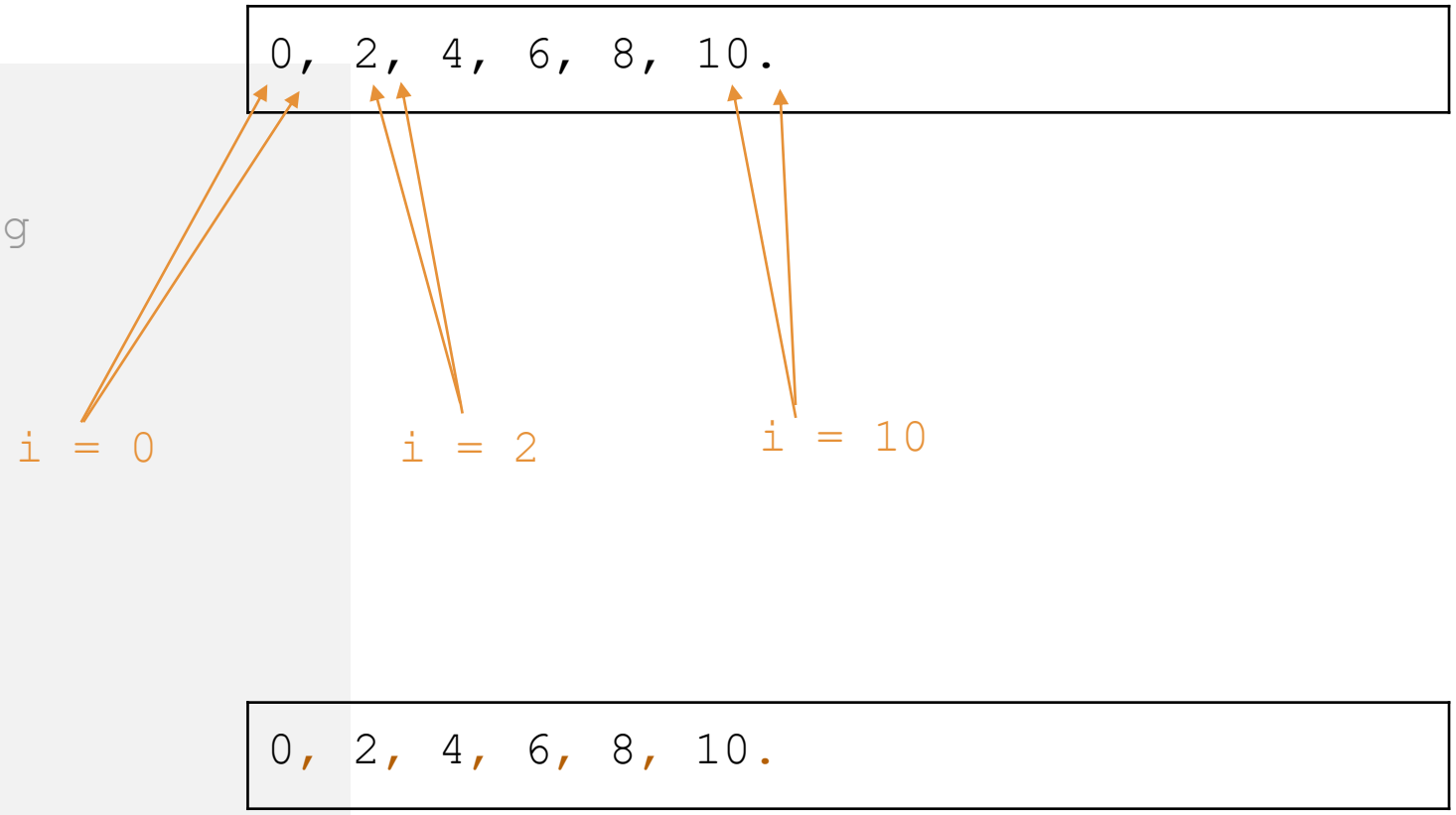

```
0cat2cat4cat6cat8cat10cat
```

# Even numbers

```
i = 0
while (i <= 10):
  # determine the trailing
  if (i < 10):
    trailing = ", "
  else:
    trailing = "."

  print(i, end="")
  print(trailing, end="")

  i = i + 2
```



# Display equations

```
Enter start number: 4
Enter end number: 7

Equations:
4 + 4 = 8
5 + 5 = 10
6 + 6 = 12
7 + 7 = 14
```

```
# ask user for start number

# ask user for end number

# display equations between the two input numbers
```

# Display equations

```
# ask user for start number and end number
user_input = input("Enter start number: ")
number_start = int(user_input)

user_input = input("Enter end number: ")
number_end = int(user_input)

# display equations between the two input numbers
print("Equations:")

# initialise number to the start number
number = number_start

# repeat as long as number is <= number_end
while(number <= number_end):
    print(f"{number} + {number} = {number*2}")
    # increase the number by 1
    number = number + 1
```

```
4 + 4 = 8
5 + 5 = 10
6 + 6 = 12
7 + 7 = 14
```

# Example 1: While loops that runs forever!

```
while True:
    user_input = input("Enter something: ")
    print("You have entered: " + user_input)
```

This program will run forever!

```
Enter something: Clocks on fox tick
You have entered: Clocks on fox tick
Enter something: Clocks on Knox tock
You have entered: Clocks on Knox tock
Enter something: Six sick bricks tick
You have entered: Six sick bricks tick
Enter something: Six sick chicks tock
You have entered: Six sick chicks tock
.....
```

# Example 2: This while loop will stop if user enters q

```
while True:
    user_input = input("Enter something (or q to quit): ")
    if (user_input == "q"):
        print("Goodbye!")
        break
print("You have entered: " + user_input) print()
```

*use* **break** *to stop the loop*

```
Enter something (or q to quit): Clocks on fox tick
You have entered: Clocks on fox tick

Enter something (or q to quit): Clocks on Knox tock
You have entered: Clocks on Knox tock

Enter something (or q to quit): Six sick bricks tick
You have entered: Six sick bricks tick

Enter something (or q to quit): q
Goodbye!
```

# Example 3: Keep asking until user enters a positive number

```
Enter a positive integer: -2

Enter a positive integer: 0

Enter a positive integer: -5

Enter a positive integer: 20
You have entered: 20
```

```
Enter a positive integer: 6
You have entered: 6
```

# Example 3: Keep asking until user enters a positive number

```
while True:
    user_input = input("Enter a positive integer: ")
    number = int(user_input)
    if (number > 0):
        break
print()
print(f"You have entered: {number}")
```

User has entered a positive number. Hurray!!!

```
Enter a positive integer: -2
Enter a positive integer: 0
Enter a positive integer: -5
Enter a positive integer: 20

You have entered: 20
```

# Example 4: Counting even and odd numbers

```
Enter an integer (or q to quit): 5
Enter an integer (or q to quit): 7
Enter an integer (or q to quit): 0
Enter an integer (or q to quit): 13
Enter an integer (or q to quit): 8
Enter an integer (or q to quit): 15
Enter an integer (or q to quit): q
You have entered 2 even numbers
You have entered 4 odd numbers
```

# Example 4: Counting even and odd numbers

```
even_count = 0
odd_count = 0
while True:
    user_input = input("Enter an integer (or q to quit): ")

    if (user_input == "q"):
        break

    number = int(user_input)

    if (number%2 == 0):
        even_count += 1
    else:
        odd_count += 1

print(f"You have entered {even_count} even numbers")
print(f"You have entered {odd_count} odd numbers")
```

# The `continue` keyword

# The `continue` keyword

The **`continue`** statement terminates the closest enclosing loop.

```
for num in range(1, 10):

    if num % 2 == 0:
        print("Found an even number", num)
        continue

    print("Found an odd number", num)
```

*use **`continue`** to go to the next loop*

# The `continue` keyword

```
1 is an odd number
2 is an even number
3 is an odd number
4 is an even number
5 is an odd number
6 is an even number
7 is an odd number
8 is an even number
9 is an odd number
```

# Extra:

Learning to read Python docs

Any questions?