Ages 15+
& Adults

CODEXPLORERS

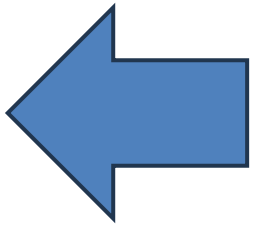
Intro to Python Programming

Week 3: Loops & Functions

Class Schedule

Class **1** : Introduction to basic Python concepts

Class **2** : Inputs, Conditionals & Decisions

Class **3** : Loops, functions & repetition 

Class **4** : Project showcase

Recap : If / Elif / Else Explained

Taking it a step further…

if checks the first condition
elif (short for "else if") checks another condition if the first one was false
else runs if none of the conditions were true.

Example :
🚦 Think of it like a traffic light
- If it's red, stop.
- Else if it's yellow, slow down.
- Else (must be green), go!

Loops

- Loops let us repeat actions automatically.
- No point copy-pasting code many times

Example : If you want to walk 4 steps, you do not have to instruct to step once, then again, then again, and then again.

Why Use Loops?

Time Savings

Makes programs neat and short

Handle big tasks easily

2 looping techniques we will learn about:

- The **for** loop
- The **while** loop

The for Loop

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

Step	What Happens
range(5)	Creates the numbers 0, 1, 2, 3, 4 (NOT 5)
for i in range(5)	For each number in that list, do something
print(i)	Print the current number

What Python actually does:

1. Set i = 0, print 0
2. Set i = 1, print 1
3. Set i = 2, print 2
4. Set i = 3, print 3
5. Set i = 4, print 4
6. Then stop (because 5 is NOT included)

Another way for a for Loop

for i in range(start,end):

    print(i) #REMEMBER: the end number is NOT printed.

Example:

One last way!!!

hint this will be useful for our coding exercise on the Countdown Timer

for variable name in range(start,end,increment/decrement):

perfect for needing numbers that are decreasing or increasing by a certain amount rather than just by 1

Important Example

```
1   for y in range(2,5,3):
2       print(y)
```

What will this print?

If you said it will print 2,5… REMEMBER:

for loops will NOT print the number that is at the end range

It will therefore print 2 only.

Quick quiz

Quiz

Answers

The `while` Loop

- A while loop is used when you don't know exactly how many times you need to repeat something.
- It keeps going as long as the condition is True

```
count = 0

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

The `while` Loop

Step	What Happens
1	Start: count = 0
2	Check: Is count < 5? (Yes)
3	Print count (which is 0)
4	Increase count by 1 (count = 1)
5	Check: Is count < 5? (Yes)
6	Print count (which is 1)
7	Increase count by 1 (count = 2)
8	Check: Is count < 5? (Yes)
9	Print count (which is 2)
10	Increase count by 1 (count = 3)
11	Check: Is count < 5? (Yes)
12	Print count (which is 3)
13	Increase count by 1 (count = 4)
14	Check: Is count < 5? (Yes)
15	Print count (which is 4)
16	Increase count by 1 (count = 5)
17	Check: Is count < 5? (No)
18	**Stop the loop**

```
count = 0

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

The `while` Loop

❗ **Important:**

- **If you forget to change the variable inside the loop**, it will run **forever**!
  ➔ (This is called an **infinite loop**.)
- Always make sure something inside the loop **changes the condition**.

💡 **Why while loops are useful:**

Perfect when you **don't know how many times** you need to repeat.

Example: Keep asking until someone types the right answer!

Coding Activity : Countdown Timer

Activity 1

- Use a for loop to count down from 10.
  - Print each number with a pause if you like!

Two ways to do:

```
1   for i in range(10,0,-1):
2       print(i)
```

```
1   num = 10
2   while(num >=1):
3       print(num)
4       num-=1
```

You code : Keep Guessing the secret word

Write a program that:

- Tries and guesses a secret word from the input

Sample Code

```
# Secret word
secret_word = "python"

# Ask the user to guess
guess = ""

while guess != secret_word:
    guess = input("Guess the secret word: ")

print("You guessed it! The secret word was 'python'.")
```

Intro to Functions

Functions : Reusable blocks of code

- These help simplify programs.
- Concise
- Avoids repeating code
- Break big problems into manageable chunks

Intro to Functions

Day-to-Day Analogy

A kitchen blender could be a function.

- You **put ingredients** (fruit, milk, ice) **into** the blender.
- You **press a button** (run the blender).
- The blender **does its job** and gives you **a smoothie**!
- ✅ You don't need to know **how** the blender works inside —
  ✅ You just **use it** whenever you need a smoothie.

Coding Activity – Creating a simple function

```
# Define the function
def say_hello():
    print("Hello, Python Explorer!")

# Call (use) the function
say_hello()
```

More about Functions

Functions can also take in parameters and return values.

```
# Define the function
def double_number(x):
    result = x * 2
    return result

# Call (use) the function
answer = double_number(5)
print(answer)
```

Coding Activity – For loop + Function

- Write a program that prints a greeting 5 times. Keep the greeting into a function

Sample code

```
# Step 1: Define the function
def say_hello():
    print("Hello, Python Explorer!")

# Step 2: Use a for loop to call it 5 times
for i in range(5):
    say_hello()
```

Key takeaways & Upcoming topics

- Loops (For loops, while loops) help repeat actions
- Functions help organize code

Next Week

- Mini-project showcase!
  - Combine conditionals, loops and functions to create something cool!

THANK YOU