Day 2: Control Structures

What Are Conditionals? (If, Elif, Else)

- **Imagine you're deciding what to wear based on the weather.** If it's sunny, you wear shorts. If it's rainy, you wear a raincoat. If it's cold, you wear a sweater. This is what we call a **conditional** in programming.
- **In Python, we use "if", "elif", and "else" to make decisions**:
  - **If** it's sunny, do this.
  - **Elif** (which means "else if") it's rainy, do that.
  - **Else** (if nothing else is true), do something else.

Example:

```
python

weather = "sunny"
Ok
if weather == "sunny":
    print("Wear shorts!")
elif weather == "rainy":
    print("Wear a raincoat!")
else:
    print("Wear a sweater!")
```

What Are Nested Conditionals?

- **Imagine you have two decisions to make, one inside the other.** Let's say you want to decide what to wear, but you also need to decide whether to take an umbrella based on the weather.
- **First decision**: Is it sunny or not?
- **Second decision (inside the first one)**: If it's sunny, is it hot or cool?
- This is called a **nested conditional** because you have one "if" statement inside another.

Example in Python:

```
python

weather = "sunny"
temperature = "hot"
```

```
if weather == "sunny":
    if temperature == "hot":
        print("Wear shorts!")
    else:
        print("Wear a light jacket!")
else:
    if weather == "rainy":
        print("Take an umbrella!")
    else:
        print("Wear something warm!")
```

What Are Conditional Operators?

- **Conditional operators** are like the tools we use to compare things. Imagine you're comparing two toys, two numbers, or even two ideas, and you want to know how they relate to each other.

- These operators help us answer questions like:

  - Are two things equal?
  - Is one thing bigger or smaller than the other?
  - Is something true or false?

Common Conditional Operators:

1. **Equal to (==)**: Checks if two values are the same.
2. **Not equal to (!=)**: Checks if two values are different.
3. **Greater than (>)**: Checks if one value is bigger than another.
4. **Less than (<)**: Checks if one value is smaller than another.
5. **Greater than or equal to (>=)**: Checks if one value is bigger than or the same as another.
6. **Less than or equal to (<=)**: Checks if one value is smaller than or the same as another.

Example:

```
python

age = 12

if age >= 13:
    print("You can watch a PG-13 movie!")
else:
    print("You are too young to watch a PG-13 movie.")
```

What is a `for` Loop?

- **Imagine you have a list of chores to do, like cleaning your room, doing your homework, and feeding your pet.** You want to go through each chore, one by one, and check it off your list.
- A **`for` loop** in Python is like a helper that goes through each item on your list and does something with it.

How the `for` Loop Works:

- **The `for` loop syntax**:python

```
for item in list_of_items:
    # Do something with the item
```

-
-

Example:

```
python

# List of fruits
fruits = ["apple", "banana", "cherry"]

# Loop through each fruit and print it
for fruit in fruits:
    print(fruit)
```

What is a `while` Loop?

- **Imagine you're playing a game where you need to keep jumping until you reach a certain height.** You don't know how many jumps it will take, but you know you need to keep jumping until you get there.
- A **`while` loop** in Python is like telling the computer, "Keep doing something **while** this condition is true."

How the `while` Loop Works:

- **The `while` loop syntax**:python

```
while condition_is_true:
    # Do something
```

-

-

Example:

```
python

# Counting from 1 to 5
number = 1

while number <= 5:
    print(number)
    number += 1  # This adds 1 to the number each time the
loop runs
```

What Are Loop Control Statements? (Break, Continue, Pass)

- **Sometimes, you need to stop or skip an action in a loop.**
  - **`break`**: Stops the loop completely.
  - **`continue`**: Skips to the next item in the loop and continues.
  - **`pass`**: Does nothing; it's just a placeholder.

Examples:

```
python

# break example
for i in range(1, 6):
    if i == 3:
        break  # Stops the loop at 3
    print(i)

# continue example
for i in range(1, 6):
    if i == 3:
        continue  # Skips printing 3 and goes to the next
number
    print(i)

# pass example
for i in range(1, 6):
    if i == 3:
        pass  # Does nothing special when i is 3
    print(i)
```

Let's Practice with Simple Problems!

1. **Is the Number Positive or Negative?**
    - **Problem**: Let's ask Python if a number is positive (like +5), negative (like -3), or zero.
    - **How it works**: Python will check and tell us what type of number we have.

Solution:

```
python

num = int(input("Enter a number: "))
if num > 0:
    print("The number is positive.")
elif num == 0:
    print("The number is zero.")
else:
    print("The number is negative.")
```

2. **Is the Number Even or Odd?**
    - **Problem**: Let's ask Python if a number is even (like 2, 4, 6) or odd (like 1, 3, 5).
    - **How it works**: Python will do the math for us and tell us if the number is even or odd.

Solution:

```
python

num = int(input("Enter a number: "))
if num % 2 == 0:
    print("The number is even.")
else:
    print("The number is odd.")
```

3. **Counting to 10**:
    - **Problem**: Let's ask Python to count from 1 to 10.
    - **How it works**: Python will use a loop to count up, just like you would with your fingers.

Solution:

```
python
```

```
for i in range(1, 11):
    print(i)
```

4. **Summing Numbers**:
   - **Problem**: Let's ask Python to add up all the numbers from 1 to 100.
   - **How it works**: Python will keep adding the numbers one by one until it reaches 100.

Solution:

```
python

total = 0
for i in range(1, 101):
    total += i
print("The sum is:", total)
```

5. **Factorial Fun**:
   - **Problem**: Let's ask Python to calculate the factorial of a number. (A factorial is when you multiply a number by every number below it, like 5! = 5 × 4 × 3 × 2 × 1).
   - **How it works**: Python will multiply the numbers together until it reaches the number you asked for.

Solution:

```
python

num = int(input("Enter a number: "))
factorial = 1
for i in range(1, num + 1):
    factorial *= i
print("The factorial is:", factorial)
```

In Summary:

- **Conditionals** help your program make decisions like "if it's sunny, wear shorts."
- **Loops** help your program repeat actions, like saying "Good Morning!" to each friend.
- **Nested Conditionals** are like making a decision inside another decision.
- **Conditional Operators** compare things, like checking if two numbers are the same.
- **For Loops** repeat actions for each item in a list, and **While Loops** keep going as long as something is true.
- **Loop Control Statements** like `break`, `continue`, and `pass` help you control what happens in your loops.