جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم

قسم الانظمة الطبية الذكية

Intelligent Medical Systems Department

Lecture: (1)

Python Basics

Subject: Data structure

Class: second

Lecturer: Prof. Mehdi Ebady Manaa

Programmer:- Fatima Hussein Jawad

Study Year: 2024-2025

1. Introduction to Python :

Python is an easy and fun programming language, widely used in web development, data analysis, artificial intelligence, and others. Python is characterized by being a high-level, open source, and multi-purpose language.

Python features:

- ✓ Easy to read and write.
- ✓ Cross-platform (can be run on Windows, Linux, MacOS).
- ✓ Huge libraries that facilitate various tasks (such as the NumPy library, Pandas, Matplotlib).

2. Installing Python :

Before starting, make sure to download Python from the official website. You can also use integrated development environments (IDEs) such as PyCharm or VSCode to facilitate programming.

Printing simple statements using print()

The first thing we usually start with is printing the sentence "Hello, World!".

```
print("Hello, World!")
```

Explanation: print() is a function in Python that is used to print strings or values.

Variables :

Variables are containers for storing data. In Python, you can create variables without specifying a data type.

```
x = 10          # x holds an integer.
```

```
y = "Hello"              # y holds a string.
z = 3.5                  # z holds a float.
```

Arithmetic Operations :

Python can easily perform basic arithmetic operations.

```
a = 10
b = 3
print(a + b)      # Addition
print(a - b)      # Subtraction
print(a * b)      # Multiplication
print(a / b)              # Division
print(a % b)      # Remainder
print(a ** b)     # Power (a raised to the power of b)
```

Conditional Statements:

1. Conditional statements allow us to control the flow of the program based on conditions.

```
age = 18
if age >= 18:
        print("You are an adult.")
else:
        print("You are a minor.")
```

Explanation: if checks the condition, if it is true the code is executed, otherwise the else part is executed.

Example: Checking a number if it is positive

```
num = 10

if num > 0:

        print("The number is positive.")
```

Explanation: The message is printed only if the number is greater than 0. If the condition is not met, nothing is executed.

2. Using if-else :

Example: Checking a number if it is positive or negative

```
num = -5

if num > 0:

        print("The number is positive.")

else:

        print("The number is negative.")
```

Explanation: If the number is positive, the first message is printed, and if it is not positive (i.e. negative), the second message is printed.

3. Using if-elif-else :

Example: Classifying a number as positive, negative, or zero

```
num = 0

if num > 0:

        print("The number is positive.")

elif  num == 0:

        print("The number is zero.")
```

```
else:
        print("The number is negative.")
```

Explanation: If the number is greater than 0, the first message is printed. If it is equal to 0, the second message is printed, and if it is less than 0, the third message is printed.

Loops:

1. for loops

- **Used to repeat a block of code**:

```
for i in range(5):
        print(i)          # will print the numbers from 0 to 4
```

- **Using for to iterate over a list:**

Example: Print each element in a list

```
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
        print(fruit)
```

Explanation: On each iteration, a new value from the list fruits is assigned to the variable fruit and the value is printed.

- **Using for with an inner conditional statement :**

Example: Print only even numbers

```
for i in range(10):
if i % 2 == 0:
print(f"{i} is even.")
```

Explanation: The loop prints only even numbers from 0 to 9 using a condition inside the loop that checks if the value of i is divisible by 2 without a remainder.

2. while loops

Continues to repeat as long as the condition is true.

```
count = 0
while count < 5:
print(count)
count += 1      # The value of the variable is incremented by 1
```

Example: While loop to print numbers from 1 to 5

```
i = 1
while i <= 5:
print(i)
        i += 1          # Incrementing the variable i by 1 in each iteration
```

Explanation: The loop continues to iterate as long as the value of the variable i is less than or equal to 5. After each iteration, the value of i is incremented by 1.

Using while with a compound conditional statement:

Example: Checking the user's input until the correct number is entered

```
number = 0
while number < 1 or number > 10:
number = int(input("Enter a number between 1 and 10: "))
```

```
print(f"You entered {number}.")
```

Explanation: This loop continues to iterate over the user's input request until he enters a number between 1 and 10.

Using break and continue :

Example: Using break to exit a loop

```
for i in range(10):
    if i == 5:
    break          # Stops the loop when it reaches the number 5
    print(i)
```

Explanation: The loop stops as soon as the value of i reaches 5 because of using break.

Example: Using continue to skip the iteration :

```
for i in range(10):
 if i % 2 != 0:
    continue          # Skips odd numbers
    print(i)
```

Explanation: The loop skips printing odd numbers using continue, and prints only even numbers.

Basic Data Structures :

1. Lists :

A list is a type of variable that can contain multiple elements.

```
my_list = [1, 2, 3, 4, 5]
print(my_list)
print(my_list[0])        # Accessing the first element
```

2. Dictionaries :

Dictionaries contain key/value pairs.

```
my_dict = {"name": "Fatima", "age": 25, "job": "Teacher"}
```

code:

```
my_dict = {"name": "Fatima", "age": 25, "job": "Teacher"}

print(my_dict["name"])          # Access the value using the key
```