Introduction to Python

DR. UGUR GUVEN

AEROSPACE ENGINEER & PROFESSOR

What is Python?

- Phyton is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.
- Python is a programming language that lets you work quickly and integrate systems more efficiently.

Phyton Fundamentals

1. Variables: In Python 3, variables are created by assigning a value to a name. For example, x = 5 creates a variable called x and assigns the value 5 to it.
2. Data types: Python 3 supports several built-in data types, including integers, floats, strings, booleans, lists, tuples, and dictionaries.
3. Operators: Python 3 supports a variety of operators, including arithmetic operators (+, -, *, /), comparison operators (>, <, ==, !=), and logical operators (and, or, not).
4. Control flow statements: Python 3 supports several control flow statements, including if-else statements, for loops, and while loops. These statements allow you to control the flow of execution in your code..

5. Input and output: In Python 3, you can use the input() function to get user input, and the print() function to output text to the console.

Python Basic Program Examples

- Hello World

```
# This program prints Hello, world!
print('Hello, world!’)
```

- Add Two Numbers

```
# This program adds two numbers
num1 = 1.5
num2 = 6.3
# Add two numbers
sum = num1 + num2
# Display the sum
print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
```

Python Output

- Print command is used for basic output

```
print("Hello", "how are you?")
Hello how are you?
```

```
x = ("apple", "banana", "cherry")
print(x)
'apple', 'banana', 'cherry'
```

Python Input

- Input command is used for inputting data (similar to read command in FORTRA

```
# Python program showing input
val = input("Enter your value: ")
print(val)

name = input('What is your name?\n')
# \n ---> newline  ---> It causes a line break
print(name)
```

Input and Output Together

```
print("Enter First Number: ")
numOne = int(input())
print("Enter Second Number: ")
numTwo = int(input())
res = numOne+numTwo
print("\nAddition Result = ", res)
res = numOne-numTwo
print("Subtraction Result = ", res)
res = numOne*numTwo
print("Multiplication Result = ", res)
res = numOne/numTwo
print("Division Result = ", res
```

```
Enter First Number:

4

Enter Second Number:

5

Addition Result =  9

Subtraction Result =  -1

Multiplication Result =  20

Division Result =  0.8
```

Loops

```
for i in range(5):
print("Hello World!")
```

```
Hello World!

Hello World!

Hello World!

Hello World!

Hello World!
```

If Statement

- Python supports the usual logical conditions from mathematics:
- Equals: `a == b`
- Not Equals: `a != b`
- Less than: `a < b`
- Less than or equal to: `a <= b`
- Greater than: `a > b`
- Greater than or equal to: `a >= b`
- These conditions can be used in several ways, most commonly in "if statements" and loops.
- An "if statement" is written by using the `if` keyword.

If Statement Example

```python
a = 33
b = 200
if b > a:
  print("b is greater than a")
```

Elif and Else in If Statement

- The elif keyword is Python's way of saying "if the previous conditions were not true, then try this condition”
-
```
a = 33
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
```

- The else keyword catches anything which isn't caught by the preceding conditions.
-
```
a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")
```