



DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING

Title: Python Basics

DATA MINING LAB
CSE 424



GREEN UNIVERSITY OF BANGLADESH

1 Objective(s)

- To gather knowledge of Python program.

2 Python Programming

Python is an interpreted high-level general-purpose programming language. Python's design philosophy prioritizes code readability, as evidenced by its extensive use of indentation. Its language elements and object-oriented approach are aimed at assisting programmers in writing clear, logical code for both small and large-scale projects.

3 Basics of Python

Python is a high-level, dynamically typed multi-paradigm programming language. Python code is often said to be almost like pseudo-code, since it allows you to express very powerful ideas in very few lines of code while being very readable.

3.1 Variables and Strings

Variables are used to store values. A string is a series of characters, surrounded by single or double quotes.

3.1.1 Implementation in Python

```
1 print("Hello world!")
2
3 # Hello world with a variable
4 msg = "Hello world!"
5 print(msg)
6
7 # Concatenation (combining strings)
8 first_name = 'albert'
9 last_name = 'einstein'
10 full_name = first_name + ' ' + last_name
11 print(full_name)
```

3.1.2 Input/Output

Output of the program is given below.

```
Hello world!
Hello world!
albert einstein
```

3.1.3 Casting

Casting is a method for specifying the data type of a variable. The **type()** method is used to find the data type of a variable.

3.1.4 Implementation in Python

```
1 x = str(5)      # x will be '5'
2 y = int(5)      # y will be 5
3 z = float(5)    # z will be 5.0
4
5 print(type(x))
6 print(type(y))
7 print(type(z))
```

3.1.5 Input/Output

Output of the program is given below.

```
<class 'str'>
<class 'int'>
<class 'float'>
```

3.2 User Input

Python accepts user input. That means we can request input from the user. In Python 3.6, the procedure is slightly different than in Python 2.7. The **input()** method is used in Python 3.6. The **raw_input()** method is used in Python 2.7. Python programs can prompt the user for input.

All input is stored as a string. Then it has to type cast to the required data type

3.2.1 Implementation in Python

```
1 name = input("Enter name: ")
2 print("name is: " + name)

1 age = input("Enter your age: ")
2 age = int(age)
3
4 print("Age is: ", age)
```

3.2.2 Input/Output

Output of the programs is given below.

```
Enter name: Mike
name is: Mike
```

```
Enter your age: 25
Age is: 25
```

3.3 Operators

Variables are used to store values. A string is a series of characters, surrounded by single or double quotes.

3.3.1 Arithmetic Operators

3.3.2 Implementation in Python

```
1 x = 10
2 y = 5
3
4 print(x + y)      # Addition
5
6 print(x - y)      # Subtraction
7
8 print(x * y)      # Multiplication
9
10 print(x / y)     # Division
11
12 print(x % y)     # Modulus
13
14 print(x ** y)    # Exponentiation
```

```
15  
16 print(x // y)      # Floor division
```

3.3.3 Comparison Operators

```
1 x = 10  
2 y = 5  
3  
4 print(x == y)      # Equal  
5  
6 print(x != y)      # Not equal  
7  
8 print(x > y)       # Greater than  
9  
10 print(x < y)       # Less than  
11  
12 print(x >= y)      # Greater than or equal to  
13  
14 print(x <= y)      # Less than or equal to
```

3.3.4 Logical Operators

```
1 x = 10  
2  
3 print(x < 5 and x < 10)      # Returns True if both statements are true  
4  
5 print(x < 5 or x < 4)       # Returns True if one of the statements is true  
6  
7 print(x < 5 and x < 10)      # Reverse the result, returns False if the result is  
    true
```

3.3.5 Lab Task (Please implement yourself and show the output to the instructor)

- Write a Python program which accepts the user's first and last name and print them in a single line as full name.
- Write a Python program that accepts an integer (n) and computes the value of $n + nn + nnn$.
- Write a Python program which accepts the radius of a circle from the user and compute the area.
- Write a Python program to find whether a given number (accept from the user)

3.4 List

A list stores a series of items in a particular order. You access items using an index, or within a loop.

3.4.1 Implementation in Python

```
1 # Create a List  
2 thislist = ["apple", "banana", "cherry"]  
3 print(thislist)  
4  
5 # Lists allow duplicate values:  
6 thislist = ["apple", "banana", "cherry", "apple", "cherry"]  
7 print(thislist)
```

3.4.2 Input/Output

Output of the program is given below.

```
['apple', 'banana', 'cherry']  
['apple', 'banana', 'cherry', 'apple', 'cherry']
```

3.4.3 List Length

Use the `len()` method to find out how many items are in a list:

3.4.4 Implementation in Python

```
1 thislist = ["apple", "banana", "cherry"]  
2 print(len(thislist))  
3  
4 thislist = [1, 5, 7, 9, 3]  
5 print(len(thislist))
```

3.4.5 Input/Output

Output of the program is given below.

```
3  
5
```

3.4.6 Access List Items

List items are indexed and can be accessed by referring to the index number:

3.4.7 Implementation in Python

```
1 list = ["apple", "banana", "cherry"]  
2  
3 # Get the first item in a list  
4 print(list[0])  
5  
6 # Get the last item in a list  
7 print(list[-1])
```

3.4.8 Input/Output

Output of the program is given below.

```
apple  
cherry
```

3.4.9 Range of Indexes

A range of indexes can be specified by indicating where the range begins and ends. When specifying a range, the return value will be a new list with the specified items.

3.4.10 Implementation in Python

```
1 thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
2 print(thislist[2:5])
3
4 # This will return the items from index 0 to index 5.
5 print(thislist[:5])
6
7 # This will return the items from index 2 to the end.
8 print(thislist[2:])
```

3.4.11 Input/Output

Output of the program is given below.

```
['cherry', 'orange', 'kiwi']

['apple', 'banana', 'cherry', 'orange', 'kiwi']

['cherry', 'orange', 'kiwi', 'melon', 'mango']
```

3.4.12 Change List Items

3.4.13 Implementation in Python

```
1 thislist = ["apple", "banana", "cherry"]
2
3 # To change the value of a specific item, refer to the index number
4 thislist[1] = "blackcurrant"
5 print(thislist)
6
7 # Change the second value by replacing it with two new values
8 thislist[1:2] = ["blackcurrant", "watermelon"]
9 print(thislist)
```

3.4.14 Input/Output

Output of the program is given below.

```
['apple', 'blackcurrant', 'cherry']

['apple', 'blackcurrant', 'watermelon', 'cherry']
```

3.4.15 Add List Items

Use the **append()** method to add a new item to the end of the list.

3.4.16 Implementation in Python

```
1 thislist = ["apple", "banana", "cherry"]
2 thislist.append("mango")
3 thislist.append("orange")
4 thislist.append("gauva")
5 print(thislist)
```

3.4.17 Input/Output

Output of the program is given below.

```
['apple', 'banana', 'cherry', 'mango', 'orange', 'gauva']
```

3.4.18 Insert Items

Use the **insert()** method to insert a list item at a specific index.

3.4.19 Implementation in Python

```
1 thislist = ["apple", "banana", "cherry"]
2 thislist.insert(1, "orange")
3 print(thislist)
```

3.4.20 Input/Output

Output of the program is given below.

```
['apple', 'orange', 'banana', 'cherry']
```

3.4.21 Extend List

Use the **extend()** method to append elements from another list to the current one.

3.4.22 Implementation in Python

```
1 thislist = ["apple", "banana", "cherry"]
2 list2 = ["mango", "pineapple", "papaya"]
3 thislist.extend(list2)
4 print(thislist)
```

3.4.23 Input/Output

Output of the program is given below.

```
['apple', 'banana', 'cherry', 'mango', 'pineapple', 'papaya']
```

3.4.24 Remove List Items

```
1 thislist = ["apple", "banana", "cherry"]
2 thislist.remove("banana") # Remove "banana"
3 print(thislist)
```

```
1 thislist = ["apple", "banana", "cherry"]
2 thislist.pop(1) # Remove the second item
3 print(thislist)
```

```
1 thislist = ["apple", "banana", "cherry"]
2 thislist.pop() # Remove the last item
3 print(thislist)
```

```
1 thislist = ["apple", "banana", "cherry"]
2 del thislist[0] # Remove the first item
3 print(thislist)
```

```
1 thislist = ["apple", "banana", "cherry"]
2 del thislist # Delete the entire list
```

```
1 thislist = ["apple", "banana", "cherry"]
2 thislist.clear() # Clear the list content
3 print(thislist)
```

3.4.25 Lab Task (Please implement yourself and show the output to the instructor)

- Write a Python program which accepts a sequence of comma-separated numbers from user and generate a list with those numbers.
- Write a Python program to display the first and last colors from the following list. Go to the editor
colorList == ["Red","Green","White" ,"Black"]

3.5 Conditional Statements

There are some conditional statements which can be used in python programming.

3.5.1 If statement

```
1 a = 10
2 b = 20
3 if b > a:
4     print("b is greater than a")
```

Python uses indentation (the white space at the starting of a line) to define scope in its code. Curly brackets are commonly used in other computer languages for this reason.

3.5.2 If..Else If..Else Statement

```
1 a = 20
2 b = 10
3 if b > a:
4     print("b is greater than a")
5 elif a == b:
6     print("a and b are equal")
7 else:
8     print("a is greater than b")
```

3.5.3 Input/Output

Output of the program is given below.

```
a is greater than b
```


3.5.4 Nested If Statement

```
1 x = 15
2
3 if x > 5:
4     print("Above ten, ")
5     if x > 10:
6         print("and also above 20!")
7     else:
8         print("but not above 20.")
```

3.5.5 Short Hand If statement

```
1 a = 20
2 b = 5
3
4 if a > b: print("a is greater than b")
```

3.5.6 Short Hand If statement

```
1 a = 2
2 b = 3
3
4 print("A") if a > b else print("B")
```

3.5.7 Lab Task (Please implement yourself and show the output to the instructor)

- Write a Python program to find the largest among three numbers.
- Write a Python program to display student grades.
- Write a Python program to calculate the sum of three given numbers, if the values are equal then return three times of their sum. is even or odd, print out an appropriate message to the user.

4 Discussion Conclusion

Based on the focused objective(s) to understand about the python program, the additional lab exercise made me more confident towards the fulfilment of the objectives(s).

5 Lab Exercise (Submit as a report)

- Write a Python program to find the sum of odd and even numbers from a set of numbers.
- Write a Python program to Check Triangle is Valid or Not

6 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.