By: Saad B. Younis

Experiment Number (1)

saad@coie-nahrain.edu.iq

Built-in Function

Built-in functions are pre-defined in the programming language's library, for the programming to directly call the functions wherever required in the program for achieving certain functional operations. A few of the frequently used built-in function in the Python programs are :

input()	Allowing user input
Print()	Prints to the standard output device

Variables: Variables are containers for storing data values.

Example (1)

```
name = input('Enter your name:')
print('Hello, ' + name)
 }
```

Python Data Types

In programming, data type is an important concept.

Variables can store data of different types, and different types can do different things. Python has the following data types built-in by default, in these categories:

Python Data Types

Text Type:	**str**
Numeric Types:	**int, float, complex**
Sequence Types:	**list, tuple, range**
Mapping Type:	**dict**
Set Types:	**set, frozenset**
Boolean Type:	**bool**

Example (2)-int

```python
# addition of two numbers

x = int(input("Enter First Number: ") )
y = int(input("Enter Second Number: ") )

z = x + y

print(z)
```

Example (3)-str

```python
# Ask the user for their name
name = input("What's your name? ")

# Remove whitespace from the str and capitalize the first letter of each word
name = name.strip().title()

# Print the output
print("hello,", {name})
```

Example (4)-float

```
# Get the user's input
x = float(input("What's x? "))
y = float(input("What's y? "))

# Create a rounded result
z = round(x + y)

# Print the formatted result
print(z)
```

Python Function

A function is a block of code which only runs when it is called.

You can pass data, known as parameters, into a function.

A function can return data as a result.

Example (5)-function

```
#Create Function
def my_function():
  print("Hello from a function")

#Call Function
my_function()
```

Arguments

Information can be passed into functions as arguments.

Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

Example (6)-function + Arguments

```
def my_function(fname):
  print("Hello,",fname)


my_function("Ahmad")
my_function("Rania")
```

Example (7)-function + Return Values

```
def my_function(x):
  return 5 * x

print(my_function(3))
print(my_function(5))
print(my_function(9))
```