Hacettepe University

Computer Engineering Department

Programming in  python

BBM103 Introduction to Programming Lab 1

Week 4

Fall 2017

Install PyCharm

Download Link : https://www.jetbrains.com/pycharm-edu/download/#section=windows

Guide : https://www.jetbrains.com/help/pycharm/quick-start-guide.html

Writing Your First Program

- Example 1: **printing output**

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

New function:
`print()`

- Example 2:

```
language = "Python programming language"
print(language)
```

- Example 3: **printing multiple lines**

```
print("""
|===========BBM103===========|
|                            |
|  Welcome to Programming!   |
|  Python is easy and fun!   |
|                            |
|============================|
""")
```

- Example 4: **customizing the separator between printed items**

```
print("L", "i", "n", "u", "x", sep=".")

print(*"Linux", sep=".")
```

  - **Output of both lines:** `L.i.n.u.x`

Taking Input

- Example 5: **taking the input as a string**

```
name = input("What is your name? ")
print("Hello", name, end="!\n")
```

New function: `input()`

- Example 6: **converting the input to integer**

```
number = int(input("Please enter a number: "))
print("The square of the number: ", number ** 2)
```

New function: `int()`

```
# We can do the same operation with pow() function: pow(number,2)
```

- Example 7:

```
number1 = int(input("Enter the first number: "))
number2 = int(input("Enter the second number: "))
print(number1, "+", number2, "=", number1 + number2)
```

New function:
`str.format()`

- Example 8: **formatting output**

```
url = input("Please enter the url")
print("Error! Google Chrome couldn't find {} ".format(url))
```

- Example 9: **formatting output**

```
print('{0} and {1}'.format('Tom', 'Jerry'))
```

  - **Output:** `Tom and Jerry`

Control Flow - Branching

The simplest branching statement is a **conditional**. `<condition>` has a value `True` or `False`. `<expression>` is evaluated if `<condition>` is `True`.

1

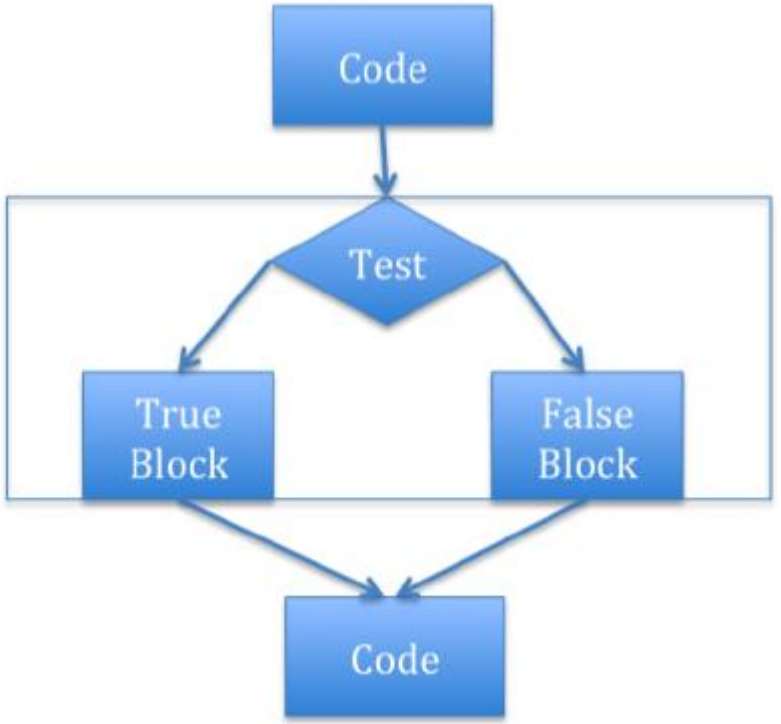
```
if <condition>:
    <expression>
    <expression>
    ...
```

2

```
if <condition>:
    <expression>
    <expression>
    ...
else:
    <expression>
    <expression>
    ...
```

3

```
if <condition>:
    <expression>
    <expression>
    ...
elif <condition>:
    <expression>
    <expression>
    ...
else:
    <expression>
    <expression>
    ...
```



- Example 10: **control flow**

```
question = input("Please enter a fruit name: ")
if question == "apple":
    print("Yes, apple is a fruit")
elif question == "banana":
    print("Yes, banana is a fruit")
elif question == "strawberry":
    print("Yes, banana is a fruit")
else:
    print("Your input " + question + " isn't a fruit.")
```

'==' is a comparison operator as opposed to '=' sign which is an assignment operator.

- Comparison operators in Python:

Operator	Description	Example
==	If the values of two operands are equal, then the condition becomes true.	(a == b) is not true.
!=	If values of two operands are not equal, then condition becomes true.	
<>	If values of two operands are not equal, then condition becomes true.	(a <> b) is true. This is similar to != operator.
>	If the value of left operand is greater than the value of right operand, then condition becomes true.	(a > b) is not true.
<	If the value of left operand is less than the value of right operand, then condition becomes true.	(a < b) is true.
>=	If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.	(a >= b) is not true.
<=	If the value of left operand is less than or equal to the value of right operand, then condition becomes true.	(a <= b) is true.

- Example 11: **control flow continued**

New function:
`len()`

```
username = input("Your username: ")
password = input("Your password : ")
total_weight = len(username) + len(password)
message = "Your username and password has a total of {} characters!"
print(message.format(total_weight))
if total_weight > 40:
    print("The total length of your username and password ",
          "should not exceed 40 characters!")
else:
    print("Welcome to the system!")
```

len() returns the length of its argument.

- Example 12: **control flow continued – checking if two numbers are divisible**

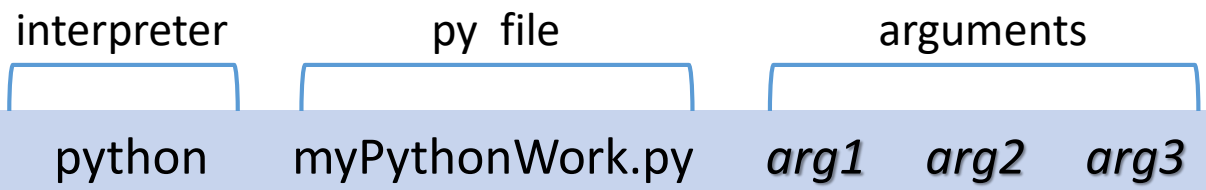
```
number1 = int(input("Please enter the number to be divided: "))
number2 = int(input("Please enter the divisor: "))
if number1 % number2 == 0:
    print("{} can divide {} without a remainder!".format(number2, number1))
else:
    print("{} can't divide {} without a remainder!".format(number2, number1))
```

'%' is a modulus operator which divides the left hand operand by the right hand operand and **returns the remainder**.

Command-line Arguments in Python

- Python provides a **getopt** module that allows you to use command line arguments.
- To access to any command-line argument you should use **sys** module.
- This modules provides two functionalities:
    1. **sys.argv** is the list of command-line arguments
    2. **len(sys.argv)** is the number of command-line arguments.

Example:



sys.argv[0] is the program file name, i.e., *myPythonWork.py*
sys.argv[1] is *arg1* whereas sys.argv[2] is *arg2*, and the like.

Exercises

1. Print a message which states whether a year which is taken as input is a leap year or not to the screen.

2. Convert a number which is taken from the input to binary format and print it to the screen.

3. Calculate the roots of $x^2 + bx + c = 0$ and print the result to the screen (b and c are taken from the input).

Exercise Solutions

```
"""Exercise-1 : Leap year"""

year = int(input("Please write a year to check whether it is a leap year or not\n"))

if year % 4 == 0 :
    print("{} is a leap year".format(year))
else :
    print("{} is not a leap year".format(year))
```

```
"""Exercise-2 : Number basen on 2"""

number = int(input("Please write a number\n"))
baseTwo =""

while number >= 2:
    baseTwo = str(number % 2) + baseTwo
    number = number // 2

baseTwo = str(number) + baseTwo
print(baseTwo)
```

```
"""Exercise-3 : Calculate of the roots of quadratic equation"""

numberB = int(input("Please write value of b\n"))
numberC = int(input("Please write value of c\n"))
numberA = 1

delta = pow(numberB , 2) - 4 * numberA * numberC

if delta > 0:
    root1 = (-numberB + pow (delta , 0.5)) / 2 * numberA
    root2 = (-numberB - pow (delta , 0.5)) / 2 * numberA

    print("Roots of the equation is {0} and {1}.".format(root1,root2))
elif delta ==0:
    root = (-numberB) / 2 * numberA

    print("Root of the equation is {}.".format(root))
else:
    print("There is no root in this equation")
```