

Introduction to Information Systems

Prepared by ENG. Eman Hamdy

جامعة الجميع الذكية

تسعى لتكون الكيان الأول الذي يتبنى مبدأ المجتمع التعليمي
التعاوني ويقدم العلم مقابل العمل لتنمية وبناء المجتمعات.

Python Programming Language



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

What is Programming?

1. Just like we humans can understand a few languages (Arabic, English, etc.), so is the case with computers.
2. Computers understand instructions that are written in a specific syntactical form called a programming language.
3. Simply, programming is a way to tell computers to do a specific task.
4. Some of the popular Programming languages are Python, C, C++, Java, etc

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

- It is used for:
 1. web development (server-side),
 2. software development,
 3. system scripting.

Why Python?

1. Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
2. Python has a simple syntax similar to the English language.
3. Python has syntax that allows developers to write programs with fewer lines than some other programming languages.

1. Python was designed for readability and has some similarities to the English language with influence from mathematics.
2. Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
3. Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Go to Download Python page on the official site <https://www.python.org/downloads/> and click Download Python 3.

- When the download is completed, double-click the file and follow the instructions to install it.
- When Python is installed, a program called IDLE is also installed along with it. It provides graphical user interface to work with Python.

- Creating Variables
- Variables are containers for storing data values.
- Unlike other programming languages, Python has no command for declaring a variable.
- A variable is created the moment you first assign a value to it.

```
x = 5  
y = "John"  
print(x)  
print(y)
```

```
5  
John
```

- Variables do not need to be declared with any particular type and can even change type after they have been set.

```
x = 4  
x = "Sally"  
print(x)
```

Sally

- String variables can be declared either by using single or double quotes:

```
x = "John"  
# is the same as  
x = 'John'
```

#Legal variable names:

```
myvar = "John"  
my_var = "John"  
_my_var = "John"  
myVar = "John"  
MYVAR = "John"  
myvar2 = "John"
```

#Illegal variable names:

```
2myvar = "John"  
my-var = "John"  
my var = "John"
```



- Assign Value to Multiple Variables

```
x, y, z = "Orange", "Banana", "Cherry"
```

```
print(x)  
print(y)  
print(z)
```

```
Orange  
Banana  
Cherry
```

```
x = y = z = "Orange"
```

```
print(x)  
print(y)  
print(z)
```

```
Orange  
Orange  
Orange
```

script.py

```
1 one = 1  
2 two = 2  
3 three = one + two  
4 print(three)  
5  
6 hello = "hello"  
7 world = "world"  
8 helloworld = hello + " " + world  
9 print(helloworld)
```

IPython Shell

```
3  
hello world
```

```
In [1]: |
```



- The Python print statement is often used to output variables.
- To combine both text and a variable, Python uses the + character:

```
x = "Python is "  
y = "awesome"  
z = x + y  
print(z)
```

```
Python is awesome
```

```
x = "awesome"  
print("Python is " + x)
```

```
Python is awesome
```

- If you try to combine a string and a number, Python will give you an error:

```
x = 5  
y = "John"  
print(x + y)
```

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
script.py  
1 # This will not work!  
2 one = 1  
3 two = 2  
4 hello = "hello"  
5  
6 print(one + two + hello)
```

IPython Shell

```
Traceback (most recent call last):  
  File "<stdin>", line 6, in <module>  
    print(one + two + hello)  
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Output Variables

- Create a variable outside of a function, and use it inside the function

```
x = "awesome"

def myfunc():
    print("Python is " + x)

myfunc()
```

```
Python is awesome
```

- Create a variable inside a function, with the same name as the global variable

-

```
x = "awesome"

def myfunc():
    x = "fantastic"
    print("Python is " + x)

myfunc()

print("Python is " + x)
```

```
Python is fantastic
Python is awesome
```



- Variables do not need to be declared with any particular type and can even change type after they have been set.

```
x = 4  
x = "Sally"  
print(x)
```

Sally

- String variables can be declared either by using single or double quotes:

```
x = "John"  
# is the same as  
x = 'John'
```

1. Python has commenting capability for the purpose of in-code documentation.
2. Comments start with a `#`, and Python will render the rest of the line as a comment.
3. you can add a multiline string (triple quotes) in your code, and place your comment inside it:
To add a **multiline comment** you could insert a `#` for each line

- In Python, the data type is set when you assign a value to a variable:

Example	Data Type
<code>x = "Hello World"</code>	str
<code>x = 20</code>	int
<code>x = 20.5</code>	float

There are three numeric types in Python:

- `int`
- `Float`
- `String`

- String literals in python are surrounded by either single quotation marks, or double quotation marks.
- `'hello'` is the same as `"hello"`.

#You can use double or single quotes:

```
print("Hello")  
print('Hello')
```

```
Hello  
Hello
```

```
x = 1  
y = 35656222554887711  
z = -3255522
```

```
print(type(x))  
print(type(y))  
print(type(z))
```

```
<class 'int'>  
<class 'int'>  
<class 'int'>
```

```
x = 1.10  
y = 1.0  
z = -35.59
```

```
print(type(x))  
print(type(y))  
print(type(z))
```

```
<class 'float'>  
<class 'float'>  
<class 'float'>
```



In programming you often need to know if an expression is **True** or **False**.

```
print(10 > 9)  
print(10 == 9)  
print(10 < 9)
```

```
True  
False  
False
```

Python Assignment Operators

- Assignment operators are used to assign values to variables

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3

Shortcut Operators

Operator	Equivalent to
<code>X += Y</code>	<code>X = X + Y</code>
<code>X -= Y</code>	<code>X = X - Y</code>
<code>X *= Y</code>	<code>X = X * Y</code>
<code>X /= Y</code>	<code>X = X / Y</code>
<code>X //= Y</code>	<code>X = X // Y</code>
<code>X **= Y</code>	<code>X = X ** Y</code>
<code>X %= Y</code>	<code>X = X % Y</code>

```

>>> x=1
>>> x+=1
>>> print(x)
2
>>> x-=1
>>> print(x)
1
>>> x*=8
>>> print(x)
8
>>> x/=6
>>> print(x)
1
>>> x=8
>>> x/=6
>>> print(x)
1.3333333333333333
>>> x=2
>>> x**=3
>>> print(x)
8

```

```

>>> val = 1
>>> val *= 8
>>> val += 34 % 10
>>> print(val)
12

```

```

>>> g = 4
>>> h = 3
>>> g **= h
>>> print(g)
64

```

```
X=input(" Enter a number : ")
```

user enter integer number

```
X=int(input(" Enter a number : "))
```

Enter more than one number

```
X,Y,Z,A= eval(input("Enter a number: "))  
print("X= ",X)  
print("Y= ",Y)  
print("Z= ",Z)  
print("A= ",A)
```

Import Math (Library)

```
import math
```

```
x=math.pow(2,3)
```

```
print("x=",x) ->x=8.0
```

```
import math
```

```
x=math.sqrt(9)
```

```
print("x=",x) ->x=3.0
```



Enter more than one number

$$Y = x + 1 * x + 2 * x + 3$$

```
import math
```

```
x=10
```

```
y=math.sqrt(x+1)* math.sqrt(x+2)*  
math.sqrt(x+3)
```

```
print("y=",y) ->y=41.424
```

$$Y = x^3 + 3x^2 + 3x + 1$$

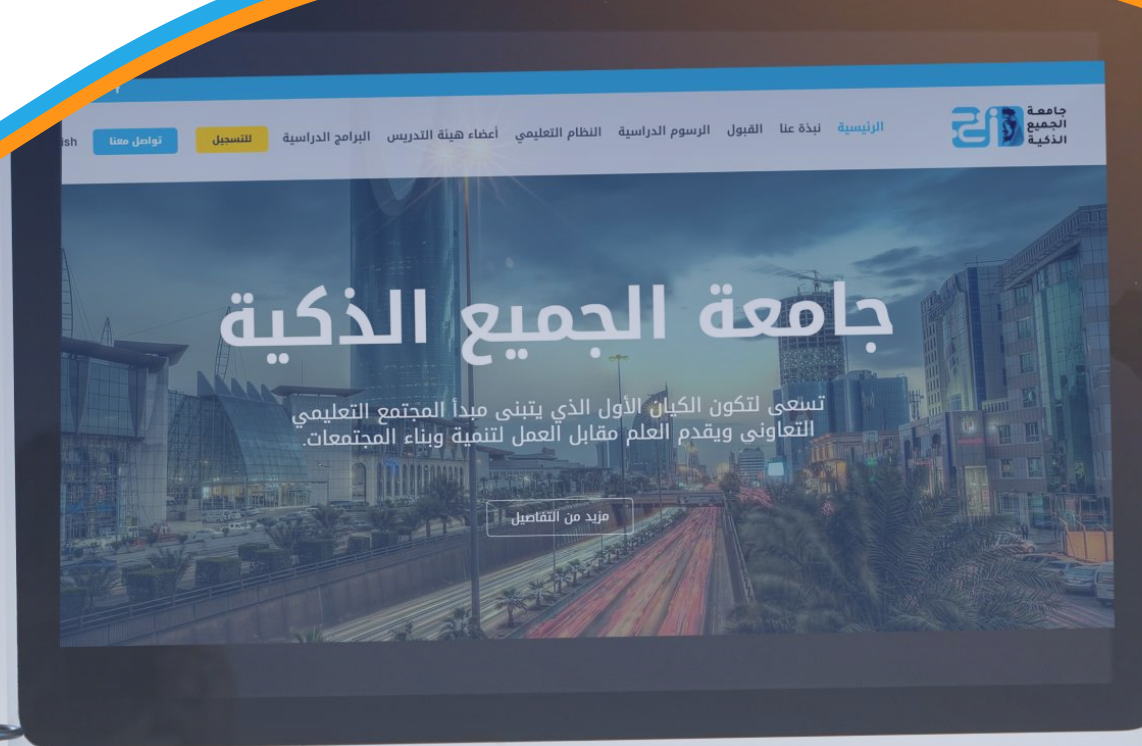
```
import math
```

```
x=10
```

```
y=math.pow(x,3)+ 3*x**2+ 3*x+1  
print("y=",y) ->y=1331.0
```



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY



Thank you