Section 1 Network Programming

Monday, February 18, 2019 10:46 AM

What's the network programming???

- Also called Socket Programming.
- Achieving the communication between an application on a pc and another application on another pc.
- EXAMPLES:-
  - Chat applications:-
    - Examples of applications in section:-
      - Concurrent response from the server to multiple clients.
      - Fork.
  - Time server:-

    For making a global time reference, not the pc time

    يعني إن الـ"server" يكون المهيمن على تحديد الوقت عشان على أساسه يحدد إمتى خدمة تبدأ أو تنتهي ... مش إن الكمبيوتر اللي بياخد الخدمة هو اللي يغير الوقت على مزاجه وياخد الخدمة اللي عاوزها ف أي وقت حتى لو منتهية.
  - Web servers:-

    Uploading the web-site on a server and retrieving its contents by any client.

What's the difference between compiled, and interpreted programs???

1. Compiled:-
   - Like C++
   - The compiler extract a .exe file.
   - Program isn't compiled again for execution.
   - A program compiled on a certain machine can't operate on different machine type.
     - That's called "machine dependent".

2. Interpreted:-
   - Every time gets interpreted for execution.
   - Can run on any machine ===> machine independent
   - But slow
   - Like python

3. Java is in the middle state, compiles till the part on machine dependent part which's done on the machine itself.

What's anaconda???

- Python language has versions ===> 2.7, 3.4, 3.5, 3.6, and 3.7
- IDE ===> Spyder is perfect for writing python applications
  - There's also Jupyter IDE.
- Python has different versions, Spyder has different versions.
- Anaconda performs the role of compatibility between the language, and the IDE

What's the structure of python program???

What's the difference between previously studied languages, and python???

- No semi-colon
- No brackets
- No datatype specification when declaring a variable:-
  - X=5 ===> it defines an integer variable
  - X = 5.0 ===> double
  - X = 'A' ===> String

The compiler knows the statement ends when the line actually ends "when the cursor moves to a lower line", if your statement took more than one line, you'd have to write a back slash (\ ) at the end of the line to indicate that the statement took more than usual.

What's about the brackets, how could we give it away???

Python is a language of indentation المسافات الفارغة أو محاذاة بداية السطور
"Making lines with the same beginning".

```
x=6
if (x>5):
    print (x)
    print ("Hello")
    x = 10
    print (x)
print ("python")
print (x)
```

```
In [9]: x=6
   ...: if (x>5):
   ...:     print (x)
   ...:     print ("Hello")
   ...:     x = 10
   ...:     print (x)
   ...:
   ...: print ("python")
   ...: print (x)
6
Hello
10
python
10
```

That's done with "if, while, for, method, and class declaration", or any block in general.

Object Types:-

- Mutable قابلة للتعديل
  - its contents are dynamic "can be changed (Add, delete, modify)"
  - List and Dictionary are examples of mutable object types.
    - They're like the ArrayList in java

1. List ===> identified by […..,…….,……]
   - Examples:-

```
y = ["Ibrahim","Karam",3]
print (y)
y.append("Mohammed")
print (y)
y.remove("Karam")
print (y)
del y[2]
print (y)
```

```
In [5]: y = ["Ibrahim","Karam",3]
   ...: print (y)
   ...: y.append("Mohammed")
   ...: print (y)
   ...: y.remove("Karam")
   ...: print (y)
   ...: del y[2]
['Ibrahim', 'Karam', 3]
['Ibrahim', 'Karam', 3, 'Mohammed']
['Ibrahim', 3, 'Mohammed']
```

2. Dictionary ===> a key, and a value
   - Example:-

```
y = {'name': "Ibrahim", "age": 24}
print (y)
```

```
In [3]: y = {'name':"Ibrahim", "age": 24}
   ...: print (y)
{'name': 'Ibrahim', 'age': 24}
```

   - For printing certain key's value:-

```
print (y['name'])
```

```
In [4]: y = {'name':"Ibrahim", "age": 24}
   ...: print (y['name'])
Ibrahim
```

   - There's no difference between single quotations and double quotations in Python.

- Immutable غير قابلة للتعديل
  - Int
  - Char
  - Double
  - String
  - Tuple ===> exactly like a list Z = (……..,……..,……)

For commenting:-

- # for one line
- Three-double quotations at the beginning and the end of multi-lined comment

  """

  Comment statement(s)

  ….

  ….

  """

```
"""
non
executed
statements
"""
```

How to convert a list to a tuple, and vice versa???

- Z = list (z)
- Example:-

  Z = ('Mohammed', 'Ibrahim')

  print (Z)

  Z = list (Z)

  print (Z)

  Z.append ('karam')

  print (Z)

```
In [6]: Z = ('Mohammed', 'Ibrahim')
   ...: print (Z) ===>"mainly immutable"
   ...: Z = list (Z) ===>"casting to mutable list"
   ...: print (Z)
   ...: Z.append ('karam') "allowed append process"
   ...: print (Z)
('Mohammed', 'Ibrahim')
['Mohammed', 'Ibrahim']
['Mohammed', 'Ibrahim', 'karam']
```

- Tuple (z)

```
y = ["Ibrahim","Karam",3]
print (y)
y.append("Mohammed")
print (y)
y.remove("Karam")
print (y)
del y[2]
print (y)
tuple (y)
y.app..... "can't append"
```

- What happens when declaring
  1. x=50,
  2. Then x=60,
  3. Then x=50, again???
  1. It declares an object of the class int called 'x' with value 50, has a certain id,
  2. Then declares another object also called 'x', has a different id, points to it leaving the first one.
  3. Then turning back to the object with value 50 with the same first id.

```
In [7]: x=50
   ...: print (id(x))
   ...: x=60
   ...: print (id(x))
   ...: x=50
   ...: print (id(x))
   ...: print (type(x))
140710590724464
140710590724784
140710590724464
<class 'int'>
```

Could we execute only some statements of the whole written program in the IDE???

Yes, by selecting the needed statements to be executed,
Then click an hold the ctrl button, plus the Enter button.