



مقرر التحليل العددي

د. يمار الحموي

م. اية خيربك

م. ندى جنيدي

العملي

الفصل الثاني 2022-2023

في البايثون لا تحتاج الـ
الى الـ enter لاعلان
نهاية الـ statement

• >>> print()

print() without arguments results in a blank line ⇔ '\n' # Blank line

```
print("Python is powerful")
```

```
print('Python is powerful')
```

Output Statement

عبارة الخرج

VARIABLE
↓

```
>>> message = 'Please wait while the program is loading...'
```

```
>>> print(message)
```

Output Statement

عبارة الخرج

```
print('Good Morning!', end= ' ') # print with end whitespace
```

```
    '\n' # Blank line
```

```
    ' ' # Empty line
```

Inline Comments



```
print(What do you call bear with no teeth ? A gummy bear\n')  
What do you call bear with no teeth ? A gummy bear
```

```
print(What do you call bear with no teeth ?\n A gummy bear')  
What do you call bear with no teeth ?  
A gummy bear
```

```
print('Welcome\nto\n\nPython!')  
Welcome  
To  
  
Python!
```

```
print('\nWelcome\nto\nPython!')  
  
Welcome  
To  
Python!
```

Output Statement

عبارة الخرج

- `print('Sum is', 7 + 3)`

Sum is 10

- `print('Display "hi" in quotes')`

Display "hi" in quotes

- `print('Display 'hi' in quotes')`

File "<ipythoninput219bf596ccf72>", line 1

```
print('Display 'hi' in quotes')
```

^

SyntaxError: invalid syntax

```
print("""Display "hi" and 'bye' in quotes""")
Display "hi" and 'bye' in quotes
```

```
print('Display \'hi\' in quotes')
Display 'hi' in quotes
```

I will add that " is useful when ' is inside the string, and vice versa.

Output Statement

عبارة الخرج

- Ignoring a Line Break in a Long String

```
>>> print('this is a longer string, so we \
... split it over two lines')
```

this is a longer string, so we split it over two lines

Backslash(\) is the continuation character

ليس من الضروري ظهور هذه النقاط ...
في جميع البيئات الخاصة بالبايثون
عند كتابة البرنامج

Within Python, a **backslash** (\) means a continuation character.

Also, if it is set at the edge of a line, it is estimated that the line is continued, disregarding following newlines.

تستخدم continuation character مع أنواع
الـ quotes الثلاثة

Output Statement

عبارة الخرج

```
triple_quoted_string = """This is a triplequoted  
... string that spans two lines"""  
print(triple_quoted_string)
```

This is a triplequoted
string that spans two lines

عند استخدام الـ triple quotes و بمجرد ضغط
enter بعد كل سلسلة محرفية يفهم الـ python
interpreter ان السلاسل المحرفية سيتم فصلها
بحيث تقع كل سلسلة على سطر.
اي يتم تفعيل خاصية الـ continuation
prompt و ذلك فقط مع الـ triple quotes

Variable Names

أسماء المتحولات

• اسم المتحول (A variable name), مثال x , يمثل معرفا (identifier) لهذا المتحول.

- A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume).
- Rules for Python variables:
 - A variable name must start with a letter or the underscore character
 - A variable name cannot start with a number
 - A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and `_`)
 - Variable names are case-sensitive (age, Age and AGE are three different variables)

Variable Names

أسماء المتحولات

- #Legal variable names: أسماء المتحولات المسموحة
- myvar = "John"
- my_var = "John"
- _my_var = "John"
- myVar = "John"
- MYVAR = "John"
- myvar2 = "John"

- #Illegal variable names: أسماء المتحولات غير المسموحة
- 2myvar = "John"
- my-var = "John"
- my var = "John"

Input Statement

عبارة الدخل

```
name = input("Enter your name please: ")  
print(name)
```

عبارة الدخل مع رسالة نصية

```
name=input()  
print("inter your name please :",name)
```

عبارة الدخل بدون رسالة نصية

Input Statement

عبارة الدخل

التحكم بنمط الدخل: مشكلة تابع الدخل input أنه يعيد دائما قيمة من النمط str من أجل ذلك نحتاج الى اعادة نمط الدخل الى النمط الذي نحتاج اليه.

```
x=input('input your name plz:\n')  
print(x)  
type(x)
```

output

```
input your name plz  
Nada  
Nada  
str
```

```
x=input('input your Id  
number plz:\n')  
print(x)  
type(x)
```

output

```
input your Id number plz  
98  
98  
str
```

Input Statement

عبارة الدخل

التحكم بنمط الدخل: رأينا أن مشكلة تابع الدخل `input` أنه يعيد دائما قيمة من النمط `str`, ماذا لو كنا نريد أن يدخل المستخدم قيما صحيحا؟, من أجل ذلك نحتاج الى اعادة نمط الدخل الى النمط الذي نحتاج اليه.

```
num1=int(input('Enter integer value please:'))
num2=int(input('Enter integer value please:'))
answer=num1+num2
print('The answer is=',answer)
```

هذه العملية تدعى عملية
الـCasting و تعني قسر
النمط الى نمط اخر

output

```
Enter integer value please:7
Enter integer value please:8
The answer is= 15
```

Input Statement

عبارة الدخل

التحكم بنمط الدخل: لو لم نستخدم طريقة الـ casting لأصبح المثال السابق ينفذ كما يلي:

```
num1=input('Enter integer value please:')  
num2=input('Enter integer value please:')  
answer=num1+num2  
print('The answer is=',answer)
```

لاحظ هنا كيف اعتبرت الارقام
سلاسل محرفية, ولاحظ كيف لعبت
عملية الجمع هنا دور أداة الوصل
[concatenation] بين سلسلتين
محرفيتين

```
Enter integer value please:7  
Enter integer value please:8  
The answer is= 78
```

Input & Output

عبارة الدخل وعبارة الخرج

```
firstname = input ("Please enter your first name: ")  
surname = input ("Please enter your surname: ")  
print ("Hello", firstname, surname)
```

```
kilo = int (input ("Please enter the number of kilos: "))  
pound = kilo*2.204  
print ("That is ", pound, "pounds ")
```

Python Data types

Numeric Data Types	String Data Types	Boolean Type
<p>Integer -,0,+ >>> print(42) # <class 'int'> 42</p> <p>Note 8 ≠ 8.0 integer float</p>	<p>Str >>> print('hello') # <class 'str'> hello</p> <div data-bbox="1072 739 1679 882" style="border: 1px solid black; padding: 5px;"> <pre>Type (Rami) str</pre> </div>	<p>Bool True False >>> print(True) # <class 'bool'> True</p> <p>>>>my_bool = 5 > 8 >>>print(my_bool) False</p>
<p>Float 8.0 >>> print(3.14) # <class 'float'> 3.14</p>	<div data-bbox="1072 1053 1342 1248" style="border: 1px solid black; padding: 5px;"> <pre>x=7+6j type(x) complex</pre> </div> <div data-bbox="1375 958 1719 1152" style="border: 1px solid black; padding: 5px;"> <pre>x=8 type(x) Int</pre> </div>	
<p>Complex 5+9J >>> print(1 + 2j) # <class 'complex'> (1+2j)</p>	<div data-bbox="1375 1225 1719 1358" style="border: 1px solid black; padding: 5px;"> <pre>type(10.5) float</pre> </div>	

Python Data types

Sequence Types	Set Data Types
<p>List</p> <pre>>>> print([1, 2, 3]) # <class 'list'> [1, 2, 3]</pre>	<p>Set</p> <pre>>>> print({'red', 'green', 'blue'}) # <class 'set'> {'red', 'green', 'blue'}</pre>
<p>Tuples</p> <pre>>>> print((1, 2, 3)) # <class 'tuple'> (1, 2, 3)</pre>	<p>frozenset</p>
<p>Range</p> <pre>range(start, stop, step) >>>range(2,6,2) 2,4 >>>range(25, 0 , -5) # reverse range 25, 20, 15, 10, 5 >>>range(10)</pre>	

Python Data types

Mapping Data Types

Dict

```
>>> print({'name': 'Alice', 'age': 42}) # <class 'dict'>  
{'name': 'Alice', 'age': 42}
```