

الدرس دا هيتكلم عن ال **Functions & return**، وطبعا معروف اهمية الفانكشن فكل لغات البرمجه والفانكشن بتعمل ايه، المختلف عندنا فى البايثون هو اختلاف ال syntax، لكن هعمل مراجعه عنها علشان ننشط الدماغ.

ال function هى عبارته عن كود بعمله واديله اسم علشان لما استدعى الاسم ف اى مكان فى البرنامج بتاعى ينفذلى جزء الكود اللى جوا ال function دى، علشان فى اجزاء اكواد بحتاجها اكتر من مره فى البرنامج فمش كل شويه هقعد اكتبها فى البرنامج فانا بستدعيها باسمها اللى سميته ليها.

الفانكشن بتتنفذ لما اناديلها او زى م قلت اذكر اسمها ف اى مكان فى البرنامج او فى نوع تانى ودى اللى بتنفذ نفسها اسمها self-invoke function هنعرفها بعدين.

الفانكشن وانا بعرفها بيتكتب ف تعريف الاقواس بتوعها متغيرات بستخدمها فى الكود اللى جوا الفانكشن وفى الحاله دى (اللى هى حالة تعريف الفانكشن) بيتسموا parameters وممكن مضيفش متغيرات خالص لو الفانكشن مش محتاجه تتعامل مع متغيرات.

لما بنادى على الفانكشن (function call) ف اى حته فى البرنامج بحط قيم للمتغيرات (parameters) اللى عرفتها وانا بعرف الفانكشن اللى عايز يطبق عليها الكود اللى جوا الفانكشن وفى الحاله دى بيبقا اسمهم arguments.

الفانكشن ممكن اخليها تطلعلى output بانى اكتب كود print جواها او اخليها ترجع الداتا من غير طباعه ودا الاستخدام الاشهر والادق للفانكشنز، وهنا بيجى دور ال **return**، ودى بترجعلى الخرج اللى عايز اخرجه من الفانكشن لما انادى عليها من غير طباعه وفى الحاله دى ممكن اعرف متغير واساويه بالفانكشن وهتبقا قيمته قيمة خرج الفانكشن (السطر القبل الاخير فى الكود).

فى عندنا built-in functions ودى اللى مبنيه فى اللغه واسمها معروف واللى استخدمناها قبل كدا وبتنفذ مهمات معروفه زى upper() و capitalize() وكل الفانكشنز اللى زى كدا.

وفى ال user-defined functions ودى اللى انا بعملها بقا وبسميها وبحدد بتنفذ ايه ودا اللى كنا بنتكلم عنه فى الدرس دا.

اظن مفيش كلام بعد كدا عن الفانكشنز، دى مكانتش مراجعه دا كان شرح، اللى عمل الفانكشنز نفسه ميعرفش انها بتعمل كل دا.

```
# -----
# -- Function And Return --
# -----
# [1] A Function is A Reusable Block Of Code Do A Task
# [2] A Function Run When You Call It
# [3] A Function Accept Element To Deal With Called [Parameters]
# [4] A Function Can Do The Task Without Returning Data
# [5] A Function Can Return Data After Job is Finished
# [6] A Function Create To Prevent DRY (Don't Repeat Yourself)
# [7] A Function Accept Elements When You Call It Called [Arguments]
# [8] There's A Built-In Functions and User Defined Functions
# [9] A Function Is For All Team and All Apps
# -----

def function_name():

    return "Hello Python From Inside Function"
```

```
dataFromFunction = function_name()  
  
print(dataFromFunction)
```

#### CODE

```
exe "c:/Users/Muhammad/Documents/Python Course/first.py"  
Hello Python From Inside Function
```

#### OUTPUT