

الدرس دا هيتكلم عن ال **Function Recursion**، وهو ان الفانكشن بتنادى نفسها (يعنى انا بنفذ الفانكشن جوا كود الفانكشن نفسه)، ممكن تكون عندى زى كونسبت ال loop وتعمل infinite loop لو انا مش كاتب الكود بتاعى صح.

المثال الموجود فى الكود هو عبارته عن فانكشن بتنضفلى الكلمه من الحروف المكرره اكثر من مره وترجعلى الكلمه سليمه من غير تكرار، هو مثال متقدم شويه فى ال recursion بس فهمك ليه يوضح جزء كبير من موضوع ال recursion، وانا هحاول اوضح فيه على قدم اقدر.

عندى الفانكشن متقسمه ل 3 اجزاء، اول if بتعمل check هل الكلمه مكونه من حرف واحد؟ لو اه يبقا اكيد مفيش تكرار ورجعلى الكلمه بتاعتي (اللى هى الحرف الواحد) وخلصى ع ال if اللى بعدها.

تاني if بتعمل check، هل الحرف الاول يساوى الحرف التانى ولا لا؟ لو اه يبقا اكيد فى تكرار واللى انا محتاج اعمله انى اشيل حرف من الاتنين، فابستخدام فكرة ال slicing اللى درسناها فى دروس ال formatting، بعد م عرفت ان فى تكرار ف اول حرفين هسيب اول حرف واخد الكلمه من اول الحرف التانى لآخر الكلمه، ويبقا كدا عندى اتخلصت من حرف من المتكررين، واخد الكلمه لما عملتلها slicing ادخلها فى parameter الفانكشن نفسها تانى علشان تنفذ بقاى الخطوات وال checks وهو دا ال recursion انى قلت للفانكشن بتنفذ تانى وانا جواها.

لما استدعى الفانكشن تانى هتبدأ هى تعيد نفسها من الاول على الكلمه الجديده اللى اديتها ليها (اللى معمولها slicing).

طيب خلصت الحروف الاولى من التكرار يجى دور الجزء التالت وهو انى اخد اول حرف (اللى مش متكرر) وارجعه واعمله concatenation مع الفانكشن تانى بس واخده parameter لباقى الكلمه اللى لسه معمولتلهاش check علشان تبدأ الفانكشن بتنفذ عليها من الاول وكل اما يطلعلى حرف مش متكرر ف اول الكلمه هيتاخد ويتجمع على الحرف اللى كان مش متكرر قبل كدا لحد م تطلعلى الكلمه من غير تكرار.

طبيعى انك تكون مش مستوعب 100% لان موضوع ال recursion بيكون لاف شويه فحاول تدقق فالكود وتمشى معاه واحده واحده وتشوف خرج كل مرحله وازاى هيتربط باللى بعدها. وممكن تشوف امثله تانيه وشروحات وتدرّب نفسك اكثر على الموضوع لحد م تتقنه.

```
# -----
# -- Function Recursion --
# -----
# -----
# -- To Understand Recursion, You Need to First Understand Recursion --
# -----

# Test Word [ WWWooooorrrldd ] # print(x[1:])

def cleanWord(word):

    if len(word) == 1:

        return word

    print(f"Print Start Function {word}")
```

```

if word[0] == word[1]:

    print(f"Print Before Condition {word}")

    return cleanWord(word[1:])

print(f"Print Before Return {word}")

return word[0] + cleanWord(word[1:])

# Stash [ World ]

print(cleanWord("WWWooooorrrldd"))

```

CODE

```

exe "c:/Users/Muhammad/Documents/Python Course/first.py"
Print Start Function WWWooooorrrldd
Print Before Condition WWWooooorrrldd
Print Start Function Wooooorrrldd
Print Before Condition Wooooorrrldd
Print Start Function Wooooorrrldd
Print Before Return Wooooorrrldd
Print Start Function oooooorrrldd
Print Before Condition oooooorrrldd
Print Start Function oooooorrrldd
Print Before Condition oooooorrrldd
Print Start Function oooooorrrldd
Print Before Condition oooooorrrldd
Print Start Function oooooorrrldd
Print Before Condition oooooorrrldd
Print Start Function oooooorrrldd
Print Before Return oooooorrrldd
Print Start Function rrrrldd
Print Before Condition rrrrldd
Print Start Function rrrrldd
Print Before Condition rrrrldd
Print Start Function rrrrldd
Print Before Return rrrrldd
Print Start Function rldd
Print Before Condition rldd
Print Start Function rldd
Print Before Return rldd
Print Start Function ldd
Print Before Return ldd
Print Start Function dd
Print Before Condition dd
World

```

OUTPUT