

في الدرس دا هنتكلم عن ال built-in function **zip()**، وظيفتها انها بتعمل iterables مكونه من العناصر المناظره لبعضها في مجموعة iterables.

يعنى على سبيل المثال انا عندي اثنين list بدخلهم جوا الفانكشن فهي بتاخذ اول عنصر من ال list الاولى مع اول عنصر من الثانيه وتانى عنصر من الاولى مع تانى عنصر من الثانيه وهكذا وبتحطلي كل عنصرين مع بعض ف tuple ويكون مكون عندي iterable تالت بيضم كل ال tuples المكونه من العنصرين المناظرين لبعض.

لو عندي اكثر من اثنين list هياخذ العنصر الاول من كل واحده ويحطهم ف tuple وتانى عنصر من كل واحده ويحطهم ف tuple وهكذا.

لو عملت loop على المتغير اللي خليته يشيل قيمة ال zip() هixرجلي ال tuples اللي جمعهم من كل ال lists.

طيب لو ال lists مش متساويين فعدد العناصر ؟ بيعمل tuples على حسب عدد العناصر اللي في اقل ال list.

```
# -----
# -- Practical => Loop on Many Iterators With Zip() --
# -----
# zip() Return A Zip Object Contains All Objects
# zip() Length Is The Length of Lowest Object
# -----

list1 = [1, 2, 3, 4, 5]
list2 = ["A", "B", "C", "D"]
tuple1 = ("Man", "Woman", "Girl", "Boy")
dict1 = {"Name": "Muhammad", "Age": 36, "Country": "Egypt", "Skill": "Python"}

test = zip(list1, list2)

for item in test:
    print(item)

for item1, item2, item3, item4 in zip(list1, list2, tuple1, dict1):

    print("List 1 Item =>", item1)
    print("List 2 Item =>", item2)
    print("Tuple 1 Item =>", item3)
    print("Dict 1 Key =>", item4, "Value =>", dict1[item4])

ultimateList = zip(list1, list2)
print(ultimateList)
for item in ultimateList:
    print(item)
```

CODE

```
n Course/first.py"
(1, 'A')
(2, 'B')
(3, 'C')
(4, 'D')
List 1 Item => 1
List 2 Item => A
Tuple 1 Item => Man
Dict 1 Key => Name Value => Muhammad
List 1 Item => 2
List 2 Item => B
Tuple 1 Item => Woman
Dict 1 Key => Age Value => 36
List 1 Item => 3
List 2 Item => C
Tuple 1 Item => Girl
Dict 1 Key => Country Value => Egypt
List 1 Item => 4
List 2 Item => D
Tuple 1 Item => Boy
Dict 1 Key => Skill Value => Python
<zip object at 0x000001FD2B423040>
(1, 'A')
(2, 'B')
(3, 'C')
(4, 'D')
```

OUTPUT