

setdefault()

بترجع القيمة بتاع ال key اللى انا محدده كأول argument فى الفانكشن، لو ال key مش موجود اصلا ف ال dict بتضيفه وبتضيفه القيمة اللى بحطها ك argument تانى فى الفانكشن.

popitem()

بترجعلي اخر key انا ضيفته ف ال dict بالقيمة بتاعته.

items()

شفنا ف الدروس اللى فاتت فانكشن keys() كانت بترجعلي ال keys اللى جوا ال dict وال values() بترجعلي القيم اللى جوا ال dict، عندنا بقا items() بيرجع العناصر كلها، كل المفاتيح بالقيم بتاعتها وينفع اعملهم assign لمتغير تاني. بيرجعهم على شكل ليست وكل key وال value بتاعته محطوتين جوا ال list دى على شكل tuple.

fromkeys()

بتعمل assign لاكثر من key بنفس القيمة ليهم كلهم.

يعنى لو عندي tuple فيه اكثر من عنصر حابب اخليهم عبارة عن keys جوا dict وكلهم ليهم نفس القيمة بعرف متغير بالقيمة اللى عايز اضيفها لكل ال keys واكتب dict.fromkeys(a,b). الكود اوضح.

```
# -----
# -- Dictionary Methods --
# -----

# setdefault()

user = {
    "name": "Muhammad"
}
print(user)
print(user.setdefault("name", "Abusenna"))
print(user.setdefault("age", 36))
print(user)

print("=" * 40)

# popitem()

member = {
    "name": "Muhammad",
    "skill": "PS4"
}
print(member)
member.update({"age": 36})
print(member.popitem())

print("=" * 40)
```

```

# items()

view = {
    "name": "Muhammad",
    "skill": "XBox"
}

allItems = view.items()
print(view)
view["age"] = 36

print(allItems)

print("=" * 40)

# fromkeys()

a = ('MyKeyOne', 'MyKeyTwo', 'MyKeyThree')
b = "X"

print(dict.fromkeys(a, b))

```

CODE

```

exe "c:/Users/Muhammad/Documents/Python Course/first.py"
{'name': 'Muhammad'}
Muhammad
36
{'name': 'Muhammad', 'age': 36}
=====
{'name': 'Muhammad', 'skill': 'PS4'}
('age', 36)
=====
{'name': 'Muhammad', 'skill': 'XBox'}
dict_items([('name', 'Muhammad'), ('skill', 'XBox'), ('age', 36)])
=====
{'MyKeyOne': 'x', 'MyKeyTwo': 'x', 'MyKeyThree': 'x'}

```

OUTPUT