**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**

ЛАБОРАТОРНА РОБОТА № 6

**Виконав студент:**

групи БСД – 22

Муха Максим

Володимирович

**Київ – 2022**

**Тема: робота зі списками та бібліотеками через обробники.**

**Мета роботи:** отримати навички роботи зі синтаксисом, який використовується в Python для роботи зі списками та бібліотеками.

**Списки**

**Код програми**

fruits = ["apples", "bananas", "strawberies"]

for fruit in fruits:

   print(fruit)

[ print(i) for i in fruits ]

for fruit in fruits:

   fruit = fruit.upper()

print(fruits)

new\_fruits = []

for fruit in fruits:

   fruit = fruit.upper()

   new\_fruits.append(fruit)

fruits = new\_fruits

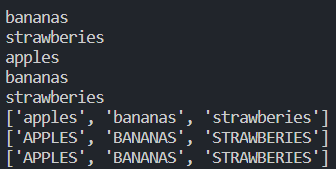
print(fruits)

fruits = ["apples", "bananas", "strawberies"]

fruits = [fruit.upper() for fruit in fruits]

print(fruits)

В цьому прикладі, ми використали скорочений синтаксис, ми напряму працюємо зі списком в пам’яті та змінюємо всі значення майже одночасно.

****

**Рис.1.** Результат

Інші приклади:

З першого прикладу замінемо слова на цифри (1, 0)

bits = [False, True, False, False, True, False, False, True]

new\_bits = []

for b in bits:

   if b == True:

       new\_bits.append(1)

   else:

       new\_bits.append(0)

print(bits)

print(new\_bits)



**Рис.2.** Результат роботи програми

Також можемо записати трішки скороченою формою

bits = [False, True, False, False, True, False, False, True]

new\_bits = []

super\_bits = [ 1 if b == True else 0 for b in bits ]

print(bits)

print(super\_bits)



**Рис.3.** Результат роботи програми

Ми також можемо використовувати й строковий тип даних

my\_string = "HelloMyNameIsVasia"

my\_string = "".join(

   [ i if i.islower() else " " + i for i in my\_string ]

   )[1:]

print(my\_string)



**АБО**

my\_string = "helloMyNameIsvasia"

my\_string = "".join(

   [ i if i.islower() else " " + i for i in my\_string ]

   )[1:]

print(my\_string)

****

**Словники**

За допомогою списків, ми можемо об’єднувати списки в словники.

Для цього використаємо функцію **zip()**.

names = ['John', 'Kate', 'Martin']

profs = ['programmer', 'dancer', 'writer']

my\_dict\_1 = {}

for (key, value) in zip(names, profs):

   my\_dict\_1[key] = value

print(f"#1 dict1 {my\_dict\_1}")

my\_dict\_1 = {}

my\_dict\_1 = {key:value for (key, value) in zip(names, profs)}

print(f"#2 dict1 {my\_dict\_1}")



**Рис.4.** Результат роботи програми

А тепер без функції **zip()**.

names = ['John', 'Kate', 'Martin']

profs = ['programmer', 'dancer', 'writer']

my\_dict\_2 = {}

for i in range(3):

   my\_dict\_2[names[i]] = profs[i]

print(f"#1 my\_dict\_2 {my\_dict\_2}")

my\_dict\_2 = {}

my\_dict\_2 = {names[i]:profs[i] for i in range(len(names))}

print(f"#1 my\_dict\_2 {my\_dict\_2}")



**Рис.5.** Результат роботи програми

**Завдання**

Добавляємо функцію replace(" ", "\_")) для заповнення пропуску символом

« \_ »

keys = ["id", "name", "username", "email"]

users = ["Harry Potter", "Ronald Wisley", "Hermyni Granger"]

data = [

       {

           key:(i if key == 'id'

               else users[i] if key == "name"

               else (users[i][0:3].lower() + users[i][-3:].lower()) if key == "username"

               else ("".join(j for j in users[i]).lower().replace(" ", "\_"))+"@gmail.com")

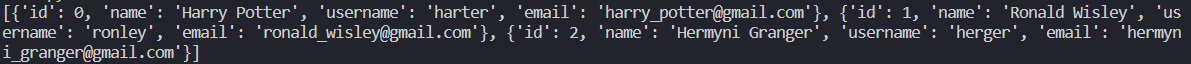
           for key in keys

       }

       for i in range(len(users))

       ]

print(data)

****

**Рис.6.** Результат виконання