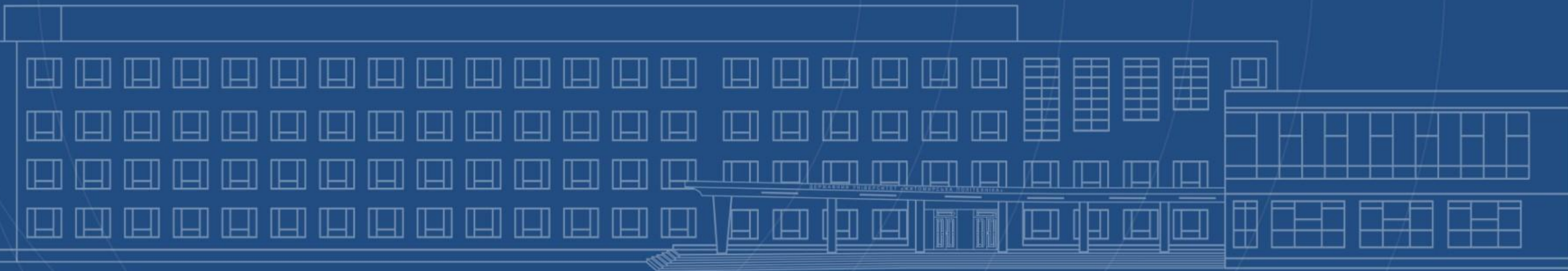


АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРИЗОВАНИХ ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНИХ СИСТЕМ

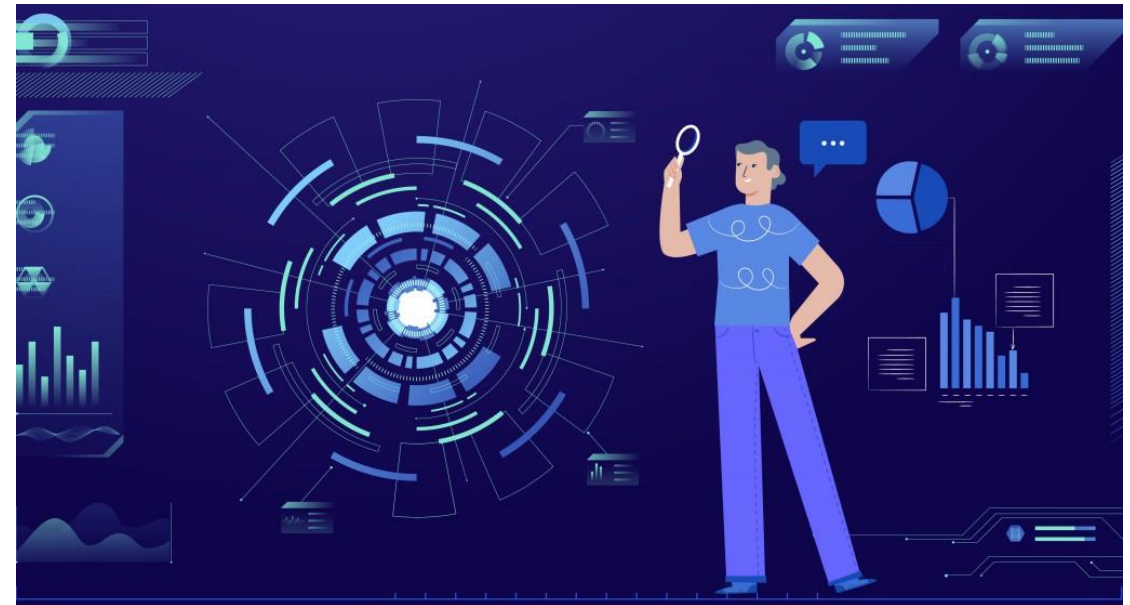


Лекція 11

Тема: Запис і переміщення даних

1. Читання і запис даних в файл.

2. Переміщення по файлу.



1. Читання і запис даних в файл

Для читання файлу він відкривається з режимом **r** (Read), і потім ми можемо отримати його вміст різними методами:

read (): зчитує весь вміст файлу в один рядок;

read (n): зчитує **n** байтів від положення маркера

readline (): зчитує один рядок з файлу;

readlines (): зчитує всі рядки файлу в список.

```
with open("hello.txt", "r") as file:
```

```
    str1 = file.readline()
```

```
    print(str1, end="")
```

```
    str2 = file.readline()
```

```
    print(str2)
```

```
with open("hello.txt", "r") as file:
```

```
    for line in file:
```

```
        print(line, end="")
```

Щоб прочитати потрібний рядок :

```
a = open('ім'я файлу').read() .split('\n')[номер рядка(з нуля)]
```



Метод `readline` можна використовувати для порядкового зчитування файлу в циклі `while`

```
with open("hello.txt", "r") as file:
    line = file.readline()
    while line:
        print(line, end="")
        line = file.readline()
```

#Якщо файл не великий можна використати `read()`

```
with open("hello.txt", "r") as file:
    content = file.read()
    print(content)
```

При читанні файлу ми можемо зіткнутися з тим, що його кодування не збігається з ASCII. В цьому випадку ми явно можемо вказати кодування за допомогою параметра `encoding`:

```
filename = "hello.txt"
with open(filename, encoding="utf8") as file:
    text = file.read()
```

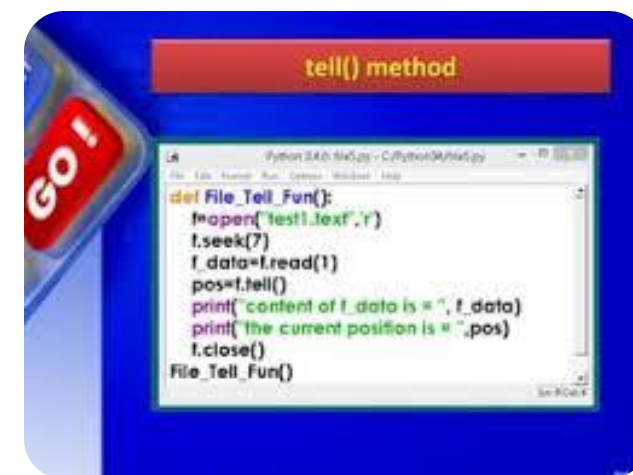


Метод tell

Якщо повторно викликати метод `read()` на файловому об'єкті (не виконуючи команду `close()`) отримаємо лише порожній рядок. Це відбувається тому, що після першого прочитання покажчик знаходиться наприкінці файлу. Щоб дізнатися поточну позицію покажчика можна використовувати метод `tell()`.

```
my_file = open("some.txt")  
my_file.read(10)  
print («Покажчик знаходиться на позиції:», my_file.tell())  
my_file.close()
```

метод `tell()` повідомляє, в скількох байтах від початку файлу знаходиться покажчик.



Метод seek(n)

```
my_file.seek(offset, [from])
```

Аргумент **offset** вказує на скільки байт перейти. Необов'язковий аргумент **from** означає позицію, з якої починається рух. 0 означає початок файлу, 1 нинішня позиція, 2 кінець файлу.

```
my_file = open("some.txt", "r")
print(my_file.read(10))
print(«Вказівник на позиції: ", my_file.tell())
# Повертаємося на початок
my_file.seek(0)
print(my_file.read(10))
my_file.close()
```



Запис у текстовий файл

Щоб відкрити текстовий файл на запис, необхідно застосувати режим **w** (перезапис) або **a** (дозапис). Потім для запису застосовується метод **write (str)**, в який передається (записується) рядок.

!!!! Варто відзначити, що записується саме рядок, тому, якщо потрібно записати числа чи дані інших типів, то їх попередньо потрібно конвертувати в рядок.

```
with open("hello.txt", "w") as file:  
    file.write("hello world")
```

```
with open("hello.txt", "a") as file:  
    file.write("\ngood bye, world")
```



```
f = open("ex.txt", "w") # відкриваємо в режимі запису/створення
```

```
f.write("fff hhh kkk\n")
```

```
f.write("fff hhh kkk")
```

```
f.close()
```

```
f1= open("ex.txt", "r") # відкриваємо в режимі читання
```

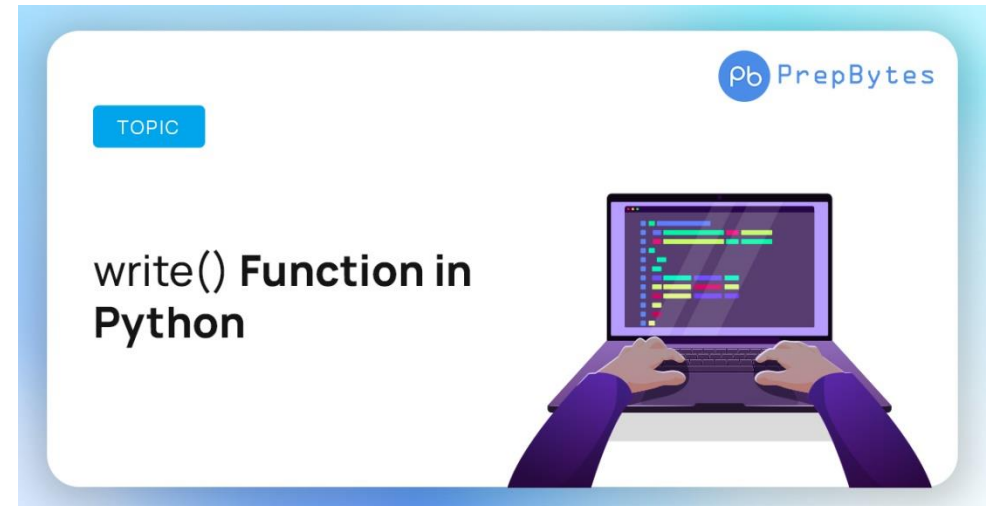
```
for k in f1: # text=f1.read()
```

```
    print(k) # print(text)
```

Результат:

```
fff hhh kkk
```

```
fff hhh kkk
```



2. Переміщення по файлу

Файл використовується для постійного зберігання даних. Іноді для цілей програмування нам потрібно перемістити розташування файлу з одного шляху на інший. Це завдання можна виконати за допомогою сценарію Python різними способами. **Перемістити** () є найбільш використовуваним методом Python для переміщення файлу з одного каталогу в інший, визначений у **шутіл** модуль.



Інший спосіб переміщення розташування файлу за допомогою **перейменувати** () метод, який визначений у **os** модуль. Ці два методи можна використовувати для переміщення файлу з одного каталогу в інший, як пояснюється в цьому посібнику.

Спосіб переміщення файлу з одного місця в інше з оригінальною назвою показаний у наступному сценарії. Шутіл модуль імпортується у сценарій для використання `move ()` функція переміщення файлу. Шлях модуль імпортується для використання `існує ()` функція перевірки даного імені файлу існує чи ні. Якщо файл існує, буде визначено шлях призначення файлу, куди файл буде переміщено. Місце розташування буде надруковано після переміщення файлу. Якщо файл не існує, буде надруковано повідомлення про помилку.

```
import shutil

old_file = '/Users/datagy/new_name.txt'
destination = '/Users/datagy/new_folder'

shutil.move(old_file, destination)
```

Обробка файлів у Python за допомогою модуля **os** включає створення, перейменування, переміщення, видалення файлів та папок, а також отримання списку всіх файлів та каталогів та багато іншого.

Модуль **os** вбудований, тому для роботи з ним нічого не потрібно встановлювати.

getcwd() - виведення поточної директорії

```
import os
# вивести текущу директорію
print(«Поточна директорія:", os.getcwd())
```

```
# Get Current Working Directory with os.getcwd()
import os
cwd = os.getcwd()

print('Current Working Directory is: ', cwd)
# Returns: Current Working Directory is: /Users/datagy
```

```
1 # Import the os module
2 import os
3
4 # Get a working directory with os.getcwd()
5 cwd = os.getcwd()
6
7 # Set a working directory
8 os.chdir('/Users/datagy/Documents') # Using an absolute path
9 os.chdir('../Data') # Using a relative path
```

mkdir ("назва папки") - створення каталогу (папки, директорії)

```
os.mkdir("PL")
```

chdir ("назва папки") зміна поточного каталогу

```
os.chdir("folder")
```

os.chdir("..") повернення в попередній каталог

makedirs("назва папки\"назва папки\"назва папки")

```
os.makedirs("PL\PL1\PL1_2")
```

rename() перейменування файлів

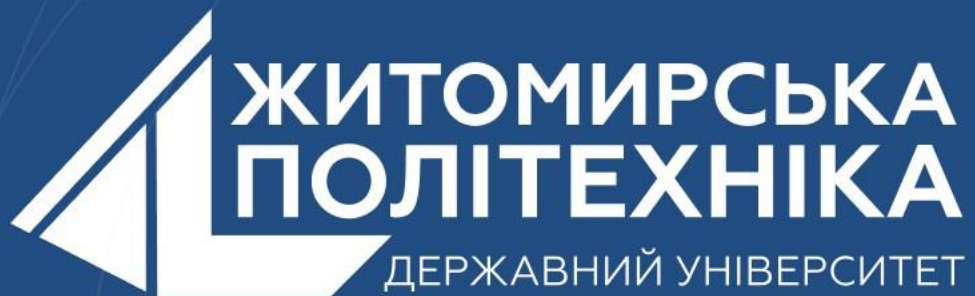
```
os.rename("text.txt", "renamed.txt")
```

replace() переміщення файлів

```
os.replace("renamed.txt", "folder\renamed.txt")
```



   @ZTUEDUUA



- Розвиваємо лідерів
- Створюємо інновації
- Змінюємо світ на краще

