

Лабораторне заняття №6

з навчальної дисципліни

Спеціалізовані мови програмування

Python (advanced)

на тему:

ВВОД ТА ВИВОД

Мета роботи

Ознайомитися з особливостями функцій роботи з вводом та виводом, а також з json в мові програмування Python 3

Хід роботи

- Самостійно на ПК реалізувати програмний код наведений нижче

```
"""Приклад відкриття файлу для читання"""
```

```
def read_file(fname):
```

```
    """ Функція для читання файлу fname  
    і виведення його вмісту на екран  
    """
```

```
    # Відкриття файлу для читання
```

```
    file = open(fname, 'r')
```

```
    # Виведення назви файлу
```

```
    print('File ' + fname + ':')
```

```
    # Зчитання вмісту файлу по рядково
```

```
    for line in file:
```

```
        # Вивод рядка. Переведення рядка в файлі зберігається в рядку, тому
```

```
        # виводимо без додаткового переведення рядка.
```

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

```
    # Закриття файла
```

```
    file.close()
```

```
if __name__ == '__main__':
```

```
    read_file('data/file.txt') # Тестовий файл file.txt знаходиться в каталозі data відносно даного модуля.
```

""" Приклад використання функції os.path.join
для роботи з місцем знаходження файлу """

Модуль, який містить функції для роботи з місцем знаходження в файлової системі
import os.path

```
def read_file(fname):  
    """Функція для читання файлу fname  
    і виведення його вмісту на екран  
    """  
  
    file = open(fname, 'r')  
    print('File ' + fname + ':')  
    for line in file:  
        print(line, end='')  
  
    file.close()  
  
if __name__ == '__main__':  
    # Функція os.path.join з'єднує частини шляху в файловій системі необхідним для даної платформи роздільником  
    read_file(os.path.join('data', 'file.txt'))
```

```
"""Приклад запису даних в текстовий файл"""
```

```
import os.path
```

```
text = "Hello!
```

```
I am a text file. And I had been written with a Python script  
before you opened me, so look up the docs and try to delete  
me using Python, too.
```

```
"""
```

```
def write_text_to_file(filename, text):
```

```
    """Функція для запису в файл filename рядка text"""
```

```
    # Відкриття файлу для запису
```

```
    f = open(filename, "w")
```

```
    # Запис рядка text в файл
```

```
    f.write(text)
```

```
    # Закриття файла
```

```
    f.close()
```

```
if __name__ == '__main__':
```

```
    write_text_to_file(os.path.join('data', 'example02.txt'), text)
```

ПРИКЛАД №4

"""Використання оператора with для автоматичного закриття файлу"""

```
import os.path
```

```
# Побудова імені файлу
```

```
filename = os.path.join('data', 'file.txt')
```

```
# Оператор with автоматично закріє файл при закінченні виконання операторів
```

```
# всередині нього або виникненні виключення
```

```
with open(filename) as file:
```

```
    print(file.read())
```

ПРИКЛАД №5

```
"""Приклад відкриття текстового файлу на читання  
із зазначенням кодування"""
```

```
# __file__ - це атрибут модуля, в якому зберігається ім'я його файлу  
# вихідного коду  
with open(__file__, 'r', encoding='utf-8-sig') as file:  
    for number, line in enumerate(file):  
        print('{0}\t{1}'.format(number + 1, line), end='')  
  
print()
```



```
"""Приклад відкриття файлу на читання та запис"""
```

```
import os.path
import statistics
import datetime
```

```
def calculate_stats(filename):
    with open(filename, 'r+') as file:
        numbers = [float(line) for line in file.readlines()
                    if line != '\n' and not line.lstrip().startswith('#')]
```

```
    sum_ = sum(numbers)
    mean = statistics.mean(numbers)
    median = statistics.median(numbers)
```

```
    cur_time = datetime.datetime.now()
```

```
    fmt = '\n' \
        '# Статистика від {time!s}\n' \
        '# Сума: {sum}\n' \
        '# Медіана: {median}\n' \
        '# Среднє: {mean}'
```

```
    print(fmt.format(time=cur_time,
                      mean=mean,
                      median=median,
                      sum=sum_),
          file=file)
```

```
if __name__ == '__main__':
    filename = os.path.join('data', 'example05.txt') #Тестовий файл example05.txt знаходиться в каталозі data відносно даного модуля.
    calculate_stats(filename)
```

ПРИКЛАД №7

```
"""Приклад відкриття файлу для дозапису"""
```

```
import os.path  
import datetime
```

```
log_file = os.path.join('data', 'example02.txt')  
with open(log_file, 'a') as log:  
    print(datetime.datetime.now(), file=log)
```

ПРИКЛАД №8

"""Приклад перезапису файлу"""

```
import os.path
```

Тестовий файл example07.txt знаходиться в каталозі data відносно даного модуля.

```
filename = os.path.join('data', 'example07.txt')
```

```
# Зчитання даних з файлу
```

```
with open(filename, 'r') as file:
```

```
    lines = file.readlines()
```

```
# Модифікація даних
```

```
lines.insert(2, 'inserted line\n')
```

```
# Перезапис файла
```

```
with open(filename, 'w') as file:
```

```
    file.writelines(lines)
```

```
""" Приклад використання файлового об'єкта io.StringIO """
```

```
import io
```

```
# Створення потоку
```

```
stream = io.StringIO() # или io.StringIO('початкове значення')
```

```
# Запис даних в потік
```

```
stream.write('asdf in memory')
```

```
# Отримання рядка з об'єкта StringIO
```

```
print(stream.getvalue())
```

```
# Виведення поточної позиції
```

```
print('Current position:', stream.tell())
```

```
# Перехід в початок потоку
```

```
stream.seek(0)
```

```
# Запис даних в потік
```

```
stream.write('data')
```

```
# Виведення поточної позиції
```

```
print('Current position:', stream.tell())
```

```
# Читання решти даних в потоці
```

```
print(stream.read())
```

```
# Виведення поточної позиції
```

```
print('Current position:', stream.tell())
```

```
# Отримання рядка з об'єкта StringIO
```

```
print(stream.getvalue())
```

""""Приклад використання json""""

ПРИКЛАД №10

```
import json
import os.path
```

```
data = [
    {
        'name': 'John',
        'age': 20,
    },
    {
        'name': 'Mary',
        'age': 19
    }
]
```

```
filename = os.path.join('data', 'example10.json')
```

Сериалізація

```
with open(filename, 'w') as file:
    json.dump(data, file)
```

Десериалізація

```
with open(filename, 'r') as file:
    read_data = json.load(file)
print(read_data)
```

Завдання на самостійну роботу

Оформити звіт

Заняття закінчено.
Дякую за увагу!