WEB OF SYNERGY:
International Interdisciplinary Research Journal
Volume 2 Issue 3, Year 2023 ISSN: 2835-3013
https://univerpubl.com/index.php/synergy

Options for Working with Files in the Python Programming Language

Zaripov Nozimbek Nayimovich
PhD, Associate Professor of the Exact, Sciences Department in the Bukhara State Pedagogical Institute

Hasanov Behzod Normurot o'g'li
Bukhara State Pedagogical Institute, Student of Stage 2 of the direction, Mathematics and informatics

Article Information

Received: January 26, 2022
Accepted: February 27, 2023
Published: March 28, 2023

Keywords: *program, mode, File, functions, web application, open, component, application, binary, read, append, write, create, text, campus memory, folder, object, Function, information, close, mode, module.*

ABSTRACT

One of the pressing issues is the approach of general education schoolchildren with a separate method in teaching programming languages, the development of simplified methods, the assignment of tasks taking into account the age of students in a time when modern information technologies are widely developed. Learning the possibilities of working with files in the Python programming language will help you understand Python well. Files are used in programming as a set of data used in Python. Allows storage of other objects in programming such as files, data, functions, classes, and modules. This article highlights files in the Python programming language, such as working with them, opening, modifying, editing files, deleting them, and closing files.

INTRODUCTION. Python's syntax is also relatively simple and easy to understand, making it a great language for beginners to learn. It uses indentation to denote code blocks, which forces developers to write clean and readable code. Python is also an open-source language, which means that the source code is freely available and can be modified and distributed by anyone. This has led to a large and active community of developers who contribute to its development, improve its functionality, and create new libraries and tools for it. Overall, Python is a versatile and powerful programming language that is widely used in many industries and fields. Its popularity and ease of use make it an ideal language for beginners and experts alike.

METHODS. Working with files is one of the important parts in Python. Especially when working with web applications. Python has the ability to generate, read, Update and delete files. Opening files is done with the open() function. In this case, this function receives two parameters: filename and mode. Mode refers to the purpose for which the file is opened. These modes are as follows:

r – Read – open the file for reading. If the file is not available, an error will occur.

a – open to add more to the Append-file. Opens a new file if the file is not available.

w – Write – open to write to the file. If the file is not available, a new file will open.

x – Create – generate a new file. An error occurs if such a file exists.

You may encounter various situations when working with a file. For example, access to the file may not be allowed, and h. k. In such cases, an error may occur in a certain line of the program, and the file may not be closed using the close() method in subsequent lines. For this, Python has the with ..as operator, which ensures that the file is closed in any case.

Syntax:

with open(file, mode) as file_name

commands

The open file file_name variable is defined through with and the sequence of commands listed in commands is executed. The file is then automatically closed regardless of what happens.

Python provides various options for working with files in the programming language. Some of the common options include:

Opening a File: The built-in function 'open()' is used to open a file in Python. It takes two arguments: the path to the file and the mode in which it will be opened. The different modes include read ('r'), write ('w'), append ('a'), binary ('b'), and text ('t') modes.

Reading a File: Once a file is opened in Python, we can read the contents of the file using the 'read()' method. This method returns the entire content of the file as a string.

Writing to a File: We can also write data to a file in Python using the 'write()' method. This method writes a string to the file.

Closing a File: It is important to close a file after reading or writing to it. This can be done using the 'close()' method.

Creating a File: We can create a new file in Python using the 'open()' function with the 'w' mode. This will create a new file if it does not already exist.

Appending to a File: If we want to add new data to an existing file, we can use the 'a' mode to open the file and use the 'write()' method to add new data to the file.

Deleting a File: We can delete a file in Python using the OS module. The OS module provides various functions for working with files and directories in Python.

Working with Directories: We can also work with directories in Python using the OS module. The OS module provides functions for creating, deleting, and navigating directories.

Python also has an With operator to define working with files. This operator automatically close the resources that must be used when the file is opened.

The following code section opens, reads and closes the file:

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

data = file.read()

print(data)

DISCUSSION. Working with files in Python is a common task that involves opening, reading, writing, and closing files. Python provides built-in functions and modules to perform these operations. Here is a brief overview of how to work with files in Python:

Opening a File: The first step in working with files is to open them using the open() function. The function takes two arguments - the name of the file to be opened and the mode in which the file is to be opened. For example, to open a file named "example.txt" in read mode, you can use

the following code:

There are many approaches to working with direct file programming, and files can also be used as databases. Therefore, the Python programming language is a very powerful tool for working with data. With the help of other modules, it is possible to analyze data, edit and perform other operations on files. The following are widely established among Python file modules:

OS - Helps to perform operations on files and directories.

SHUTIL - Helps to copy, move and delete files and directories.

GLOB - Helps to search for files in directories.

PATHLIB - Provides many options when working with files and directories.

CSV - helps to work with CSV files.

JSON - Helps to work with JSON files.

PICKLE - Provides the ability to store and load objects on files.

ZIPFILE - helps work on ZIP files.

For example, the **OS.PATH.EXISTS()** function is used to check whether a file or folder exists using the OS module. The following code shows how to check if a file named "example.txt" exists using the **OS** module:

```
import os
if os.path.exists("example.txt"):
print("File exists.")
otherwise:
print("The file does not exist.")
```

This code uses the OS.PATH.EXISTS() function to identify the file. If the file exists, it will output the text "File exists", otherwise it will output the text "File does not exist".

Closing files. It is important to always close files after they have been opened, to avoid potential errors and data corruption. This is done with the close() method.

```
# opening a file
file = open('filename.txt', 'r')
# reading from the file
content = file.read()
# closing the file
file.close()
```

Create a program to write the lines of the poem entered in the file in reverse order to another file. Use the readlines() method in the program.

```
file=open('d:/dars.txt', 'w')
for i in range(4):
k=input()
text=file.write(k+'\n')
file.close()
```

```
file2=open('d:/dars.txt', 'r')
file3=open('d:/dars2.txt', 'w')
text1=file2.readlines()
for i in range(4):
text2=file3.write(str(text1[3-i]))
file2.close()
file3.close()
```

Write a program that determines whether a line in a file contains the @ symbol.

```
file=open('d:/dars.txt', 'w')
for i in range(4):
k=input()
text=file.write(k+'\n')
file.close()
file2=open('d:/dars.txt', 'r')
text2=file2.readlines()
for i in range(4):
for j in range(0,len(text2[i])):
if text2[i][j]=='@':
print(f'{i}-satrda bor')
break
elif len(text2[i])-1==j:
print(f'{i}-satrda yo`q')
file2.close()
```

CONCLUSION. In this article, we explored the different ways to work with files in Python. We learned how to open, read, write, append, and close files, and saw examples of how to use these methods. With these powerful capabilities, developers can automate many file-related tasks and streamline their workflow.

REFERENCES.

1. Zaripov Nozimbek Nayimovich. “Pedagogical problems and solutions in the use of programming environment in the teaching of computer science and information technology” International conference on multidisciplinary research and innovative technologies. – 2021. – T. 1. – C. 95-98.
2. Зарипов Н.Н. Использование иностранного опыта в обучении информатике и информационным технологиям в школе //Проблемы современного образования. – 2020. – №. 6. – С. 213-218.
3. Carol Tice, Neil Tortorella “Freelance Business Bootcamp How to Launch, Earn, and Grow into a Well-Paid Freelancer” January 21, 2015. 29 p.
4. https://www.freelancer.com/

5. И.Ш. Садуллаев, Н.Н. Зарипов “Персональная учебная среда учащегося в режиме дистанционного обучения” - Международна научна школа «Парадигма». Лято, 2015.
6. Zaripov Nozimbek Nayimovich, Hasanov Behzod Normurot o’g’li “Python dasturlash tilini o ‘qitishda funksiyalardan foydalanish metodikasi” Talqin va tadqiqotlar ilmiy-uslubiy jurnali, 2023. - 15-19.
7. Зарипов Н.Н. Творческий подход при уровневой дифференциации учащихся // Инновационные технологии современной научной деятельности: Стратегия, Задачи, Внедрение. – 2019. – С. 27-29.
8. "Python Programming: An Introduction to Computer Science" by John Zelle. – 20-35 p.
9. "Effective Python: 59 Specific Ways to Write Better Python" by Brett Slatkin. – 75-86 p.