

How can “with” be used in Python to open files, and what are some examples of its usage?

Authored by
stats writer

July 2, 2024

RECOMMENDED CITATION

stats writer (2024). *How can “with” be used in Python to open files, and what are some examples of its usage?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=165707>

The "with" keyword in Python is used as a context manager to open and manage files in a more efficient and safe manner. It ensures that the file is automatically closed after the code block is executed, even if an exception occurs. This helps to avoid any potential errors and ensures that system resources are properly managed.

Some examples of using "with" to open files in Python include:

1. Reading a text file:

```
with open("myfile.txt") as file:  
    content = file.read()  
    print(content)
```

2. Writing to a file:

```
with open("myfile.txt", "w") as file:  
    file.write("Hello, World!")
```

3. Appending to a file:

```
with open("myfile.txt", "a") as file:  
    file.write("How are you?")
```

4. Reading and writing to a file:

```
with open("myfile.txt", "r+") as file:  
    content = file.read()  
    file.write("This is a new line.")
```

Overall, using "with" in Python ensures that files are properly opened, closed, and managed, making file handling more efficient and reliable.

Use "with" in Python to Open Files (Including Examples)

You can use the following syntax to open a file in Python, do something with it, and then close the file:

```
file = open('my_data.csv')
```

```
df = file.read()
```

```
print(df)
```

```
file.close()
```

The problem with this approach is that it's very easy to forget to close the file.

A better approach is to use `with open`, which uses the following basic syntax:

```
with open('my_data.csv') as file:
```

```
df = file.read()
```

```
print(df)
```

Using this approach, the file that you're working with is automatically closed so that you don't have to remember to use `file.close()`.

The following examples show how to use `with open` in different scenarios.

Example 1: Use With Statement to Read File

The following code shows how to use the "with" statement to read a file into Python and print the contents of the file:

`withopen('my_data.csv') as file:`

```
df = file.read()
```

```
print(df)
```

```
,points,assists,rebounds
```

```
0,11,5,6
```

```
1,17,7,8
```

```
2,16,7,8
```

```
3,18,9,10
```

```
4,22,12,14
```

```
5,25,9,12
```

```
6,26,9,12
```

```
7,24,4,10
```

```
8,29,8,11
```

The contents of the file are printed and the file is automatically closed without us typing `file.close()`.

Example 2: Use With Statement to Write File

The following code shows how to use the "with" statement to write some text out to a file:

```
withopen('data_out.csv', 'w') as file:
```

```
file.write('Some text to write to CSV file')
```

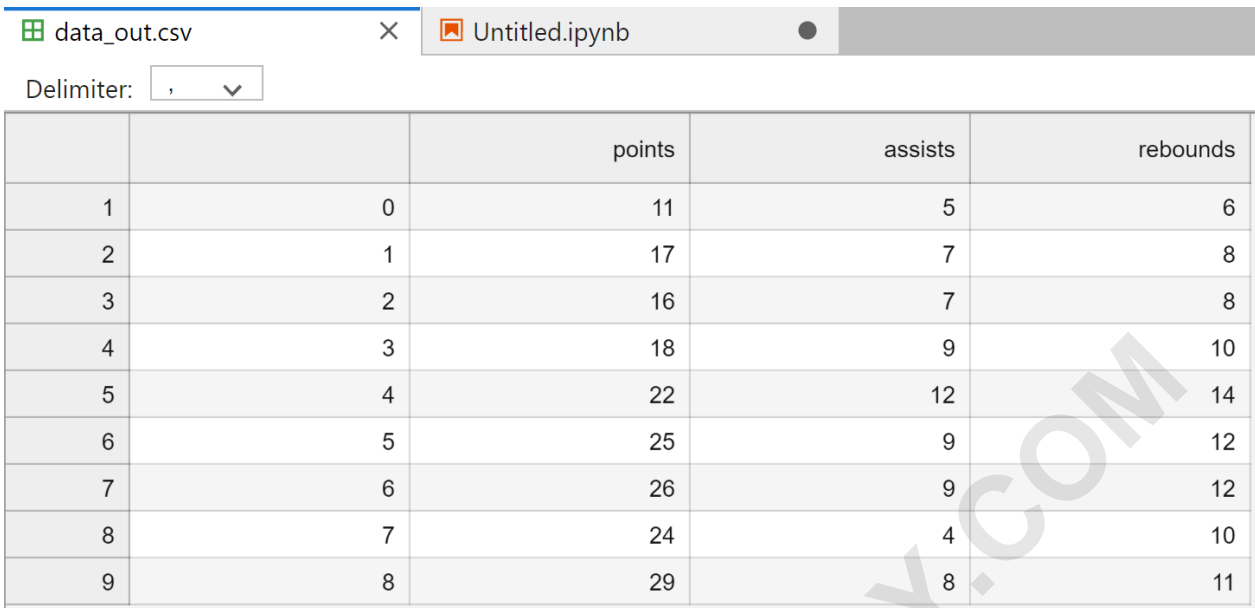
Note that the 'w' within the open() statement tells Python to use 'write' mode with the file as opposed to read mode.

Example 3: Use With Statement to Read & Write Files

We can also open several files at once within a single "with" statement.

```
withopen('my_data.csv', 'r') as infile,  
open('data_out.csv', 'w') as outfile:  
for line in infile:  
outfile.write(line)
```

If we navigate to the location where we wrote 'data_out.csv' then we can view the contents of the file:



		points	assists	rebounds
1	0	11	5	6
2	1	17	7	8
3	2	16	7	8
4	3	18	9	10
5	4	22	12	14
6	5	25	9	12
7	6	26	9	12
8	7	24	4	10
9	8	29	8	11

Note that we can use the `open()` function to open as many files as we'd like within a single "with" statement.

Additional Resources

The following tutorials explain how to perform other common operations in Python: