

In Databricks, you can use Markdown cells to add formatted text, but Markdown itself doesn't support code imports directly. However, you can document the process in Markdown for clarity. Here's how you can explain the steps to import a Python file in Databricks using Markdown:

How to Import a Python File in Databricks

To import a Python file into Databricks, follow one of these methods:

1. Using the Workspace

- Upload the Python file to the **Workspace**:
 1. Go to **Workspace** in the left sidebar.
 2. Navigate to the desired folder.
 3. Click on the dropdown menu and select **Import**.
 4. Upload your `.py` file.

- To import and run the file in a notebook:

```
%run /Workspace/Folder_Name/File_Name
```

2. Using the DBFS (Databricks File System)

- Upload the file to DBFS:

```
dbutils.fs.cp("file:/local/path/to/yourfile.py", "dbfs:/path/to/destination/yourfile.py")
```

- Add the file to your Python path:

```
import sys
sys.path.append('/dbfs/path/to/destination/')
```

- Import the file as a module:

```
import yourfile
```

3. Uploading as a Library

- Package the `.py` file as a module.
 - In Databricks, go to **Workspace > Create > Library** and upload your `.py` or `.whl` file.
 - Attach the library to your cluster.
-

4. Direct Code in Notebooks

- You can also directly paste the code from the Python file into a notebook cell.
- Use `%python` if needed:

```
%python
# Code from your Python file
```

This allows you to work with Python files in different ways in Databricks.

This Markdown text can be used to document the process in a Databricks notebook.