Jupyter Notebooks

python .py files

PYTHON KA

.py Files:

- **Nature:** Plain text files containing Python code.
- **Usage:** Ideal for writing scripts, modules, and packages that can be imported or executed.
- **Execution:** Run from start to finish using the Python interpreter (python script.py).
- **Environment:** Can be executed in any environment where Python is installed.
- **Advantages:**
  - Lightweight and simple.
  - Easy to manage with version control systems.
  - Suitable for building applications and libraries.
- **Limitations:**
  - Lacks built-in support for rich media or inline outputs.
  - Less interactive; not ideal for exploratory analysis.

Jupyter notebooks

PYTHON KA

.ipynb Files:

- **Nature:** JSON formatted files used by Jupyter Notebooks.
- **Usage:** Designed for interactive computing; commonly used in data science for analysis and visualization.
- **Execution:** Run in cells within a Jupyter environment; cells can be executed independently.
- **Environment:** Requires Jupyter Notebook, JupyterLab, or compatible IDEs (like VSCode with extensions).
- **Advantages:**
  - Highly interactive; immediate feedback from code execution.
  - Supports rich media: images, plots, videos, and markdown text.
  - Excellent for demonstrations, tutorials, and exploratory data analysis.
- **Limitations:**
  - Not ideal for production code or building applications.
  - Version control can be cumbersome due to embedded outputs and JSON format.
  -

Jupyter notebooks

PYTHON KA

Feature	.py Files	.ipynb Files
File Extension	.py	.ipynb
Format	Plain text script written in Python	JSON file containing a combination of code cells, markdown text, and outputs
Primary Use Case	Writing Python scripts, modules, and packages for execution as programs	Interactive computing, data analysis, visualization, teaching, and exploratory programming
Execution	Runs as a whole script from start to finish	Runs code interactively in individual cells; cells can be executed out of order
Interactivity	Limited interactivity; requires re-running the script to see changes	High interactivity; supports modifying and running individual cells, viewing outputs immediately
IDE Support	Supported by all Python IDEs and text editors (e.g., VSCode, PyCharm, Sublime)	Requires Jupyter Notebook, JupyterLab, or IDEs with Jupyter support (e.g., VSCode with Python extension)
Content Types	Contains only code (and comments)	Can contain code, formatted text (Markdown), images, equations (LaTeX), and rich media outputs

Jupyter notebooks

PYTHON KA

Feature	.py Files	.ipynb Files
Sharing	Shareable as scripts; others can run the code if they have the necessary setup	Shareable as notebooks; recipients can view code, outputs, and markdown; can be converted to other formats
Export Options	N/A (it's the source code)	Can be exported to formats like HTML, PDF, Markdown, or .py script
Embedding Media	Cannot embed images or rich media directly	Can embed images, videos, interactive plots, and other rich media within the notebook
Learning Curve	Requires knowledge of Python programming and script execution	Beginner-friendly interface; suitable for learning, with immediate feedback and visual outputs
Execution Environment	Runs in the Python interpreter or terminal	Runs in a Jupyter environment (web-based interface)
Dependency Management	External; dependencies managed via requirements files or virtual environments	Can include cells to install dependencies within the notebook (e.g., using !pip install commands)
Collaboration	Code collaboration via traditional methods; merging codebases	Collaborative features limited; however, platforms like JupyterHub enable multi-user environments
Performance	Generally faster execution; suitable for production code	Slight overhead due to interactive environment; not typically used for production deployment
Commenting and Documentation	Comments added using #; documentation via docstrings	Can include extensive markdown documentation alongside code for better explanation and context

When to use what?

- **Use .py files when:**
  - Developing Python applications, modules, or packages.
  - Writing scripts that need to be run from the command line or scheduled tasks.
  - You require better performance and efficiency.
  - Maintaining code with version control systems where clean diffs are necessary.
- **Use .ipynb files when:**
  - Performing data analysis, visualization, or machine learning tasks.
  - Teaching or learning Python and programming concepts.
  - Sharing interactive code with explanations and outputs.
  - Prototyping and experimenting with code snippets.

Jupyter notebooks are better

1. Interactive Computing
2. In line Data Visualization
3. Documentation and Narrative with MarkDown Cells and LaTex support
4. Reproducibility and Sharing
    a. Self contained documents
    b. Export Options
    c. Collaborative plateforms
        i. Google Collab
        ii. Binder
        iii. Everynote
        iv. JupyterHUB
5. Educational Benefits for easy learning
6. Easy to EDA (Exploratory Data Analysis)
7. Support Multiple Languages
8. Integration with Big Data Tools (Apache Spark and Hadoop)
9. Extensibility with Widgets and Extensions (ipywidgets)
10. Presentation Capabilities (using tools like RISE)
11. Much more....