The pyjupyter package

Bright Bara
barabright62@gmail.com

March 7, 2026

1 Introduction

The pyjupyter package provides a lightweight environment for typesetting Python code in LaTeX documents.

It combines the functionality of the listings package for syntax highlighting with the visual capabilities of tcolorbox to produce structured and readable code blocks.

The goal of the package is to provide a simple and robust solution for including Python code in :

- scientific and technical reports
- programming assignments
- lecture notes
- computational and reproducible research documents

The visual style is inspired by Jupyter notebooks while remaining minimal and fully compatible with standard LaTeX workflows.

2 Basic usage

Load the package in the document preamble :

```
\usepackage{pyjupyter}
```

The package defines a single environment called jupyter.

Example

```python
# Example Python code
def square(x):
    return x ** 2

print(square(4))
```

3 Important notes

3.1 About the brackets

The `jupyter` environment must always be invoked with brackets immediately after `\begin{jupyter}` :

```
\begin{jupyter}[]
    ...
\end{jupyter}
```

This applies even when no options are specified.

The reason for this is that Python comments start with the character `#`. In TEX the character `#` is also used internally to denote macro parameters.

If the environment is started without the optional argument brackets and the first line of code begins with a Python comment, TEX may interpret the character incorrectly during argument parsing.

Providing the optional brackets ensures that the environment is fully initialized before the code content is processed by the `listings` engine. This prevents compilation errors when the first line of the code block is a Python comment.

3.2 About highlighting of operators and spaces

The current version of `pyjupyter` relies on the `listings` engine for syntax highlighting. Due to the way this engine parses characters, mathematical and logical operators (such as `+`, `-`, `=`, `*`, etc.) are highlighted in **violet** only when they are surrounded by spaces.

If the code is written in a compact way (e.g., `x=5+2`), the operators will remain in the default text color (black).

- **Correct highlighting** : `x = 5 + 2`
- **No highlighting** : `x=5+2`

While this is a technical limitation of the underlying engine, it also promotes the **PEP 8** style guidelines, which recommend surrounding operators with spaces for better readability.

```
# Operators are highlighted with spaces:
a = 10
b = 5
result = a + b

# Operators remain black without spaces:
result=a+b
```

4 Line numbering

The package provides a convenient option called `numbered` to enable line numbering.

```
'''
Example with line numbers
'''
for i in range(5):
    print(i)
```

Using the `numbered` option is the recommended way to activate line numbers.

It internally configures the underlying `listings` settings to produce consistent numbering.

5 Customization

5.1 Passing options to `tcolorbox`

The `jupyter` environment accepts any option supported by `tcolorbox` :

```
\begin{jupyter}[title=\textbf{Code}, colframe=Navy, colback=blue!2]
    print("Custom background color")
\end{jupyter}
```

Code

```
print("Custom background color")
```

5.2 Passing options to `listings`

Advanced users may still pass options directly to the `listings` engine using the `listing options` key.

```
\begin{jupyter}[listing options={basicstyle=\ttfamily\small}]
    print("Custom listings options")
\end{jupyter}
```

```
print("Custom listings options")
```

6 License

This package is distributed under the *LaTeX Project Public License (LPPL)* version 1.3c.