The *pyluatex* package

Tobias Enderle
https://github.com/tndrle/PyLuaTeX

v0.1.2 (2021/07/14)

Execute Python code on the fly in your LaTeX documents

PyLuaTeX allows you to execute Python code and to include the resulting output in your LaTeX documents in a *single compilation run*. LaTeX documents must be compiled with LuaLaTeX for this to work.

1 Example

1. LaTeX document `example.tex`

```
\documentclass{article}

\usepackage{pyluatex}

\begin{python}
import math
import random

random.seed(0)

greeting = 'Hello PyLuaTeX!'
\end{python}

\newcommand{\randint}[2]{\py{random.randint(#1, #2)}}

\begin{document}
\py{greeting}

$\sqrt{371} = \py{math.sqrt(371)}$

\randint{2}{5}
\end{document}
```

2. Compile using LuaLaTeX (shell escape is required)

```
lualatex -shell-escape example.tex
```

Note: Running LaTeX with the shell escape option enabled allows arbitrary code to be executed. For this reason, it is recommended to compile trusted documents only.

1.1 Further Examples

The folder `example` contains additional example documents:

- `readme-example.tex`
  The example above
- `sessions.tex`
  Demonstrates the use of different Python sessions in a document
- `data-visualization.tex`
  Demonstrates the visualization of data using *pgfplots* and *pandas*

For more intricate use cases have a look at our tests in the folder `test`.

2 Installation

PyLuaTeX is not yet available through package managers or CTAN[1].

To install PyLuaTeX, do the following steps:

1. Locate your local *TEXMF* folder
   The location of this folder may vary. Typical defaults for TeX Live are `~/texmf` for Linux, `~/Library/texmf` for macOS, and `C:\Users\<user name>\texmf` for Windows. If you are lucky, the command `kpsewhich -var-value=TEXMFHOME` tells you the location. For MiKTeX, the folder can be found and configured in the *MiKTeX Console*.
2. Download the latest release[2] of PyLuaTeX
3. Put the downloaded files in the folder `TEXMF/tex/latex/pyluatex` (where `TEXMF` is the folder located in 1.)
   The final folder structure must be

```
TEXMF/tex/latex/pyluatex/
|-- pyluatex-interpreter.py
|-- pyluatex-json.lua
|-- pyluatex.lua
|-- pyluatex.sty
|-- ...
```

3 Reference

PyLuaTeX offers a simple set of options, macros and environments.

[1] `https://ctan.org`
[2] `https://github.com/tndrle/PyLuaTeX/releases/latest`

3.1 Package Options

- `verbose`
  If this option is enabled, Python input and output is written to the log file.
  Example: `\usepackage[verbose]{pyluatex}`
- `executable`
  Specifies the path to the Python executable. (default: `python3`)
  Example: `\usepackage[executable=/usr/local/bin/python3]{pyluatex}`

3.2 Macros

- `\py{code}`
  Executes `code` and writes the output to the document.
  Example: `\py{3 + 7}`
- `\pyc{code}`
  Executes `code`
  Example: `\pyc{x = 5}`
- `\pyfile{path}`
  Executes the Python file specified by `path`.
  Example: `\pyfile{main.py}`
- `\pysession{session}`
  Selects `session` as Python session for subsequent Python code.
  The session that is active at the beginning is `default`.
  Example: `\pysession{main}`

3.3 Environments

- `python`
  Executes the provided block of Python code.
  The environment handles characters like `_`, `#`, `%`, `\`, etc.
  Code on the same line as `\begin{python}` is ignored, i.e., code must start on the next line.
  If leading spaces are present they are gobbled automatically up to the first level of indentation.
  Example:

```
\begin{python}
    x = 'Hello PyLuaTeX'
    print(x)
\end{python}
```

4 Requirements

- LuaLaTeX
- Python 3
- Linux, macOS or Windows

Our automated tests currently use TeX Live 2021 and Python 3.7+ on Ubuntu 20.04, macOS Catalina 10.15 and Windows Server 2019.

5 How It Works

PyLuaTeX runs a Python `InteractiveInterpreter`[3] (actually several if you use different sessions) in the background for on the fly code execution. Python code from your LaTeX file is sent to the background interpreter through a TCP socket. This approach allows your Python code to be executed and the output to be integrated in your LaTeX file in a single compilation run. No additional processing steps are needed. No intermediate files have to be written. No placeholders have to be inserted.

6 License

LPPL 1.3c[4] for LaTeX code and MIT license[5] for Python and Lua code.

We use the great json.lua[6] library under the terms of the MIT license[7].

[3] https://docs.python.org/3/library/code.html#code.InteractiveInterpreter
[4] http://www.latex-project.org/lppl.txt
[5] https://opensource.org/licenses/MIT
[6] https://github.com/rxi/json.lua
[7] https://opensource.org/licenses/MIT