

# PythonTeX Gallery

Geoffrey M. Poore

July 19, 2012

## Abstract

PythonTeX allows you to run Python code from within L<sup>A</sup>T<sub>E</sub>X documents and automatically include the output. This document serves as an example of what is possible with PythonTeX.\*

## 1 General Python interaction

We can typeset code that is passed to Python, and pull in the results.

This can be simple. For example, `??` returns the following:

```
?? PythonTeX ??
```

Or we could access the printed content verbatim (it might contain special characters):

```
?? PythonTeX ??
```

Python interaction can also be more complex. `??` returns `?? PythonTeX ??`. In this case, the printed results include L<sup>A</sup>T<sub>E</sub>X code, which is correctly interpreted by L<sup>A</sup>T<sub>E</sub>X to ensure that there is not an extra space after the 16. Printed output is saved to a file and brought back in via `\input`, and the `\endinput` command prevents L<sup>A</sup>T<sub>E</sub>X from treating the newline at the end of the file as justification for a space character.

But we don't have to typeset the code. It can be hidden. And then we can access it later: `??`.

## 2 Python console environment

PythonTeX includes an environment that emulates a Python interactive session. Commands are entered within the environment, each line is treated as input to an interactive session, and the result is typeset.

```
?? PythonTeX ??
```

---

\*Since PythonTeX runs Python code (and potentially other code) on your computer, documents using PythonTeX have a greater potential for security risks than do standard L<sup>A</sup>T<sub>E</sub>X documents. You should only compile PythonTeX documents from sources you trust.

### 3 Basic SymPy interaction

PythonTeX allows us to perform algebraic manipulations with SymPy and then properly typeset the results.

We create three variables, and define  $z$  in terms of the other two.

**?? PythonTeX ??**

Now we can access what  $z$  is equal to:

$$z = ??$$

Many things are possible, including some very nice calculus.

**?? PythonTeX ??**

$$?? = ??$$

It's easy to use arbitrary symbols in equations.

**?? PythonTeX ??**

$$?? = ??$$

### 4 Plots with matplotlib

We can create plots with matplotlib, perfectly matching the plot fonts with the document fonts. No more searching for the code that created a figure!

**?? PythonTeX ??**

### 5 Basic pylab interaction

**?? PythonTeX ??**

$$\int_0^\infty e^{-x^2} dx = ??$$

### 6 An automated derivative and integral table

PythonTeX allows some amazing document automation, such as this derivative and integral table. Try typing that by hand, fast!

**?? PythonTeX ??**

**?? PythonTeX ??**

## 7 Step-by-step solutions

Using SymPy, it is possible to typeset step-by-step solutions. In this particular case, we also use the `mdframed` package to place a colored background behind our code.

**Step-by-Step**

**Integral**

**Evaluation**

**?? PythonTeX ??**

**?? PythonTeX ??**

## 8 Including stderr

PythonTeX allows code to be typeset next to the stderr it produces. This requires the package option `stderr`

**?? PythonTeX ??**

This code causes a syntax error:

**?? PythonTeX ??**

The file name that appears in the message can be customized using the package option `stderrfilename`.