

Pyxam Cheat Sheet

Running Pyxam

Usage `pyxam.py [Options] template`

Command list:

<code>out</code>	<code>-o</code>	[file]	Set the output directory
<code>temp</code>	<code>-tmp</code>	[file]	Set the temporary directory
<code>figure</code>	<code>-fig</code>	[file]	Set the figure directory
<code>number</code>	<code>-n</code>	[int]	Set the number of exams to generate
<code>sample</code>	<code>-smp</code>	[int]	Set the default sample number
<code>recompilation</code>	<code>-r</code>	[int]	Set the number of recompilations
<code>title</code>	<code>-t</code>	[title]	Set the exam name
<code>format</code>	<code>-f</code>	[format]	Set the export format (tex, pdf, dvi, html)
<code>shell</code>	<code>-shl</code>	[shell]	Set the shell (python, matlab, octave, julia)
<code>method</code>	<code>-m</code>	[method]	Set the selection method (random, sequence)
<code>population</code>	<code>-p</code>	[file]	Set the class list for mixing
<code>solutions</code>	<code>-s</code>		Enable solutions
<code>alphabetize</code>	<code>-a</code>		Enable lettered versioning rather than numbered
<code>clean</code>	<code>-c</code>		Enable LaTeX cleanup
<code>interactive</code>	<code>-i</code>		Enable interactive pdflatex
<code>logging</code>	<code>-l</code>		Enable logging
<code>debug</code>	<code>-d</code>		Disable file cleanup (keep temporary files)

For more details see `README.md`

LaTeX Commands

Python:

<code><<>>= ... @</code>	Run Python code and print verbatim
<code><<echo=False>>= ... @</code>	Run Python code silently
<code><%...%></code>	Run Python code snippet and print the result
<code><%=...%></code>	Run Python code snippet silently
<code>\Pexpr{...}</code>	Run Python code snippet and print the result

Importing Questions:

<code>\Pimport{file}</code>	Import a single question file
<code>\Pimport{file1 file2}</code>	Import either file1 or file2
<code>\Pimport[n]{file}</code>	Import a single question file n times
<code>\Pimport{dir}</code>	Import a single question from directory dir
<code>\Pimport[n]{dir1 dir2 dir3}</code>	Import n questions from dir1, dir2, or dir3

Constants:

<code>\Pconst{VERSION}</code>	Get the exam version number or letter
<code>\Pconst{TITLE}</code>	Get the exam title
<code>\Pconst{STUDENT}</code>	Get a student's name
<code>\Pconst{STUDNUM}</code>	Get a student's number

Options:

<code>\Parg{args}</code>	Equivalent to running <code>pyxam.py [args] template</code>
--------------------------	---

Examples

See `examples/template.tex` for a simple exam that implements all of Pyxam's features

See `examples/exam.tex` for examples of more complex problems

See `examples/github.tex` for an introductory guide to github

See `README.md` for a general overview of the tools and basic usage

Development Tools

Github

The version control system used for Pyxam. Github allows for easy management and access of source code. Github can be found at <https://github.com/andtheprojectpage> for Pyxam can be found at <https://github.com/balancededge/pyxam>.

Git-Cola

A GUI client for github on Unix systems. A convenient tool when working with a larger number of files in sub directories where the command line may be less suitable. Git-Cola can be installed through Yast.

PyCharm

A Python IDE with all the bells and whistles. PyCharm makes programming Python easy and enjoyable whilst also still being one of the most responsive editors available. PyCharm Community edition is free and can be found at <https://www.jetbrains.com/pycharm/>.

Dillinger.io

A Markdown browser based editor. Dillinger is simple and elegant a great solution for writing Markdown documents. Dillinger can be found at <http://dillinger.io/>.

Emacs

A powerful and configurable text editor. An ideal environment when working in a large variety of programming languages and with a large number of file formats. Emacs can be installed through Yast.

Emacs Shortlist

<code>M-p</code>	Previous shell command
<code>C-x-C-v RET</code>	Refresh the currently selected buffer
<code>C-x-1</code>	Close all windows except the currently selected one
<code>C-x-2</code>	Split window vertically
<code>C-x-3</code>	Split window horizontally
<code>C-x-0</code>	Close the currently selected window
<code>C-x-k RET</code>	Kill the currently selected buffer