Python Debugger

Getting Started

`import pdb;pdb.set_trace()` `breakpoint() (New in Python 3.7)`	Start pdb inside a python script
`python -m pdb <file.py>`	Start pdb from the command line

Stepping

`n(ext)`	Step over
`s(tep)`	Step into
`r(eturn)`	Continue until the current function returns
`c(ontinue)`	Continue until the next breakpoint is encountered
`unt(il) line_number`	Continue until a specific line is encountered. Or continue until a larger line number is reached if no line_number is set
`u(p)`	Up one level in the stack trace
`d(own)`	Down one level in the stack trace
`h(elp)` `h(elp) command`	Show help
`q(uit)`	Quit debugger

Breakpoints

`b(reak)`	Show all breakpoints
`b(reak) line_number`	Set a breakpoint at a specific line
`b(reak) line_number, condition`	Set a breakpoint at a specific line, if condition is met
`b(reak) file:line_number`	Set a breakpoint in a file at a specific line
`b(reak) func`	Set a breakpoint at the first line of a function
`disable number`	Disable breakpoint number
`enable number`	Enable breakpoint number
`clear number`	Remove breakpoint number

Printing

`p(rint) expr` `pp expr`	Print the value of expr
`w(here)`	Print current position and stack trace
`l(ist)` `l(ist) start, end`	Print 11 lines of code around the current line
`a(rgs)`	Print the arguments of the current function