

Invoking the debugger

<code>import pdb; pdb.set_trace()</code>	Within the code
<code>python -m pdb file.py</code>	From the command line

Basic commands

<code>p</code>	print variable value
<code>pp</code>	pretty print
<code>a</code>	print input args of function
<code>locals</code>	show local variables
<code>n</code>	next
<code>s</code>	step inside a function
<code>c</code>	continue
<code>r</code>	run until func returns
<code><Enter></code>	repeat previous command
<code>l</code>	print nearby code
<code>l from,to</code>	print code between lines
<code>w</code>	where am I + traceback
<code>u</code>	up the call stack
<code>d</code>	down the call stack
<code>q</code>	quit
<code>h</code>	help
<code>h command</code>	info about a command

Breakpoints

<code>b 45</code>	set on line 45
<code>b 45, x == 'abc'</code>	set with condition
<code>b</code>	list current breakpoints
<code>b funcName</code>	set breakpoint in function
<code>b file.py:41</code>	set on line 41 of file.py
<code>cl</code>	clear all breakpoints
<code>cl 42</code>	clear breakpoint #42
<code>tbreak ...</code>	hit only once, syntax as <code>`b`</code>
<code>disable 42</code>	turn off breakpoint #42
<code>enable 42</code>	enable breakpoint #42
<code>ignore 42 [N]</code>	ignore bp#42 N times
<code>condition 42 x=='abc'</code>	set condition to bp (no condition = remove it)



By **ralienpp**

cheatography.com/ralienpp/

Published 18th June, 2014.

Last updated 18th June, 2014.

Page 1 of 1.

Sponsored by **Readability-Score.com**

Measure your website readability!

<https://readability-score.com>