

pdb cheatsheet

Invoking the debugger

Enter at the start of a program, from the command line:

```
python -m pdb mycode.py
```

Enter in a statement or function:

```
import pdb
# your code here
if __name__ == '__main__':
    # start debugger at the beginning of a function
    pdb.runcall(function[, argument, ...])
    # execute an expression (string) under the debugger
    pdb.run(expression)
```

Enter at a specific point in the code:

```
import pdb
# some code here
# the debugger starts here
pdb.set_trace()
# rest of the code
```

In ipython:

```
%pdb          enter the debugger automatically after an exception is raised
%debug        enter the debugger post-mortem where the exception was thrown
```

Debugger commands

| | |
|--|--|
| h (help) [<i>command</i>] | print help about <i>command</i> |
| n (next) | execute current line of code, go to next line |
| c (continue) | continue executing the program until next breakpoint, exception, or end of the program |
| s (step into) | execute current line of code; if a function is called, follow execution inside the function |
| l (list) | print code around the current line |
| w (where) | show a trace of the function call that led to the current line |
| p (print) | print the value of a variable |
| q (quit) | leave the debugger |
| b (break) [<i>lineno</i> <i>function</i> [, <i>condition</i>]] | set a breakpoint at a given line number or function, stop execution there if <i>condition</i> is fulfilled |
| cl (clear) | clear a breakpoint |
| ! (execute) | execute a python command |
| <enter> | repeat last command |