

pdb- the python debugger

- Using print statements to debug a script is a useful approach, but for larger scripts and more complex bugs a debugger is very useful to step through code and examine variables when an exception is thrown etc.
- The standard python debugger is pdb
- <https://docs.python.org/3.7/library/pdb.html#debugger-commands>
- This is a rudimentary command-line debugger but has the advantage of being built-in to python and so always available.
- (Your editor may have more convenient graphical interface to the debugger)

pdb- the python debugger

The simplest approach is to add

breakpoint()

to your script at points where you need to examine the code and live variables at that point.

(see test.py and test_debug.py code examples)

(Note: This requires python version 3.7 or higher)

pdb- the python debugger

Once breakpoint() is reached, the code enters the debugger, where commands can be given to inspect state etc. These include:

p to print a variable

l to list surrounding code

n to step to the next line

s to step into the next line

c to continue

u or d to step up and down stack frames

b to set a breakpoint

pdb- the python debugger

We can also do post-mortem debugging (jump to the site just before an exception is thrown) without modifying the code by calling python with:

```
python -m pdb test.py
```

There is much more functionality available in the debugger than we cover here, such as breakpoints. See the documentation for more details.