In [4]:

```
#Debugger helps the developer to keep track of thye code segments while it's executing

import pdb

def sequencePrinter(n):
    for item in range (n):
        pdb.set_trace() #set the tracker
        print(item)
    return

sequencePrinter(3)

#debugger options: 1) c: continue, 2) q: quit debugging, 3) p: print, 4) locals(): prin
t all local variables, 5) list: get the current execution portion, 6) h: get all the fu
nctions
```

```
> <ipython-input-4-77ed4e4ab2e1>(8)sequencePrinter()
-> print(item)
(Pdb) p locals()
{'item': 0, 'n': 3}
(Pdb) c
0
> <ipython-input-4-77ed4e4ab2e1>(7)sequencePrinter()
-> pdb.set_trace() #set the tracker
(Pdb) p locals()
{'item': 1, 'n': 3}
(Pdb) c
1
> <ipython-input-4-77ed4e4ab2e1>(8)sequencePrinter()
-> print(item)
(Pdb) p locals()
{'item': 2, 'n': 3}
(Pdb) c
2
```