Python >3.7 PDB Cheat-sheet

START	* - Indicates most commonly used	INTERACTING W/ PROGRAM	
breakpoint()	*Start pdb within script*	**b**(reak) **#**	*Show all breakpoints with #*
python -m pdb <file.py>	*Start pdb and script via CLI*	**b**(reak) <function>	*Set breakpoint at <function>*
		unt(il) **#** *	*Continue unitl line # then breakpoint*
h(elp)	***Many commands use both 1st letter or whole word***	**c**(lear) **#**	*Clears breakpoint at line #*
h <command>	*Help on command*	**ignore #**	*Ignore breakpoint at # -* ***Good for Loops***
q(uit) *	*Quit pdb*	**run / restart**	*Run/restart debugged python script*
		whatis <arg>	*Display type of argument*
MOVEMENT		**alias** [name [command [params …]]] - *Very Useful* Create an alias called 'name' that executes 'command'	
<ENTER>	*Repeat last command*		
n(ext) *	*Execute current line*		
s(tep) *	Step into function	**PDB workflow: using PDB is an iterative process.** 1. Open text editor & CLI terminal 2. Insert 'breakpoint()' at problem code/area. 2. Run script from CLI: it will halt at 'breakpoint()' & it returns Pdb interpreter 3. Investigate error, traceback, or code 4. Query variables or lines of code using 'p' or 'l' 5. Change code in text editor 6. Repeat	
c(ontinue) *	*Execute until next breakpoint*		
u(p)	*Move up one level in stack*		
d(own)	*Move down one level in stack*		
j(ump) **#**	*Jump to & execute line #*		
r(eturn)	*Execute until function 'return'*		
EXAMINE			
!<statement> *	*Execute python statement from PDB*	**NOTE:** 1. PDB will not help find/correct logical errors! 2. PDB is able to determine if variable is used first time, for example. 3. It is possible to run PDB without leaving the PDB interpreter too often using **alias, break, until and clear.**	
p(rint) var *	*Print value of expression*		
pp var	***Pretty print*** *(good for JSON, etc.)*		
l(ist) *	*List 11 lines from current breakpoint*		
l(ist) <1st #> <2nd #>	*Display between lines #1-#2*	Author Matt Curcio, ver. 1, Mar 2023, C.C.	
ll	***Long list*** *display ~20 lines*		