

## Quick Reference

Command parameters are in *italics*. Optional parameters are in square brackets. All commands can be abbreviated until they become ambiguous.

*This list is very very incomplete, and only shows things talked about in this tutorial!*

### Help Commands

<b>help <i>command</i></b>	Get help on a certain command
<b>apropos <i>keyword</i></b>	Search help for a particular keyword

### Starting and Quitting

<b>gdb [-tui] [-c <i>core</i>] [<i>exename</i>]</b>	<b>(Unix Command)</b> Start <b>gdb</b> on an executable or standalone; specify "-tui" to start the TUI GUI; specify "-c" with a corefile name to see where a crash occurred
<b>run [<i>arg1</i>] [<i>arg2</i>] [...]</b>	Run the currently loaded program with the given command line arguments
<b>quit</b>	Exit the debugger
<b>file <i>exename</i></b>	Load an executable file by name

### Breakpoints and Watchpoints

<b>break <i>location</i></b>	Set a breakpoint at a location, line number, or file (e.g. "main", "5", or "hello.c:23")
<b>watch <i>expression</i></b>	Break when a variable is written to
<b>rwatch <i>expression</i></b>	Break when a variable is read from
<b>awatch <i>expression</i></b>	Break when a variable is written to or read from
<b>info break</b>	Display breakpoint and watchpoint information and numbers
<b>info watch</b>	Same as <b>info break</b>
<b>clear <i>location</i></b>	Clear a breakpoint from a location
<b>delete <i>num</i></b>	Delete a breakpoint or watchpoint by number

### Stepping and Running

<b>next</b>	Run to the next line of this function
<b>step</b>	Step into the function on this line, if possible
<b>stepi</b>	Step a single assembly instruction
<b>continue</b>	Keep running from here
<b>CTRL-C</b>	Stop running, wherever you are
<b>finish</b>	Run until the end of the current function
<b>advance <i>location</i></b>	Advance to a location, line number, or file (e.g. "somefunction", "5", or "hello.c:23")
<b>jump <i>location</i></b>	Just like <b>continue</b> , except jump to a particular location first.

### Examining and Modifying Variables

<b>display <i>expression</i></b>	Display the value of a variable or expression every step of the program—the expression must make sense in the current scope
<b>info display</b>	Show a list of expressions currently being displayed and their numbers
<b>undisplay <i>num</i></b>	Stop showing an expression identified by its number (see <b>info display</b> )
<b>print <i>expression</i></b>	Print the value of a variable or expression
<b>printf <i>formatstr expressionlist</i></b>	Do some formatted output with <code>printf()</code> e.g. <code>printf "i = %d, p = %s\n", i, p</code>
<b>set <i>variable expression</i></b>	Set a variable to value, e.g. <code>set variable x=20</code>
<b>set (<i>expression</i>)</b>	Works like <b>set variable</b>

### Window Commands

<b>info win</b>	Shows current window info
<b>focus <i>winname</i></b>	Set focus to a particular window bby name ("SRC", "CMD", "ASM", or "REG") or by position ("next" or "prev")
<b>fs</b>	Alias for <b>focus</b>
<b>layout <i>type</i></b>	Set the window layout ("src", "asm", "split", or "reg")
<b>tui reg <i>type</i></b>	Set the register window layout ("general", "float", "system", or "next")
<b>winheight <i>val</i></b>	Set the window height (either an absolute value, or a relative value prefaced with "+" or "-")
<b>wh</b>	Alias for <b>winheight</b>
<b>set disassembly-flavor <i>flavor</i></b>	Set the look-and-feel of the disassembly. On Intel machines, valid flavors are <b>intel</b> and <b>att</b>

### Misc Commands

<b>RETURN</b>	Hit RETURN to repeat the last command
<b>backtrace</b>	Show the current stack
<b>bt</b>	Alias for <b>backtrace</b>
<b>attach <i>pid</i></b>	Attach to an already-running process by its PID
<b>info registers</b>	Dump integer registers to screen
<b>info all-registers</b>	Dump all registers to screen