GDB Cheat Sheet

With an emphasis on debugging MIPS assembly code
Jeffrey Knockel <jeff250 at unm dot edu>

r[un] [arg0 arg1 … argn]	Run program
b[reak] *address*	Add a breakpoint at address. Address can be a symbol or a file:line pair, e.g. `b yourname-lab3.S:30`
d[elete] b[reakpoints] *number*	Delete a breakpoint by number. (Use i[nfo] b[reaks] to get this number.)
c[ontinue]	Continue running program after stopped
s[tep]	Step to next line, including into function calls
n[ext]	Like step, except does not step into function calls
s[tep]i	Step exactly one instruction
n[ext]i	Like stepi, except does not step into function calls
p[rint] [/*format*] *expr*	Print expression, e.g. `p /d strlen($a0)`
printf "*format string*", *arg1*, *arg2*, …, *argn*	Use printf syntax to print variables, e.g. `printf "$t0 = %d, $t1 = %d\n", $t0, t1`
x[/*count*][*format*]	Examine memory address, e.g., to read and disassemble 6 instructions: `x/6i $t9`
cal[l] *expr*	Evaluates expression, e.g. `call foo($t0, $t1)`
i[nfo] b[reaks]	Print out current breakpoints
i[nfo] r[egisters]	Print out all registers
b[ack]t[race]	Print out backtrace
s[et] *var* = *expr*	Set a variable equal to an expression, e.g. `s $t0 = 0`
q[uit]	Quit gdb