CHIP-8 Instruction Set

http://octo-ide.com

N is a number between 0 and 15.
NN is a number between 0 and 255.
NNN is an address between 0 and 4095.
vx and **vy** are registers (0-F).
i is the memory index register.
Instructions in gray rows may modify the vF register.

Machinecode	Octo Instruction	Comments
00E0	`clear`	
00EE	`return`	Exit a subroutine
1NNN	`jump NNN`	
2NNN	`NNN`	Call a subroutine
3XNN	`if vx != NN then`	
4XNN	`if vx == NN then`	
5XY0	`if vx != vy then`	
6XNN	`vx := NN`	
7XNN	`vx += NN`	
8XY0	`vx := vy`	
8XY1	`vx |= vy`	Bitwise OR
8XY2	`vx &= vy`	Bitwise AND
8XY3	`vx ^= vy`	Bitwise XOR
8XY4	`vx += vy`	vf = 1 on carry
8XY5	`vx -= vy`	vf = 0 on borrow
8XY6	`vx >>= vy`	vf = old least significant bit
8XY7	`vx =- vy`	vf = 0 on borrow
8XYE	`vx <<= vy`	vf = old most significant bit
9XY0	`if vx == vy then`	
ANNN	`i := NNN`	
BNNN	`jump0 NNN`	Jump to address NNN + v0
CXNN	`vx := random NN`	Random number 0-255 AND NN
DXYN	`sprite vx vy N`	vf = 1 on collision
EX9E	`if vx -key then`	Is a key not pressed?
EXA1	`if vx key then`	Is a key pressed?
FX07	`vx := delay`	
FX0A	`vx := key`	Wait for a keypress
FX15	`delay := vx`	
FX18	`buzzer := vx`	
FX1E	`i += vx`	
FX29	`i := hex vx`	Set i to a hex character
FX33	`bcd vx`	Decode vx into binary-coded decimal
FX55	`save vx`	Save v0-vx to i through (i+x)
FX65	`load vx`	Load v0-vx from i through (i+x)