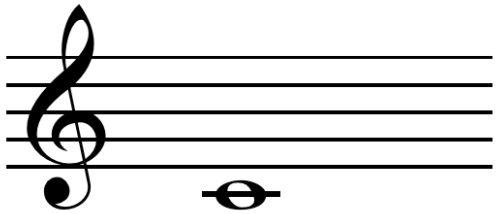
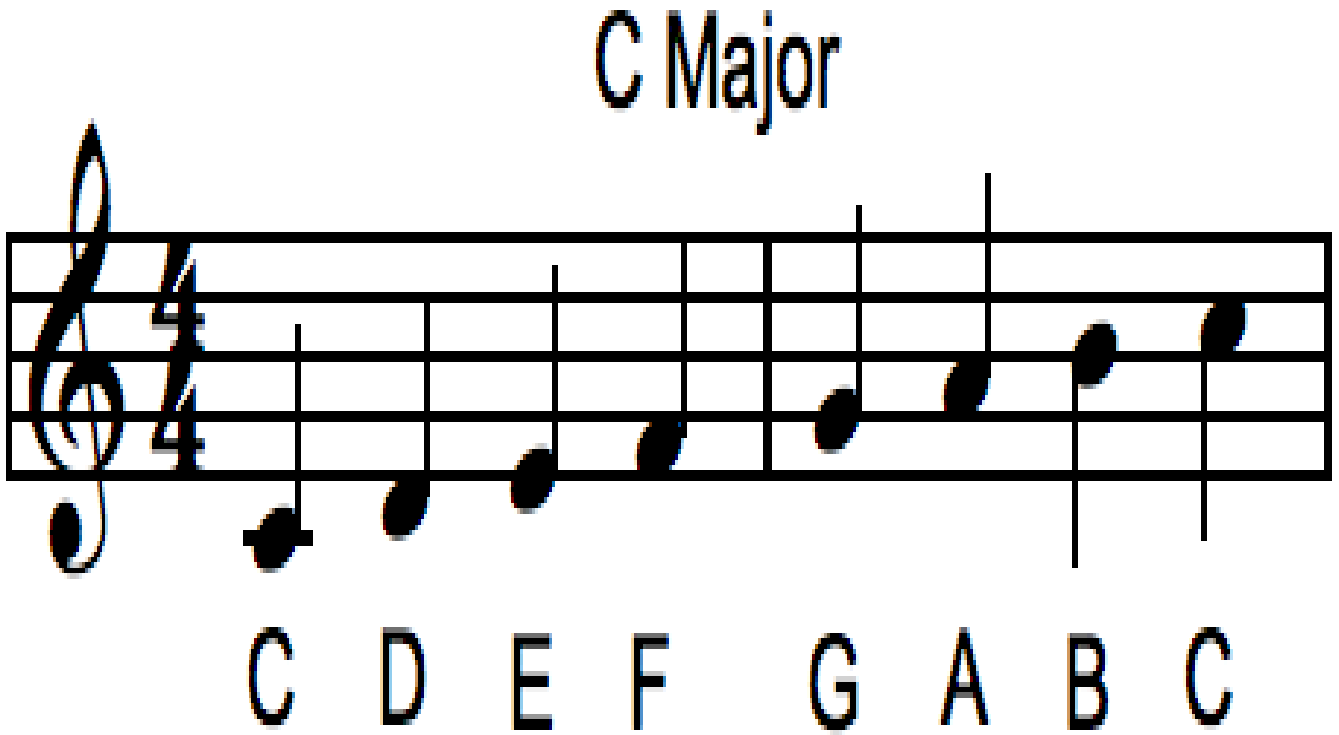
Music Workbench

Name of Notes (low to high)

Note scale of computer notes, by name low ⟵ ⟶ high							
note_B0	note_B1	note_B2	note_B3	note_B4	note_B5	note_B6	note_B7
note_C1	note_C2	**note_C3**	note_C4	note_C5	note_C6	note_C7	note_C8
note_CS1	note_CS2	note_CS3	note_CS4	note_CS5	note_CS6	note_CS7	note_CS8
note_D1	note_D2	note_D3	note_D4	note_D5	note_D6	note_D7	note_D8
note_DS1	note_DS2	note_DS3	note_DS4	note_DS5	note_DS6	note_DS7	note_DS8
note_E1	note_E2	note_E3	note_E4	note_E5	note_E6	note_E7	
note_F1	note_F2	note_F3	note_F4	note_F5	note_F6	note_F7	
note_FS1	note_FS2	note_FS3	note_FS4	note_FS5	note_FS6	note_FS7	
note_G1	note_G2	note_G3	note_G4	note_G5	note_G6	note_G7	
note_GS1	note_GS2	note_GS3	note_GS4	note_GS5	note_GS6	note_GS7	
note_A1	note_A2	note_A3	note_A4	note_A5	note_A6	note_A7	
note_AS1	note_AS2	note_AS3	note_AS4	note_AS5	note_AS6	note_AS7	

"**note_C3**" is middle C, That is:



C Major musical scale	Computer note name
	etc.↑ - note_C4 - note_B3 - note_A3 - note_G3 - note_F3 - note_E3 - note_D3 - **note_C3** (**middle C**) etc. ↓

Computer Note & Rest Names, Values, Beats & Symbols

Note duration name	Note/Rest duration time computer name	Note/Rest values	Note/Rest value	Number of beats	Note symbol	Rest symbol
Semi-brieve	`semib`	Whole note/rest	1	4 beats		
Dotted minim	`dot_minim`	Three quarter note/rest	¾	3 beats		
Minim	`minim`	Half note/rest	½	2 beats		
Crotchet	`crot`	Quarter note/rest	¼	1 beat		
Quaver	`quav`	Eight note/rest	1/8	1/2 beat		
Semiquaver	`semiq`	Sixteenth note/rest	1/16	1/4 beat		

Durations may be compounded and/or arithmetically adjusted in all music commands (`play` and `rest`). For example,

```
play(note_AS5, crot + quav); // 1.5 beats
play(note_C6, quav + semiq); // 0.75 beats
play(note_B2, semib * 1.5);  // 6 beats
play(note_FS4, crot + semiq);// 1.25 beats
play(note_DS3, quav + quav); // 1 beat
play(note_D6, crot / 3);     // 1/3 beat
play(note_DS4, dot_minim + minim);  // 5 beats
rest(quav + semiq);          // 0.75 beats
rest(crot + semiq);          // 1.25 beats
rest(semib * 1.75);          // 7 beats
rest(semiq / 2);             // 1/8 beats, demisemiquaver
etc.
```

In-built computer tempos

Computer tempo names	Number beats per minute	Other computer names	Comments

grave	40		The time interval for a crotch is automatically calculated as:
			60/tempo.
			
			Other notes are then calculated based on the crotchet value.
			
			Use `set_tempo` to change tempo value for notes and rests.
			
			Note that any tempo value may be set, e.g.
			
			`set_tempo(default_tempo * 1.5);`
			`set_tempo(190);`
			`set_tempo(adagio * 2);`
			`set_tempo(95);`
			`etc.`

largo	46		
lento	52		
adagio	56		
larghetto	60		
adagietto	66		
andante	72		
andantino	80		
maestroso	88		
moderato	100		
allegretto	104		
animato	120	default_tempo	
allegro	132		
allegro_assai	144		
vivace	160		
presto	184		
prestissimo	208		

Music commands

Command	Parameters	Description	Examples
play	note name, note duration time	Plays the given note for the given duration. Time duration can be given explicitly as seconds/fraction of seconds or as a Computer Note & Rest Name Values.	`play(note_C4, minim);` `play(note_FS3, 3.5);`
play	note name, note duration time, light number	As above, but an additional parameter can be given which is the light number to be turned on when the note is playing. The light will be turned off when the note completes.	`play(note_DS3, semib, light3);` `play(note_C5, minim, 8);`

rest	rest duration time	Rests for the given duration (period of silence).
		Time duration can be given explicitly as seconds/fraction of seconds or as a computer note/rest name value.
			`rest(crot);`
			`rest(minim+quav);`
			`rest(4);`

set_tempo	tempo time	Sets the tempo for playing notes/rests. The '`default_tempo`' is 120 beats per minute (animato).
		The currently set tempo may be queried at any time by reference to the variable '`current_tempo`'.
			`set_tempo(default_tempo);`
			`set_tempo(default_tempo * 1.5);`
			`set_tempo(allegro);`

Blank Manuscript Paper (print copies as required)

Lights Workbench

Lights numbers

Light number	Computer light name
1	`light1`
2	`light2`
3	`light3`
4	`light4`
5	`light5`
6	`light6`
7	`light7`
8	`light8`

Strobe direction values

Strobe directions computer command	Effect
`forwards`	Strobes lights in a forwards direction
`backwards`	Strobes lights in a backwards direction

`reverse`	Same as backwards, strobes lights in a reverse direction.
	For example, the following two commands do the same thing:
	
	`strobe_all_lights(backwards, 15, 0.05);`
	`strobe_all_lights(reverse, 15, 0.05);`

Light commands

Command	Parameters	Description	Example

`light_on`	light number	Turns on the given light.
		Light stays on until turned off, or a flash command executed for the same light.
			`light_on(light1);`
			`light_on(light6);`
			`light_on(5);`

`light_off`	light number	Turns off the given light.	`light_off(light1);`
			`light_off(light6);`
			`light_off(8);`

`all_lights_on`		Turns on ALL lights.	`all_lights_on();`
`all_lights_off`		Turns off ALL lights.	`all_lights_off();`

`flash_light`	light number, duration time	Flashes the given light for the given duration time.
		Note that the flash frequency is 2 x the given flash duration time.
		Flashing continues until turned off, or another light command executed for the same light.
		Any number of lights may be flashing at the same time each with different flashing durations.
			`flash_light(light2, 0.5);`
			`flash_light(light3, 0.25);`
			`flash_light(light8, 1.0);`

`strobe_all_lights`	strobe direction, strobe cycles, duration time	Strobes all lights in the given direction the number of given complete cycles given and illuminates each light in each cycle for the given duration time.	`strobe_all_lights(forwards, 10, 0.1);`
			`strobe_all_lights(backwards, 20, 0.05);`
			`strobe_all_lights(reverse, 50, 1);`

`wait`	duration time	The computer will wait for the given duration time during which time no new commands can be executed. Note that any lights set to flash will continue to flash while the wait command is pending.	`wait(3.5);` `wait(0.25);`