Sonic Pi
API Cheatsheet
v1.0

Designed by
Sam Aaron & Carrie Anne Philbin

API

play(note, *args)		
Arguments	`note`	The note to play (as a MIDI number)
	`*args`	An optional list of arguments to pass to the current synth.
Description	Play the current synth at the specified pitch (supplied as a MIDI note). Optional arguments may be passed to further control the synth. These optional arguments are unique to each synth and are defined as part of the synth description.	
Related	See `with_synth` for changing the current synth and `play_synth` for playing a specific synth.	
Examples	`play 60` `play 60, "attack", 1`	

play_synth(synth_name, *args)		
Arguments	`synth_name`	The name of the synth to play.
	`*args`	An optional list of arguments to pass to the specified synth.
Description	Play the specified synth with the supplied arguments. Ignores the current synth setting. The optional arguments are unique to each synth and are defined as part of the synth description.	
Examples	`play_synth "pretty_bell"` `play_synth "pretty_bell", "note", 60, "attack", 1`	

repeat(&block)		
Arguments	`&block`	The block of code to repeat.
Description	Repeat the block of code forever. Warning, this code will not terminate - so any code below will not get executed. Consider putting inside a thread with `in_thread`.	

Examples	`repeat do`
	`  play 60`
	`  sleep 0.25`
	`end`	

with_tempo(n)		
Arguments	`n`	The new tempo in beats per second (BPS)
Description	Sets the current tempo to the new value in BPS. This value is global and the same for all threads.	
Examples	`with_tempo 200`	

current_tempo		
Arguments		
Description	Returns the current tempo in beats per second (BPS)	
Examples	`current_tempo` `with_tempo current_tempo * 2`	

play_pattern(notes, *args)		
Arguments	`notes`	A list of MIDI notes to play.
	`*args`	Optional arguments to pass to the current synth.
Description	Plays the specified MIDI notes with the current synth at the current tempo.	

play_pattern(notes, *args)	
Example	`play_pattern [40, 42, 44, 45]` `play_pattern [40, 42, 44], "attack", 1`

play_pattern_timed(notes, times, *args)		
Arguments	`notes`	A list of MIDI notes to play.
	`times`	A list of separation times between each note in seconds.
	`*args`	Optional arguments to pass to the current synth.
Description	Plays the specified MIDI notes with the specified separation times with the current synth at the current tempo. The list of times is a list of durations to be slept between each note. For example, a times list of `[1, 2, 0.5]` would result in a sleep of 1 second between, the first and second notes, a sleep of 2 seconds between the second and third notes and a sleep of 0.5 seconds between the third and fourth notes. If there are more notes, the times are re-used in a circular fashion. For example if there is a fifth note, the time between the fourth and fifth notes will be 1 second, and if there's a sixth note, the time between the fifth and sixth notes will be 2 seconds etc. If there are fewer notes than times, the remaining times are ignored.	
Example	`play_pattern_timed [40, 42, 44, 45], [0.5, 0.5, 0.5]` `play_pattern_timed [40, 42, 44], [1, 2], "attack", 1`	

play_chord(notes, *args)		
Arguments	`notes`	A list of MIDI notes to play.
	`*args`	Optional arguments to pass to the current synth.
Description	Plays the specified MIDI notes simultaneously with the current synth	
Examples	`play_chord [40, 42, 44, 45]` `play_chord [40, 42, 44], "attack", 1`	

in_thread(&block)	
Arguments	*Code block*
Description	Runs the block of code in a separate thread. This means the code will execute concurrently to the following code after the call to `in_thread`.

Examples	`in_thread do`
	`  play 60`
	`  sleep 0.5`
	`end`

stop	
Arguments	*No arguments*
Description	Stop all sounds.
Example	`stop`

Available Synths	
Synth Name	Available Arguments
`dull_bell`	`note, attack, release`
`pretty_bell`	`note, attack, release`
`fm`	`note, divisor, depth`
`beep`	`note, attack, release`
`saw_beep`	`note, attack, release`