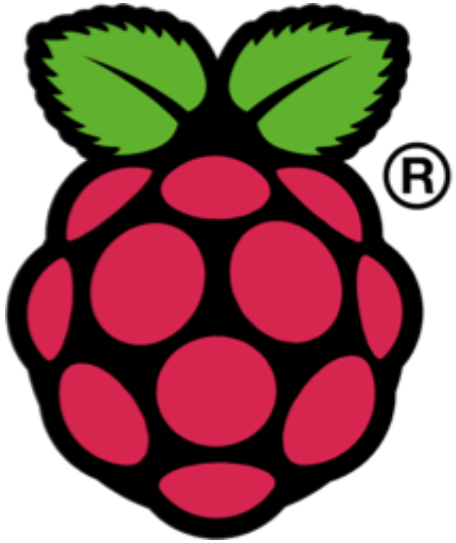
Live Coding

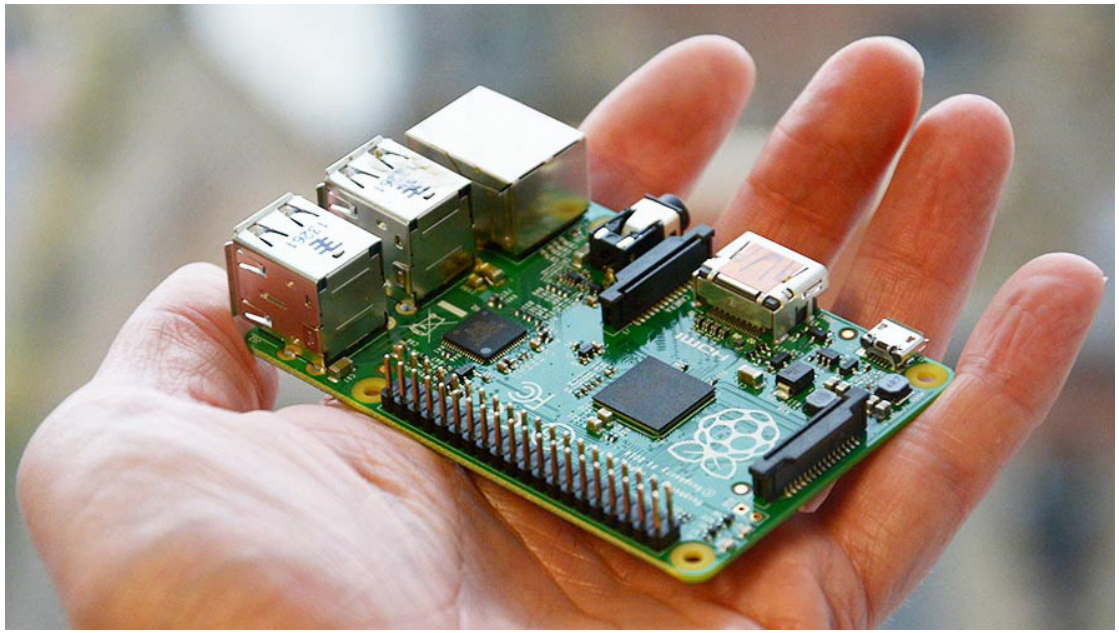
- A performing arts form featuring:
  - The writing of source code
  - The use of interactive programming
  - Improvisation, i.e. composing and playing music at the same time

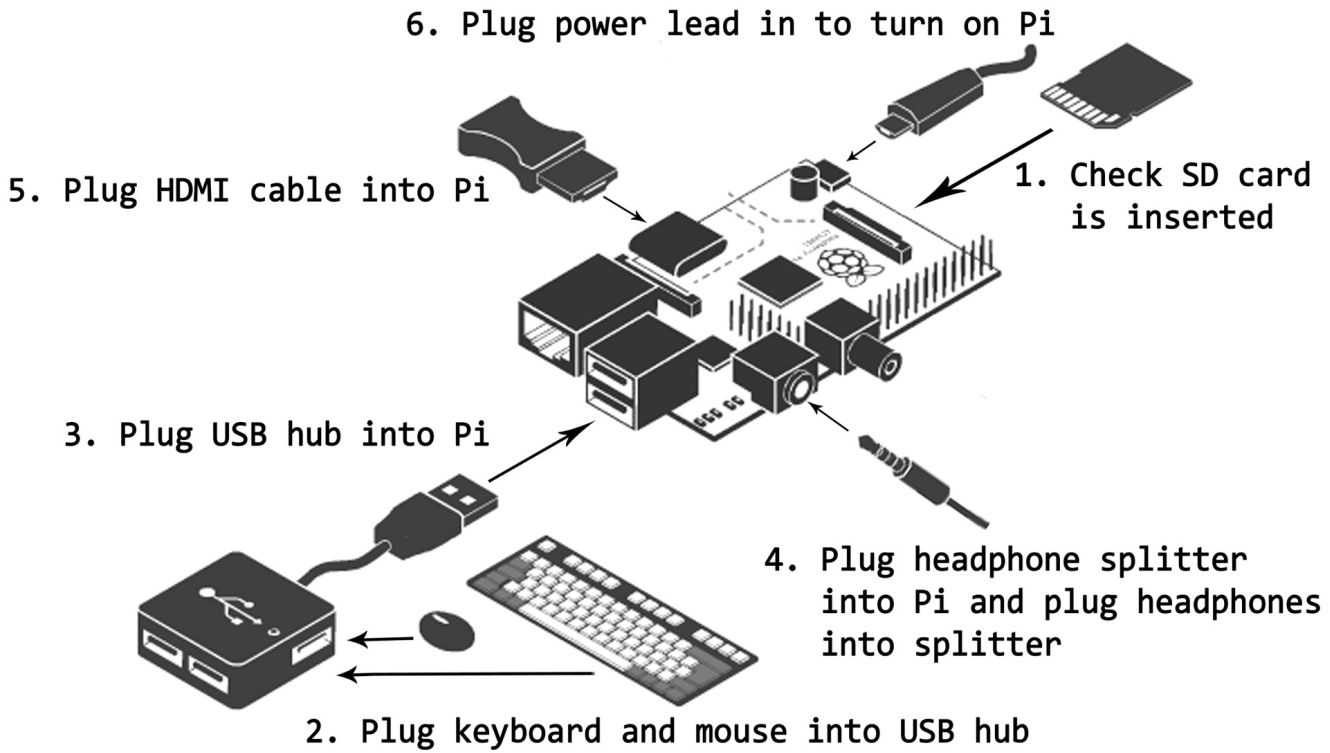
(((Sonic π)))

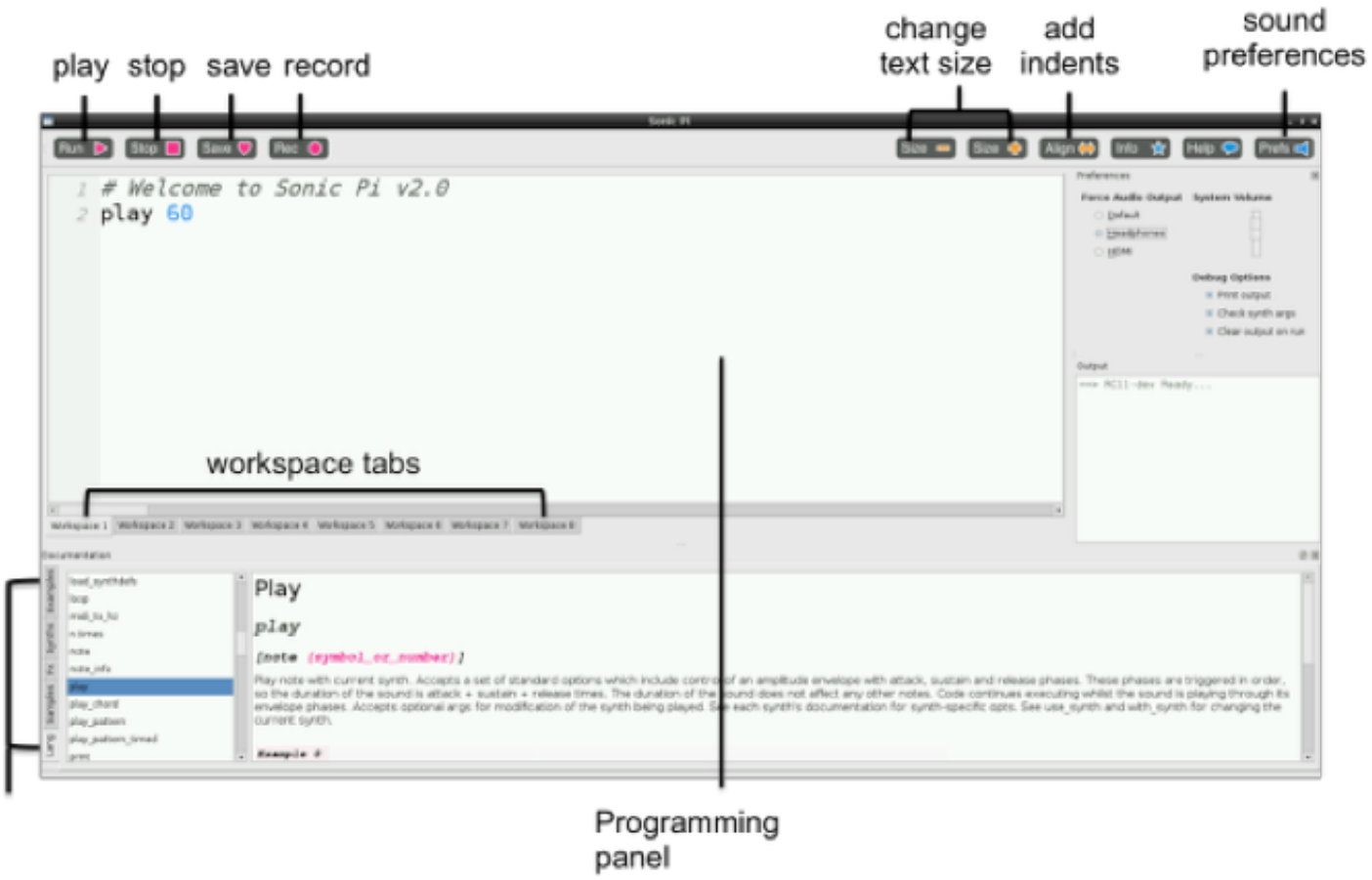
- Live coding environment
- Designed to support both computing and music lessons in schools
- Free and cross-platform
  - Windows, MacOS, Linux, Raspberry Pi



Sonic Pi: Live & Coding

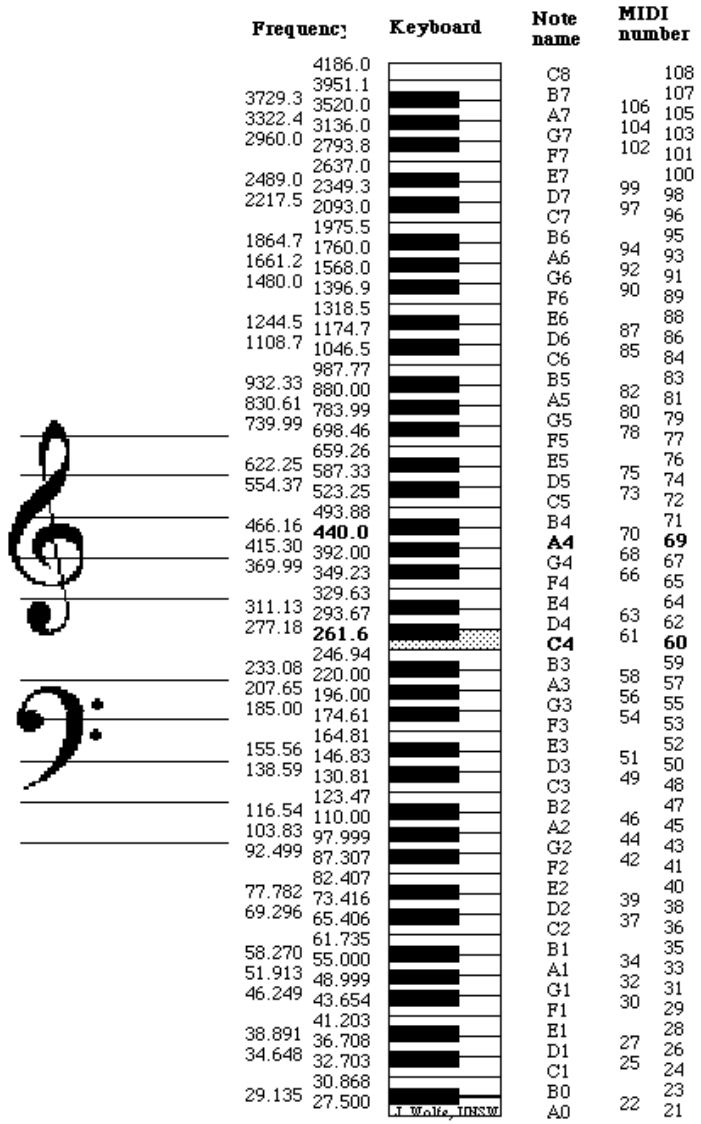


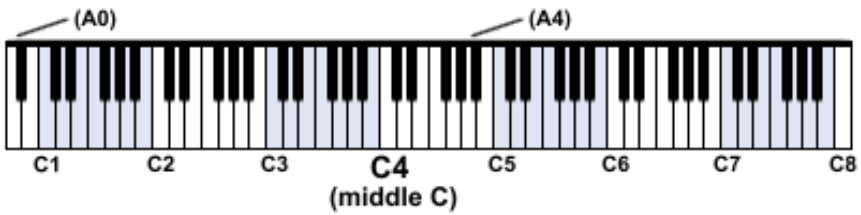


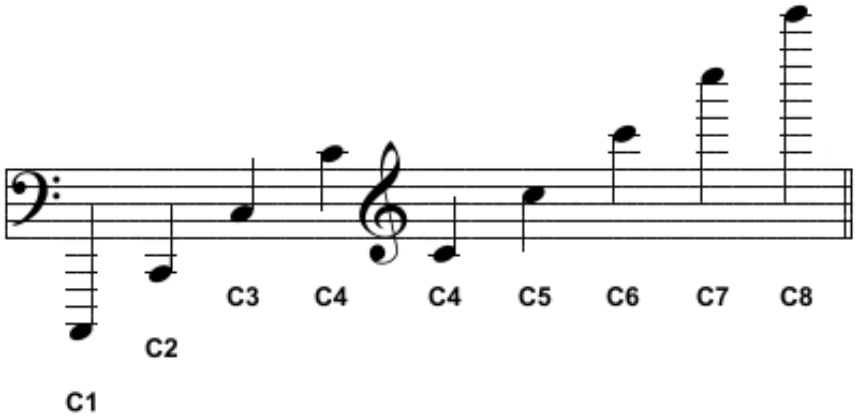


play

- play :C4
  - Play the note C4 (with default synthesizer)
- play 60
  - Play the note with MIDI number 60 (i.e. C4)
- play :60
  - What happen?

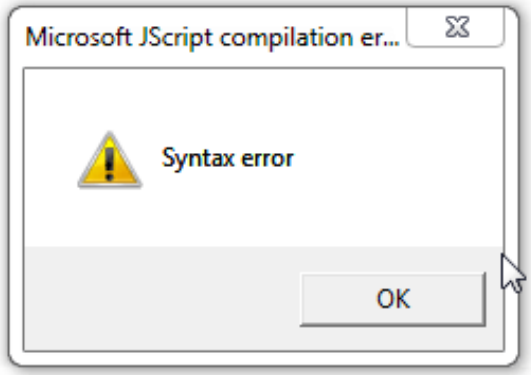






Syntax Error

- A character or string incorrectly placed in a command or instruction that causes a failure in execution
- i.e. A 'grammatical' mistake in the code based on the programming language being used



sleep

- sleep 1
  - Wait for one beat. Try and listen the differences of the two codes:
- play :C4
  play :E4
- play :C4
  sleep 1
  play :E4

use_bpm

- Use_bpm
  - Define the tempo, the default bpm is 60 (i.e. 1 second per beat)
- use_bpm 120

```
play :C4
sleep 1
play :E4
sleep 1
play :G4
```

use_synth

- Use a particular synthesizer (i.e. sound of an electronic instrument)
- use_synth :tb303

```
play :C4
sleep 1
play :E4
```

loop

- loop the quoted commands (The best thing computer can do)
- loop do

```
play :C4
sleep 1
play :E4
sleep 1
end
```

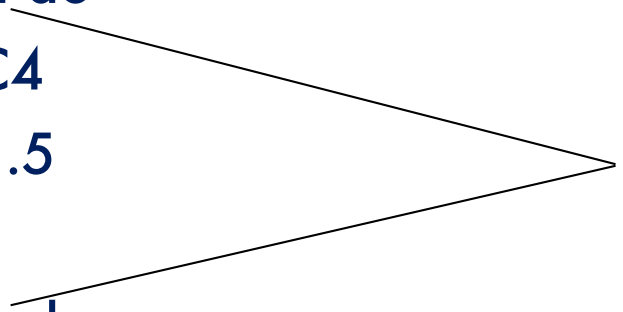
n.times

- Loop *n* times only
- 3.times do
    play :C4
    sleep 1
    play :E4
    sleep 1
  end

live_loop

- If you need more than one loop running at the same time

```
live_loop :a do
    play :C4
    sleep 1.5
end
live_loop :b do
    play :E4
    sleep 1
end
```



You need to give a name for each live_loop

Compare loop & live_loop

- live_loop :a do
    play :C4
    sleep 1.5
  end
  live_loop :b do
    play :E4
    sleep 1
  end

- loop do
    play :C4
    sleep 1.5
  end
  loop do
    play :E4
    sleep 1
  end

sample

- Play pre-recorded sample
- sample :perc_bell

rate

- Play the sample with a given speed, 1 = normal speed
- Therefore also changing the pitch of the sample
- sample :perc_bell, rate: 1.5
  sleep 1
  sample :perc_bell, rate: 1
  sleep 1
  sample :perc_bell, rate: -1.5

rrand

- Random number between a given range

```
loop do
    sample :perc_bell, rate: (rrand -1.5, 1.5)
    sleep rrand(0, 1)
end
```

choose

- Choose randomly between the given numbers
- loop do

```
    play choose([:C4, :E4, :G4])
    sleep 1
    play [:C4, :E4, :G4].choose
    sleep 1
end
```

tick

- Similar to choose but present the given numbers in sequence
- Stop when all the notes in the list has been played
- Add ring to tie the end of the loop back to the begining
- loop do

```
    play(ring :C4, :E4, :G4).tick
    sleep 1
end
```

scale

- Creates a series of MIDI notes for a given a tonic note and scale
- loop do

```
    play (scale :C4, :blues_minor, num_octaves: 2).choose
    sleep 0.5
end
```

amp

- Change the volume of a particular musical command
- loop do

```
    sample :bd_boom
    sleep 0.5
    sample :sn_dub, amp: 0.5
    sleep 0.5
end
```

with_fx

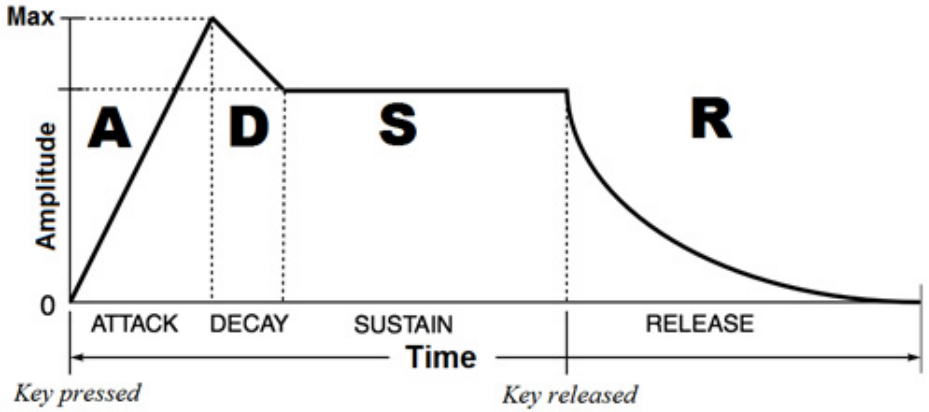
- Add sound effect to whatever you play
- (Optional) specifying parameters, see *Help*
- with_fx :echo do
    play :C5
  end

:sound_in

- Use your line in / mic of your computer as synth source
- Add sustain after it to indicate how long the mic in should be
- with_fx :pitch_shift, pitch: 6 do
  synth :sound_in, sustain: 3600
  end

attack, decay, sustain, release

- Add envelope to whatever you play



- play :C5, attack: 2, release: 6

Your Turn to Play Music

- Reference on teacher's demo
- Create your own ambient music with live coding
- Using the commands you have learnt: live_loop, use_synth, play, sleep, rrand, sample, rate
- Try to explain the choice of your artefacts – What theme is it?

Defining Patterns

- Define your own pattern with numbers (e.g. 1, 0, 0, 0)
- a = [1,0,0,0,1,0,0,0]

```
live_loop :b do
    a.each do |p|
        sample :bd_boom if p == 1
        sleep 0.125
    end
end
```

Making Drum Pattern

- a = [1,0,2,0,1,0,2,0,1,0,2,0,1,0,2,2]

```
live_loop :b do
    a.each do |p|
        sample :bd_boom if p == 1
        sample :drum_cymbal_closed
        sample :sn_dub if p == 2
        sleep 0.125
    end
end
```

sync

- Synchronize between different loops
- live_loop :a do
    sample :bd_boom
    sleep 1
  end
  live_loop :b do
    sync :a
    sample :drum_cymbal_closed, amp: 0.1
  end