



Sonic Pi

Lesson 3

Conditionals & Randomisation

*Designed by
Sam Aaron & Carrie Anne Philbin*



Licensed under CC-BY-SA 3.0

<http://creativecommons.org/licenses/by-sa/3.0/>

Conditionals & Randomisation

If we want to make some meaningful and interesting musical structures we need to learn some meaningful and interesting programming structures. In this lesson we will

| | |
|---------------------|---|
| Learning Objectives | Programs can make decisions. |
| | A simple form of decision is called a conditional. |
| | The potential decisions are called branches. |
| | Programs can contain random acts. |
| | Programmers use comments to explain interesting parts of a program. |

Teaching Progression

For the majority of the lesson, it is suggested that work is carried out in pairs. Each pair should have access to the standard equipment described below. In addition, is suggested that you have your own setup connected with a speaker for the demonstration sections.

Sonic Pi Equipment

- A Raspberry Pi with the Sonic Pi software installed per pair;
- A keyboard and mouse connected to the RPi per pair;
- A monitor connected to the RPi per pair;
- A headphone splitter connected to the RPi audio jack per pair;
- A pair of headphones connected to the splitter per pupil.

Equipment

- Computational cards with iteration and randomisation statements.

Lesson Summary

- Simple use of randomisation
- Using conditionals to make decisions.
- Combining randomisation and conditionals.
- Comments as a programming tool.

Starter

Pupils are first invited to set up and connect their Raspberry Pi hardware.

Main/development

1. Pupils should be shown how to add some randomisation to their code. This can be achieved by using the statement `rand(10)` which returns a random value between 0 and 10 (from 0 up to but not including the number you specify). You can specify other

Sonic Pi - A Computer Science Soundbite - Lesson 3

numbers for larger ranges i.e. **rand(20)** will return values from 0 to 20. Let's use this in our program by adding our random number to a note with the **+** operator:

```
3.times do
  play 60 + rand(10)
  sleep 0.5
end
```

Invite the pupils to observe the actual number of the note played in the output window.

2. Pupils are then asked to form a line as with previous lessons except this time, the line should split into two separate lines (called branches) at one point. The pupil directly before the split should be given the special **if** card and the rest of the pupils should be given cards as with previous exercises. The control card should start at the first pupil and passed down as each pupil carries out the action of their card. When it gets to the pupil with the special **if** card, they should toss a coin, and if it's heads, the the control card should be passed to the first of the two separate lines, otherwise the control card should be passed to the second of the two lines. Once the control card is in one of the separate lines then it continues until the end of that line, then the program is terminated. It should be pointed out that the line that didn't get the control card passed to it was essentially ignored. The **if** statement is called a conditional and allows for decisions to be made in the program.
3. Pupils should then be shown how to write an **if** statement in the Sonic Pi application. Ask them to copy the following code into their worksheet:

```
if rand < 0.5
  play 60
  sleep 0.5
  play 62
else
  play 72
  sleep 0.25
  play 71
  sleep 0.25
  play 70
end
```

4. Once they have copied this code, point out the syntax of the **if** statement - specifically the words **else** and **end**. These are similar to the **do** and **end** found in the iteration block discussed in the previous lesson in that they're like punctuation. The **else** separates the two different branches of the **if** statement.
5. The first line should also be discussed, this is equivalent to a coin toss in that **rand(1)** returns a random value between 0 and 1, and we're testing to see if that random value is less than 0.5. For the advanced pupils, you may wish to point out that changing the 0.5 to different values will affect the probability of which branch is selected. For example, a value

Sonic Pi - A Computer Science Soundbite - Lesson 3

of 0.1 would mean that (on average) every 10 runs only once will the first branch be selected, and the second branch will be selected 9 times.

6. The pupils are then invited to press the play button a number of times so that they can hear the different branches being executed with the decision of which to execute being random each time.
7. Pupils are invited to play around with the constructs of this less in addition to everything they've learned so far to design simple a musical program.

Plenary

Finally teach the class that the hash symbol **#** is used to make a comment. Invite them to place comments in their code to explain what is happening. This is not just for other programmers who might read their code, but for themselves in the future when they look back at old code they may have written a long time ago and forgotten about. For example:

```
# Toss a virtual coin
if rand(1) < 0.5
  #if heads, play two ascending notes
  play 60
  sleep 0.5
  play 62
else
  # if tails, play three descending notes
  play 72
  sleep 0.25
  play 71
  sleep 0.25
  play 70
end
```

| | |
|-------------------------------------|---|
| All students are able to... | Play a random note. Use an if statement. Place a comment in their code. |
| Most students are able to... | Play random notes with varying randomness (by supplying a different argument to rand .) Write useful comments. Modify code in the if branches to create different execution paths. |

| | |
|-------------------------------------|--|
| Some students are able to... | <p>Use rand to sleep for a random amount of time.</p> <p>Understand that using rand as a test with < allows for different probability distributions other than 50/50.</p> <p>Nest if statements inside other if statements.</p> <p>Have an awareness that comments are about communicating intent.</p> |
|-------------------------------------|--|