

# Learning during the COVID-19 outbreak

---

## Introduction

With the onset of the COVID-19 outbreak many students may be at home and be continuing to learn using lessons supplied by schools.

In response to this the Digital Technologies Hub team has produced some suggestions as to how students can learn about Digital Technologies in connection with other learning areas.

Here are a few suggestions.

## Digital Technologies + English

### Years 3–4 and years 5–6

#### Want to include some artificial intelligence (AI) with your creative writing?

You need a device that has a microphone.

This tool uses speech recognition, a form of AI.

It saves you typing your story.

Open the application [Voice Notepad](#).

Tell your story by speaking into your mic.

Save your work and edit your story.

How well does the AI recognise what you are saying?

Share the story you created using AI.

## Digital Technologies + Mathematics

Try some maths and coding challenges.

### Chance is an interesting area of maths.

What is the likelihood of an event happening?

For example, what is the likelihood that the sum of two dice rolled will total 12?

We can investigate this. Here's how.

### Two dice challenge

Complete this table with your predictions for the sum of two dice rolls.

The trial is for 100 rolls.

How many times do you predict the sum will equal these numbers in the table for 100 trials?

<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>

Now use this [Scratch program](#) to roll two dice and record the data for 100 trials. How close were you with your predictions?

Here's a [Scratch program](#) to predict heads or tails. Record data for 25 trials and then 50 trials and finally 100 trials. Do you see any patterns?

### **Guess a random number (higher or lower)**

Try this [Scratch program](#) which asks you to guess a random number. Remix the project. Try changing the range of numbers that you guess between. Add a scoring system.

Try creating your own [Scratch](#) program that uses random numbers. Create your own free account or log in to your own account.

## **Digital Technologies + Languages**

### **Years 3–4 and years 5–6**

Here are some fun coding ideas.

Try translating English into another language. This [program](#) built in Scratch translates English into German. Try remixing the program for another language.

Try our [chatbot](#) program. Abbey is a visitor from overseas. Ask Abbey questions to find out more about her travels. Remix the project.

## **Digital Technologies + The Arts: Music**

Try this [Basic piano](#) created in Scratch. Play a tune. Record your tune.

Here's a [Duet](#). See if you can remix the program. You might even add your own tune to replace the existing melody.

## **Digital Technologies: Data**

### **Years 3–4**

#### **Learn how to represent data in different ways.**

Represent a word in a different way by encoding the word. Create secret messages for your family to decode.

1. Write a simple message coded using a substitution of a number for each letter; for example, A=1 and Z=26. See how long it takes to 'crack the code' or 'decipher the message'.
2. Print out two columns containing the letters of the alphabet. Have one column in the correct order. But for the second, offset each letter by 2. For example, A would become C. Using this method, you can code words with each letter offset by 2. So 'dog' becomes 'fqj'.

Try this [online tool](#) to create a message for your family to decode.

### Years 5–6

#### **Represent data as binary numbers 1 or 0.**

Explore this [example](#) of an image to decode.

See if you can work out the way binary numbers are used to code the image.

Make your own instructions for your own image using binary numbers. Ask someone in your family to work out the image.

Learn more about this on Code Studio by viewing this [video](#). Then explore puzzle 1.

## Programming

### Years 7–10

#### **Basic coding course: Beginners**

Do you have students who want to learn basic coding skills? Try our [programming course](#).

This sequence shows visual coding using Scratch examples and how to move to general purpose programming languages such as Python or JavaScript. There are eight lessons that students can follow to build their skills. Lessons include video tutorials and code snippets.

The course is designed to be self-paced.

### Years 7–10

#### **Coding course Chatbots: Intermediate**

Try the AI-related course [Coding a sentimental chatbot in Python](#). This course explores natural language processing, a significant application of AI.

Teachers and students are led through the coding in Python of a chatbot, a conversational program capable of responding in varied ways to user input, including with the use of smart sentiment analysis. Each tutorial includes optional tinker challenges. The initial videos in the sequence will suit programmers with beginner skills in Python, with programs growing in complexity towards the end of the sequence. View the [overview video](#) for more information.