

## Unit: 5.1

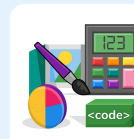
### Coding

#### Key Learning

- To begin to simplify code.
- To create a playable game.
- To understand what a simulation is.
- To program a simulation using 2Code.
- To know what decomposition and abstraction are in computer science.
- To take a real-life situation, decompose it and think about the level of abstraction.
- To understand how to use friction in code.
- To begin to understand what a function is and how functions work in code.
- To understand what the different variables types are and how they are used differently.
- To understand how to create a string.
- To understand what concatenation is and how it works.

#### Key Resources

purple  
mash



Tools



2Dos



2Chart



Free code gorilla

#### Key Vocabulary

##### Abstraction

A way of de-cluttering and removing unnecessary details to get a program functioning.

##### Action

The way that objects change when programmed to do so. For example, move or change a property.

##### Algorithm

A precise step by step set of instructions used to solve a problem or achieve an objective.

##### Concatenation

The action of linking a mixture of strings, variable values and numbers together in a series.

**Debug\ Debugging** Fixing code that has errors so that the code will run the way it was designed.

##### Decomposition

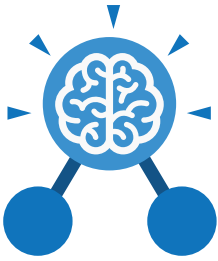
A method of breaking down a task into manageable components. This makes coding easier as the components can then be coded separately and then brought back together in the program.

##### Efficient

In coding, simplified code runs faster and uses less processing memory, it is said to be more efficient.

##### Flowchart

A diagram that uses specifically shaped, labelled boxes and arrows to represent an algorithm as a diagram.



## Unit: 5.1

### Coding

#### Key Vocabulary

##### Event

An occurrence that causes a block of code to be run.

The event could be the result of user action such as the user pressing a key (**when Key**) or clicking or swiping the screen (**when Clicked, when Swiped**) or when objects interact (collision). In 2Code, the event commands are used to create blocks of code that are run when events happen.

##### Nesting

When coding commands are put inside other commands. These commands only run when the outer command runs.

##### Physical System

In this context, this is any object or situation that can be analysed and modelled. For example modelling the function of a traffic light, modelling friction of cars moving down surfaces or modelling the functions of a home's security system.

##### Function

A block or sequence of code that you can access when you need it, so you don't have to rewrite the code repeatedly. Instead, you simply '**call**' the function each time you want it.

##### Object

Items in a program that can be given instructions to move or change in some way (action). In 2Code Gorilla, the **object types** are button number, input, text, shape turtle, character, object, vehicle, animal.

##### Properties

These determine the look and size of an object. Each object has properties such as the image, scale and position of the object.

##### Selection

A conditional decision command. When selection is used, a program will choose which bit of code to run depending on a condition. In 2Code selection is accomplished using '**if**' or '**if/else**' statements.

##### Input

Information going into the computer. This could be the user moving or clicking the mouse, or the user entering characters on the keyboard. On tablets there are other forms such as finger swipes, touch gestures and tilting the device.

##### Output

Information that comes out of the computer e.g. **sound, prompt, alert or print to screen.**

##### Repeat

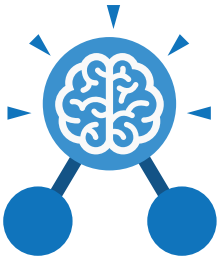
This command can be used to make a block of commands run a set number of times, until a condition is met or forever.

##### Sequence

This is when a computer program runs commands in order.

##### Simplify

In coding this is used to describe modifying the code to complete the same process with less lines of code.



# Unit: 5.1

## Coding

### Key Vocabulary

#### Timer

Use this command to run a block of commands after a timed delay or at regular intervals.

#### Variable

A named area in computer memory. A variable has a **name** and a **value**. The program can change this variable value. Variables are used in programming to keep track of things that can change while a program is running. In 2Code, variables can be **strings**, **numbers** or **computer-generated** variables to control objects of a type.

### Key Images



Design

Open design mode in 2Code.

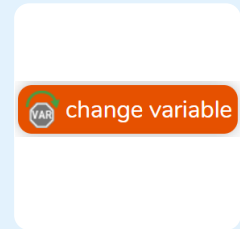


Exit Design

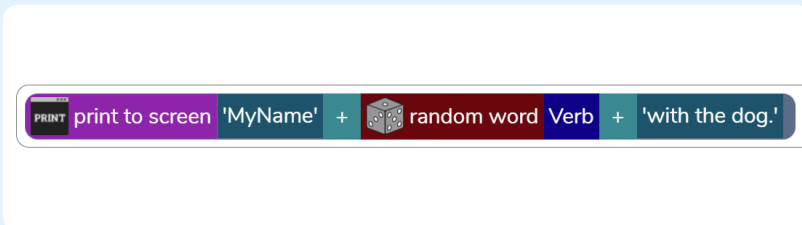
Switch to code mode in 2Code.



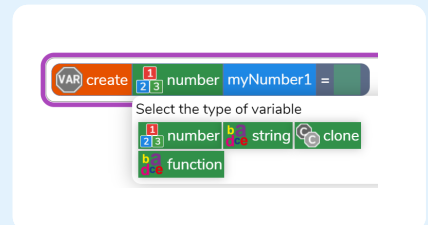
Add a new Tab to your code



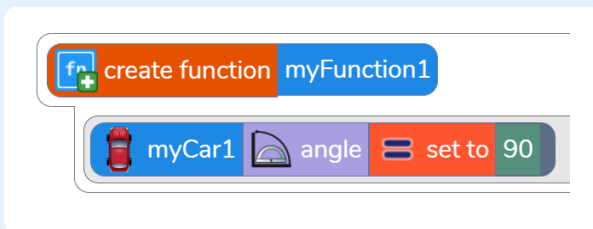
A change variable block.



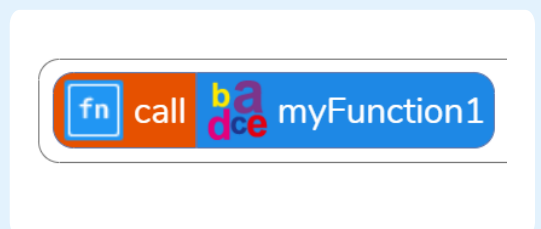
Example of combining variables and strings to print to the screen



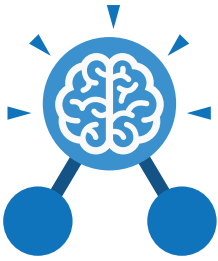
Creating a variable in 2Code



Creating a function in 2Code



Calling a function in 2Code



## Unit: 5.1

### Coding

#### Key Questions

#### What does simulating a physical system mean?

Creating a program where the objects behave as they would in the real world. For example, a football program that uses angles, speed and friction to simulate kicking a football. When simulating a physical system, you first must break the system down into parts that can be coded (decomposition). The different parts will come together to make the full simulation.

#### Describe how you would use variables to make a timer countdown and a scorepad for a game.

**Timer countdown:**  
Create a timer variable and set it to the starting number of seconds. Add a Timer command that repeats and subtracts 1 every second. Add a text object in design view to display this number.

**Score:**  
Create a variable to store the score, each time the user gains a point, change and display the value of the variable.

#### Give examples of how you could use the Launch command in 2Code.

Clicking on a button or other object in the program to opens another 2Code program or a webpage.

#### What do the terms decomposition and abstraction mean? Use examples to explain them.

Decomposition is breaking a task into its component parts so that each part can be coded separately. If you were coding a game of chess, you could decompose into the moves of the different pieces and the setup of the playing space.

Abstraction is removing unnecessary details to get the program functioning. In the example, the colour and size of the squares is not important to game play.