

# Australian Curriculum: Digital Technologies

## Year 3/4 Beginners Blockly Challenge

### Overview

<https://groklearning.com/course/aca-dt-mini-34-bk-variables/>

#### Table of contents

<b>Introduction</b>	<b>1</b>
<b>Mapping against the Australian Curriculum: Digital Technologies</b>	<b>2</b>
3/4 Band	2
5/6 Band	3
<b>What are students learning?</b>	<b>4</b>
<b>Synopsis</b>	<b>4</b>
<b>Module overview</b>	<b>4</b>
Module 1: Hello World	4
Module 2: Variables	5
Module 3: Using inputs	6
Module 4: Practice with variables	7
Module 5: Time to decide	8
<b>Related topics</b>	<b>9</b>

## Introduction

The Beginners Blockly Mini-Challenge is an interactive entry-level Blockly coding challenge. Students learn to use inputs, outputs, variables and to make decisions in their programming. The challenge focuses on food, with short challenges to create programs generating fun food combinations, and food critic comments in the style of our favourite television cooking shows.

The course addresses a number of the content descriptors of the Australian Curriculum: Digital Technologies and is a suitable entry point into teaching Digital Technologies for the year 3/4 and 5/6 cohorts.

A key emphasis of the course is variables. Students will learn to create variables, store data in variables, and access variables to retrieve data. Although variables aren't mentioned explicitly in the Australian Curriculum: Digital Technologies, they underpin the collection and representation of data. Variables are also an integral part of well designed digital solutions.

## Mapping against the Australian Curriculum: Digital Technologies

### 3/4 Band

Content Descriptor Code	Content Descriptor	Key Concepts	Addressed by Variables Challenge through:
ACTDIP009	Collect, access and present different types of data using simple software to create information and solve problems	Data collection Data interpretation	Students store data in variables, and access that data to solve problems (for example <a href="#">fries with that</a> )
ACTDIP010	Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them.	Defining (Specification) Decompose problem Functional Requirements Constraints	Using flowcharts to describe simple algorithms (for example <a href="#">flowcharts</a> )
ACTDIP011	Implement simple digital solutions as visual programs with algorithms involving Branching (decisions) and user input	Designing (Algorithms) Tracing	Creating projects requesting a user input and tailoring the output to the user's input (for example <a href="#">do you need chopsticks?</a> )

ACTDIP013	Plan, create and communicate ideas and information independently and with others, applying agreed ethical and social protocols	Plan, create and communicate with others Manage projects Safety Social Contexts	Problems require students to plan steps and decompose problems (for example <a href="#">drumroll please</a> )
-----------	--	--	---

## 5/6 Band

Content Descriptor Code	Content Descriptor	Key Concepts	Addressed by Tree through:
ACTDIP016	Acquire, store and validate different types of data, and use a range of software to interpret and visualise data to create information	Data collection Data interpretation	Students acquire and store data in their projects by creating variables, and use blockly programming to interpret data to create information (for example <a href="#">drumroll please</a> )
ACTDIP017	Define problems in terms of data and functional requirements drawing on previously solved problems	Defining (Specification) Decompose problem Functional Requirements Constraints	Students define problems and apply prior knowledge acquired in the course to solve problems (for example <a href="#">drumroll please</a> )
ACTDIP019	Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition)	Designing (Algorithms) Flowcharts Tracing	Using flowcharts to describe simple algorithms (for example <a href="#">flowcharts</a> ). This course does not however include iteration.
ACTDIP020	Implement digital solutions as simple visual programs involving	Branching Iteration	Creating projects requesting a user

	branching, iteration (repetition), and user input	Tracing	input and tailoring the output to the user's input (for example <a href="#">do you need chopsticks?</a> ). This course does not however include iteration.
ACTDIP022	Plan, create and communicate ideas and information, including collaboratively online, applying agreed ethical, social and technical protocols	Plan, create and communicate with others Manage projects Safety Social Contexts	Problems require students to plan steps and decompose problems (for example <a href="#">drumroll please</a> )

## What are students learning?

In this coding challenge, students learn about programming in Blockly, including decomposition, user input, branching and variables. This broadly covers the key concepts of the Australian Curriculum: Digital Technologies for years 3 and 4, and many of the concepts for years 5 and 6.

## Synopsis

Students create short projects to learn about user input, output, storing data in variables, and making decisions (branching). For example students will create a project asking a user for their favourite food, and the output is a comment about that food (eg 'What is your favourite food? If the answer is cheese, the output will be 'I love cheese too!').

As the course builds in complexity, students create projects saving multiple pieces of data in variables, such as user name, favourite food, and a comment about that food. The output will then combine these variables with text, and using branching to create varied responses. The course emphasises variables, with many opportunities for students to practice asking a user for input, saving a response, and creating an output based on that response.

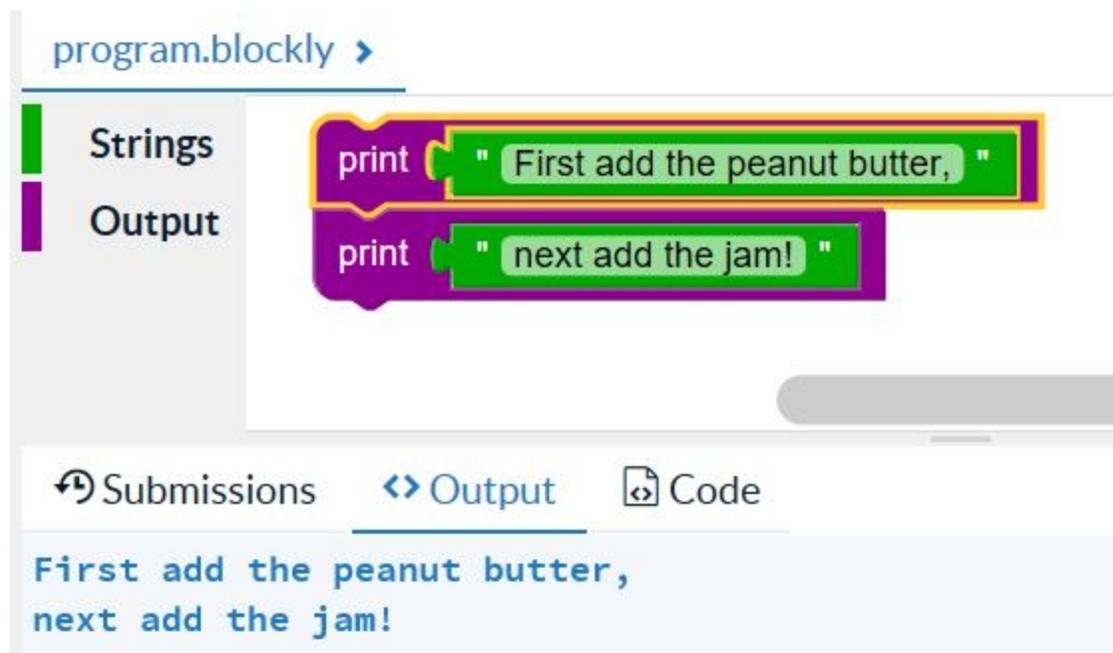
## Module overview

The challenge consists of five modules, which are summarised below.

## Module 1: Hello World

This module sets the foundation of the coding challenge. Students learn how to create a project using the Grok interface. They learn how to combine blocks to create a sequence of instructions. They learn to run code, and submit code for marking, then make necessary adjustments to their code to create a successful submission.

The modules introduces students to the key idea of combining a print command (an output) with a string (some text) to output a message on screen. Students are guided through how to combine different coding blocks to create outputs combining more than one string, or outputs over multiple lines combining more than one print command as shown in the picture below.



Log on to Grok and start module 1:

<https://groklearning.com/learn/aca-dt-mini-34-bk-variables/output/0/>

## Module 2: Variables

This module introduces students to variables. They are guided through creating a variable, saving data to a variable, accessing the data saved in the variable and using it in their project. Students will learn that:

- a variable can be reused multiple times; and
- the data stored in a variable can change.

By the end of the module students can predict what a given output will be when code combines a variable and a string (text.)

The screenshot shows a Blockly program in the 'program.blockly' editor. The code is as follows:

```

set flavour to "vanilla"
print "Don't forget to add " + flavour + " to your cupcakes!"
print "Your cupcakes must contain " + flavour + " !"

```

The output window displays the following text:

```

Don't forget to add vanilla to your cupcakes!
Your cupcakes must contain vanilla !

```

Log on to Grok and start module 2

<https://groklearning.com/learn/aca-dt-mini-34-bk-variables/variables/0/>

## Module 3: Using inputs

In this module, students learn to take input from a user, and save the input in a variable. Students learn that this allows their projects to be more flexible, without the need to edit code directly to generate a different output.

Students are given opportunities to practice asking a question, saving a response, and generating an output which combines a string (text) and data saved in a variable. Students also answer questions to check their knowledge of inputs and outputs.

See below for a problem in the module and its output.

program.blockly >

Strings  
Output  
Input  
Variables

set choice 1 to ask " Choice 1? "

set choice 2 to ask " Choice 2? "

print choice 1 " and " choice 2

print choice 2 " and " choice 1

Submissions <> Output Code

Choice 1? fish  
Choice 2? chips  
fish and chips  
chips and fish

Log on to Grok and start module 3

<https://groklearning.com/learn/aca-dt-mini-34-bk-variables/input/0/>

#### Module 4: Practice with variables

The fourth module does not introduce any new content. Rather, it allows students to consolidate their learning from modules 1 to 3. As students become more familiar with creating projects using the Grok platform, the scaffolding provided to students to solve problems is reduced. Rather than having step by step instructions and code blocks provided for them, they are transitioned to planning their project and accessing code blocks on their own. The images below show the solution for the final problem in the module.

The screenshot shows a programming environment with a sidebar on the left containing 'Strings', 'Output', 'Input', and 'Variables'. The main workspace contains a script with the following blocks:

- set event to ask "What are you judging?"
- set comment to ask "How is it?"
- print "Your" event "really is" comment

Below the workspace, there are tabs for 'Submissions', 'Output', and 'Code'. The 'Output' tab is active, displaying the following text:

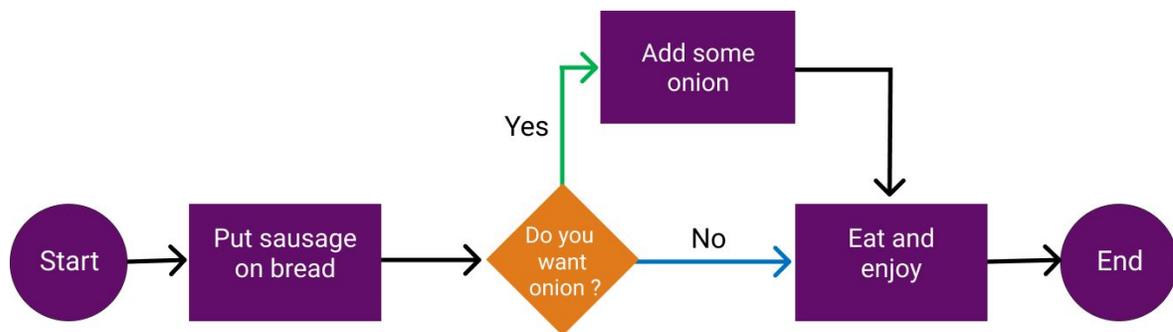
```
What are you judging? song
How is it? catchy
Your song really is catchy
```

Log on to Grok and start module 4

<https://groklearning.com/learn/aca-dt-mini-34-bk-variables/variablespractice/0/>

### Module 5: Time to decide

The final module introduces algorithms and branching (making decisions). Students ask users for an input and write code to generate differing outputs depending on that input, as shown in the flowchart below.



Students work over the module to create a final project which generates output for a fictitious cooking show judge. An example project output is shown below.

```
Item? frozen yoghurt  
Feedback? incredible  
Winner? yes  
Wow, that really was incredible frozen yoghurt  
It is time to announce...  
drumroll please...  
that you are our winner!
```

At the end of the module there is a playground environment, where students can create a project of their own, using the blocks they have explored in this course. Teachers may take the opportunity to set their own challenges using the playground (however the marking feature won't be available for these teacher-led challenges.)

Log on to Grok and start module 5, problem 0.

<https://groklearning.com/learn/aca-dt-mini-34-bk-tree/5/0/>

## Related topics

We expect students to get hungry while completing this challenge! The emphasis on food may encourage teachers and students to explore linkages to the food and nutrition aspects of the Australian Curriculum: Health and Physical Education for example:

- Identify and practise strategies to promote health, safety and wellbeing (ACPPSO36)
- Discuss and interpret health information and messages in the media and the internet (ACPPS039)
- Describe strategies to make the classroom and playground healthy, safe and active spaces (ACPPS040)

An additional fun activity for students is the jam sandwich activity, where students 'program' a teacher to make a sandwich. The students need to carefully consider each step of making a sandwich, and deliver precise instructions to ensure the sandwich making goes to plan! An example of this activity can be found [here](#).