



# Scratch Creative Computing Guide

## Summary

The [Scratch Creative Computing Guide](#) includes a teacher guide and learner workbook. There is also a series of units comprising learning activities, paired with assessment activities and templates that can be used to support use of the Scratch (MIT) platform. The Scratch Creative Computing Guide supports assessment activities with visual programming environments.

## Description

Assessment of programming projects often relies on the final product – the code and the functioning of the program. Student processes that underpin the design and development of the program are often neglected.

Activities and templates within this computing guide support students to make their learning journey visible. Students are guided through the process of self- and peer-assessment and programming project evaluations. In addition to the unit themes and learning activities, the guide includes a number of templates and activity outlines requiring students to document goals, processes and reflections, perform peer feedback, design and evaluate team projects, etc. The design, reflection and evaluation templates can be adapted to suit a range of year levels.

These activities and templates can be selected and used by teachers to support formative, or summative assessment in the classroom.

## Example templates

- Design journal: Students document their processes and reflections in a hardcopy or digital journal. The journal can be used to support formative assessment, and can be submitted with their final project to support summative assessment.
- Debug it challenges: Test students' understanding of blocks, code literacy, debugging skills and computational thinking processes.
- Critique group: This template supports peer feedback activities.
- Activity design: Students document their design goals.
- Hackathon: Students document their teamwork project progress. They design implementation and evaluation processes. This unit can guide the development of a teamwork project that could form a summative assessment project.

ScratchED have released an editable version, allowing educators to download and customise documents as needed.

## Year level bands

3–4, 5–6, 7–8





# ASSESSMENT DIGITAL TECHNOLOGIES

ADVICE ON HOW TO ASSESS DT

## CRITIQUE GROUP

FEEDBACK FOR: \_\_\_\_\_  
PROJECT TITLE: \_\_\_\_\_

FEEDBACK BY	[RED] What is something that doesn't work or could be improved?	[YELLOW] What is something that is confusing or could be done differently?	[GREEN] What is something that works well or you really like about the project?

Critique group template from the ScratchED Creative Computing Guide

## DEBUG IT! REFLECTIONS

NAME: \_\_\_\_\_

RESPOND TO THE FOLLOWING REFLECTION PROMPTS USING THE SPACE PROVIDED BELOW OR IN YOUR DESIGN JOURNAL.

**UNIT 1 REFLECTION**

+ What was the problem?

\_\_\_\_\_

+ How did you identify the problem?

\_\_\_\_\_

+ How did you fix the problem?

\_\_\_\_\_

+ Did others have alternative approaches to fixing the problem?

\_\_\_\_\_

Debug it! Reflections template from the ScratchED Creative Computing Guide





## Guidance for use

This computing guide can be used by teachers to support the design of visual programming learning activities. Teachers can implement or select from the learning activity units, or simply use the formative and summative assessment ideas and templates within the guide to support other visual programming activities (whether that be in Scratch MIT, or some other visual programming environment).

Suggestions for using this document

- Download and customise the guide to include content descriptions and achievement standards within the Australian Curriculum: Digital Technologies.
- Select assessment activities and templates from the guide (eg reflections, reports and critiques) that can be used to support assessment within your own existing visual programming activities, or for use with activities described in the guidebook.
- Provide students with the learner workbook from the start, to encourage design and planning so that progress can be documented throughout their learning journey.
- Adapt the design, reflection and evaluation templates for use with general-purpose programming activities in years 7–8.

## Australian Curriculum Digital Technologies alignment

### Years 3–4

Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them (ACTDIP010)

Explain how student solutions and existing information systems meet common personal, school or community needs (ACTDIP012)

Plan, create and communicate ideas and information independently and with others, applying agreed ethical and social protocols (ACTDIP013)

### Years 5–6

Design a user interface for a digital system (ACTDIP018)

Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition) (ACTDIP019)

Explain how student solutions and existing information systems are sustainable and meet current and future local community needs (ACTDIP021)

Plan, create and communicate ideas and information, including collaboratively online, applying agreed ethical, social and technical protocols (ACTDIP022)



## Years 7–8

Define and decompose real-world problems taking into account functional requirements and economic, environmental, social, technical and usability constraints (ACTDIP027)

Design the user experience of a digital system, generating, evaluating and communicating alternative designs (ACTDIP028)

Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors (ACTDIP029)

Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language (ACTDIP030)

Evaluate how student solutions and existing information systems meet needs, are innovative, and take account of future risks and sustainability (ACTDIP031)

Plan and manage projects that create and communicate ideas and information collaboratively online, taking safety and social contexts into account (ACTDIP032)