Visual to text coding: The Basics

**Go to lesson series** |  **Go to next lesson**

This series of lessons is to help students to transition from visual coding to text-based coding with a general-purpose programming language.

This section provides the basics in order to use the programming environments: Scratch, Python and JavaScript.

# Learning map and outcomes

In this lesson, students will:

# Learn how to concatenate data for output

# Learn how to store data in variables

# Put these together to make a very basic weather application

# The Basics

For information on setting up and choosing a language, see [*Setting up*](https://www.digitaltechnologieshub.edu.au/teachers/lesson-ideas/visual-to-text-coding-setting-up/).

The following videos provide the basics for using general-purpose programming languages.

**Concatenation**

This [video](https://youtu.be/7tfhswaGB_U?list=PL0dMM9mg1u6NKQNT1odqe-5dI-lEovDxm) discusses the process of combining data together to form a coherent output.

**Variables**

Variables are like labelled containers for storing data.

View the ‘[Variables' video](https://youtu.be/8BP81XucGF4) to learn more about how to create and use variables in your computer program.

Have students practise using variables. Interpret the following pseudocode and code it in Scratch **and** either Python or Javascript. Students should predict the output of the program (what it will display) **before** doing their code.

BEGIN

**a** ← 50

Display **a**

**b** ← 23

Display **b**

**c** ← **a** + **b**

Display **c**

END

Solution code: [Scratch](https://scratch.mit.edu/projects/307289473/), [Python](https://repl.it/%40digitechhub/Lesson-1-Learning-Construction-Step-), [JavaScript](https://jsfiddle.net/nathanesa/gejya8t2)

SIDEBAR – Strategies for coding and debugging

Students may work at vastly different paces when doing general-purpose programming, and some may seem to be quicker at spotting problems than others.

Try these strategies with your students:

* **Cheat sheets** bring together the most basic Python commands or JavaScript commands. See Resources at the bottom of this page.
* **Pedantic computer:** Think of the computer as a **really** pedantic person. It usually won’t cooperate if important punctuation (syntax) is missing, if something is spelled differently in two places, or even with the wrong upper or lower case.
* [**Pair programming**](https://www.youtube.com/watch?v=vgkahOzFH2Q) **or ‘Ask three before me:** Often it just takes a fresh pair of eyes to spot errors or bugs in a program. Before asking the teacher the student checks with other students.
* **Trace errors:** The computer always runs the program in order, line-by-line, according to the code. Study each line one-by-one and ask, ‘What does this line do?’ Consider writing down the values of variables as you go.
* **Tinker:** Encourage students to tinker with the code. Ask, ‘What would happen if I changed this?’ Alter a part of the code or add a new feature, predict what might happen, and run it to see. You can always put it back again if something goes wrong.

**About input/ output**

# This [video](https://youtu.be/8NZBr3DFTGc?list=PL0dMM9mg1u6NKQNT1odqe-5dI-lEovDxm) encourages users to build a simple weather application to consolidate knowledge learnt so far.

# Resources

* Online environments for coding in each language
	+ [**Scratch**](https://scratch.mit.edu/)
	+ [repl.it](https://repl.it/): an online environment suited to **Python**
	+ [JSFiddle](https://jsfiddle.net/): an online environment suited to **JavaScript**
* Cheat sheets listing basic commands for coding:
	+ Python [Cheatsheet](https://groklearning-cdn.com/resources/cheatsheet-python-1.pdf) (from Grok Learning)
	+ JavaScript [CheatSheet](https://htmlcheatsheet.com/js/) (Tip: Press the little blue tabs to move **Variables**, **Basics**, **Strings** and **Data Types** to the top.)