

## Intermediate Coding Using MakeCode Learning Objectives

Learning Objectives	Learning Outcome	CSTA K-12 CS Standards
P2.1) Develop a file storage strategy for a world being exported at various stages of completion	Learners will understand the importance of file versioning and backing up	
P2.2) Identify the difference between various standard programming blocks in MakeCode (e.g. Loops, Variables, Logic)	Learners will be able to identify standard programming blocks in MakeCode such as loops, variables and logic	2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.
P2.3) Predict, without running, the outcome of a series of MakeCode blocks	Learners will be able to understand the outcome of a program without running the code	2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
P2.4) Predict, without running, the outcome of a making series of changes to MakeCode blocks such as changing the order of blocks or variables	Learners will be able to demonstrate the ability to determine the differences in the output of a program, after making changes to the program, given only the program and its input without running the program	2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
P2.5) Identify an error in an existing MakeCode program (e.g. looping one too many/little times, incorrect conditions in conditional statements)	Learners will be able to identify errors in code	2-AP-19 Document programs in order to make them easier to follow, test, and debug.  2-AP-17 Systematically test and refine programs using a range of test cases

# CODING IN MINECRAFT



P2.6) Demonstrate the use of loops, variables and logic

Learners will be able to effectively use standard programming blocks in MakeCode such as loops, variables and logic

2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.

2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

P2.7) Given a problem develop and explain a MakeCode program to formulate a solution

Learners will be able to design and develop a programmatic solution for a given problem

2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

P2.8) Demonstrate the use of modularity

Demonstrate the ability to decompose problems into subproblems and create and use procedures

2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.