

Studio: Rock, Paper, Scissors in Python

40 minutes

Project studios (optional extension sessions)

Built in Python

Outcome 3.1

Outcome 3.9

WHAT THIS LESSON DOES

Students build a Rock-Paper-Scissors game in the micro:bit Python editor using lists, randomness, button input and conditionals, tracing a round by hand before they run and debug.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
3.1	creatively design and write code for short programming tasks to demonstrate the use of operators for assignment, arithmetic, comparison, and Boolean combinations
3.9	analyse code to determine its function and identify errors or potential errors

WHAT STUDENTS SHOULD BE ABLE TO DO

- Build the classic game in Python with conditionals and randomness (LO 3.1)
- Trace a round by hand before playing (LO 3.9)

BEFORE THE BELL

- Devices logged in to the Coding Ireland platform and ready for Python
- Open the micro:bit Python editor at python.microbit.org; one new project per student or pair
- Physical micro:bits and USB cables available if you plan to run the download step; otherwise stay in the simulator and use that time to play-test
- Board or screen ready to model the first imports and the choices list

WHAT YOU NEED

Item	Qty	Per	If you do not have it
tablet or laptop with Python	1	pair	run the lesson on the IWB with students taking turns

Studio: Rock, Paper, Scissors in Python

HOW THE LESSON RUNS

4 min	introduction	Start: what we're building
3 min	content	The plan: five parts to build
2 min	content	Import Modules
3 min	content	Define Choices List
5 min	content	Get User Choice
3 min	content	Generate Computer Choice
5 min	content	Determine the Winner
5 min	content	Add Game Loop
2 min	content	Download your Code
2 min	content	Wrap Up and Extensions
3 min	content	Make sense: what worked, what broke
2 min	content	Reflect
1 min	summary	What you covered

TEACHING MOVES

- Model typing the imports and the choices list on the board, and insist students type rather than paste so they notice spelling and indentation
- Before anyone runs the full compare block, pause and have the class trace one round on paper: the user choice, the computer choice, and which branch fires
- Check both buttons for scissors first in the if-elif chain, because if rock or paper is tested first a both-button press never registers as scissors
- When adding a persistent score, introduce two variables outside the outer loop and update them only inside the win or lose branches

WHAT TO WATCH FOR

- Indentation errors inside while True or if-elif: the editor will refuse to run, so realign the blocks to the same level as the model on the board
- button_a checked before both buttons: scissors never wins the input test, so reorder the chain to put both buttons first
- String mismatch in comparisons, such as an extra space or a capital letter: the scroll messages look fine but the winner logic always falls through, so match the list strings exactly

DIFFERENTIATION

- Support: Provide a printed trace table (user, computer, result) and stay on the simulator; pair stronger typists with peers who need help on indentation
- Extension: Add score variables that persist across rounds and scroll the running total after each result
- Extension: Add a shake-to-play input with the accelerometer as a second way to start a round