

Coding a Maze - Instructions

See the maze templates below

1. Try Maze 1

- Introduce the concept of “sequence”:
 - Code must be written in a specific order called a sequence
 - Just like a story wouldn’t make sense if the sentences were re-arranged in the wrong order, code won’t work if it’s written in the wrong sequence
 - This applies to the maze and also to coding more generally
- For this maze, hand out the following pre-cut coding instructions: *Go Forward, Turn Right, Turn Left, End*
- Kids need to create a long list of the instructions that they think the character needs to follow to reach the end of the maze
- The character must avoid the grey areas of the maze and only stay on the white squares
- Once kids have organized their list from top (first command) to bottom (final command), place the character at the start of the maze
- Check the code by moving the character and following the instructions in the list
 - Flip over each command as it is completed
 - Kids can make corrections as needed—young children often need corrections on left and right turns, as they must think about the turns from the characters’ perspective

2. Try Maze 2

- Introduce the concept of “loops”:
 - This is when you want to repeat steps in a sequence
 - Rather than piecing together three separate “move forward” commands, children can learn to use the code “for the next _ steps, move forward,” filling in the blank space with the amount of steps needed
- Hand out the following pre-cut coding instructions: *Go Forward, Turn Right, Turn Left, For ___ Steps, End*
- Kids need to create a long list of the instructions that they think the character needs to follow to reach the end of the maze
 - It’s a good habit to start indenting the line of code underneath the loops—this is required by some computer coding languages, and it also makes the language much more readable

- Once kids have organized their list from top (first command) to bottom (final command), place the character at the start of the maze
 - Check the code by moving the character and following the instructions in the list
 - Flip over each command as it is completed
 - Kids can make corrections as needed
3. Try Maze 3 (Note: a more advanced concept; for older children)
- Introduce the concept of “if-then-else” statements:
 - These will enable kids to think about writing as short a program as possible
 - An if-then-else statement is comparable to answering a *true or false* question—if the answer is true, a certain action occurs; if the answer is false, another action occurs
 - To get their character to walk in a straight line, kids could come up with the following code: “If > there is no wall > in front of me > go forward”
 - Hand out all of the coding instructions: *Go Forward, Turn Right, Turn Left, For ___ Steps, If, Else, Else If, There is a Wall, There is Not a Wall, In Front of Me, To My Left, To My Right, On All Three Sides*
 - Kids need to create a long list of the instructions that they think the character needs to follow to reach the end of the maze
 - There are many possible codes that can be written based on these options
 - Allow kids to experiment with the different options and see if they can write a variety of code to get their character to the end of the maze
 - There are many ways to get the character from start to finish, so keep on experimenting with different codes
 - Once participants have organized their list from top (first command) to bottom (final command), place the character at the start of the maze
 - Check the code by moving the character and following the instructions in the list
 - Flip over each command as it is completed
 - Participants can make corrections as needed
4. Try out the ten challenges on [maze blocky](#)

Coding Commands

Go Forward ↑	Go Forward ↑
Go Forward ↑	Go Forward ↑
Go Forward ↑	Go Forward ↑
Go Forward ↑	Go Forward ↑
Go Forward ↑	Go Forward ↑
Turn Left ↶	Turn Right ↷
Turn Left ↶	Turn Right ↷
Turn Left ↶	Turn Right ↷
Turn Left ↶	Turn Right ↷
Turn Left ↶	Turn Right ↷

FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END

IF	there is a wall	in front of me
IF	there is a wall	in front of me
IF	there is a wall	to my left
IF	there is a wall	to my left
ELSE IF	there is not a wall	to my right
ELSE IF	there is not a wall	to my right
ELSE IF	there is not a wall	on all 3 sides
ELSE IF	there is not a wall	on all 3 sides
ELSE	END	END
ELSE	END	END

Start							
			End				

Start							
							End