

Solution (Hints)

- Lesson 1: Learning Sequence
- Lesson 2: Learning Branch, Jump (), goto ()
- Lesson 3: Making decisions, Conditional – if ()
then else ()
- Lesson 4: Fixing Errors, Bug and Debugging
- Lesson 5: Looping with repeat, bounded loops
- Lesson 6: Understanding Functions

Bonus lessons

- Lesson A: Introducing operations, greater, less than. Boolean - TRUE, FALSE
- Lesson B: Introducing Variable, string and numeric
- Lesson C: Nested repeat – Loop in Loop

1. Sequence

Series of ordered actions or steps

Read Loud



Sequence for playing a game when in a group

- Decide on the game
- Share the rules
- Start playing

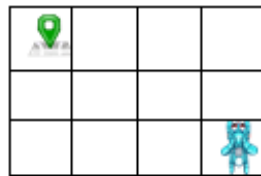
Read Loud

An example of sequence. (Forward is in direction of bunny's eyes)



Practice Exercise

Your turn to help bunny use



Practice Exercise

In 5 steps or less create a sequence of your morning routine

- 1) Wake up
- 2) Brush teeth and freshen up
- 3) Eat Breakfast
- 4) Wear School Dress
- 5) Go to School





1.1 Sequencing



Look at your code: Count how many code-cards you use in your coding today:

__ 4 __ MoveForward ();

__ 2 __ TurnLeft ();

__ 0 __ TurnRight ();



Using the symbol or text Write your Code for the game played!!



TurnLeft ();

MoveForward ();

MoveForward ();

MoveForward ();

TurnLeft ();

MoveForward ();

(THIS IS AN EXAMPLE CODE – YOUR CODE WILL DEPEND ON YOUR GAME PLAY)

2.Jump()

Jump and branch cause switching from normal path to different path Read Loud



What do you do when you find a puddle in your way? You don't go in puddle and get wet but jump



Read Loud

Jump!!

An example sequence showing jump



Practice Exercise

Your turn to help bunny. Use



Practice Exercise



Create another ways from bunny to goal.
Which one is better? Why? (the path with less cards is better since its faster, using less resources, overall efficient)



3. Conditional

Read Loud

Perform different action based on conditions. If (cond.) then (consequent) else (alternative)

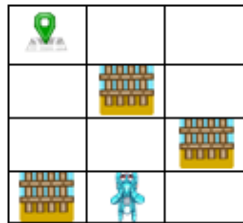


Conditional Operator

Read Loud

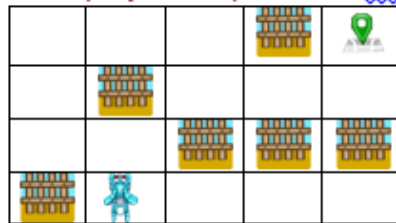
If (you are hungry) then Let's eat lunch else Eat after an hour

An example to turn around "if" there is a fence



```
MoveForward( );
If (Fence) TurnLeft( );
MoveForward( );
TurnRight( );
MoveForward( );
MoveForward( );
```

How many "if" would you need to reach ? (2 "if")



Practice Exercise

Practice Exercise

Create your own if then else (Hint : think about your 1st and 2nd favorite ice cream flavor and if you don't get 1st)

If (there's vanilla flavor available)
take a scoop
else
take chocolate flavor



4. Debugging

Debug means process of finding and fix an incorrect solution (bug)

Read Loud



Debugging a problem

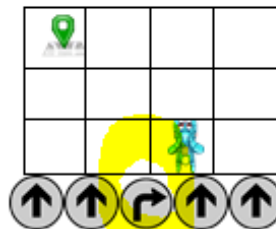
Read Loud

- Look at the bug in current solution
- Think of how to correct it (debug)
- Find and test the correct solution (fix bug)

Debug a code bug - The second move forward is extra! Bug! 



Your turn to help the bunny now, Find the bug the code!



Practice Exercise

Practice Exercise

Find 3 case where you did something wrong and then corrected it (bug-debug)

- 1) In breakfast picked up wrong cereal box, put it back and picked the right one
- 2) Biking to school took a wrong turn and corrected by taking the right turn
- 3) Light doesn't turn on, but the connection was loose. Debugged and fix the connect.



Read Loud

5. Loops

An efficient way of specifying iterations to allow code to execute over and over.



Multiple actions to loop

Read Loud

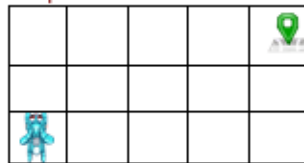
If someone asks you to do this -

Go Forward, Go Forward, Go Forward, Go Forward

You could also say

Go Forward – 4 times OR Repeat (4) – Go Forward

Circle the paths that could be converted to repeat loops



R2

Here's an example of converting to repeat



Practice Exercise

Convert this sequence to repeat by filling the blank



Practice Exercise

Think of 2 situations in your daily life where you could use repeat

- 1) Repeat (2) Drink a glass of water: (to drink 2 glass of water)
- 2) Repeat (5) Return Book to machine: (to return 5 books in the library)



6. Function

Read Loud

In this section kids learn Functions.

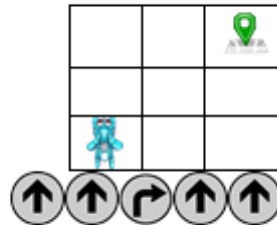


What is a function?

Read Loud

- A piece of code that can be called over and over.
- Function helps write efficient code

Function use



$f = \uparrow \uparrow$

$f \text{ } \curvearrowright \text{ } f$

Your turn, create code for bunny to reach destination using function already defined (Use : stair case function)



$f = \uparrow \curvearrowright \uparrow \curvearrowleft$

Complete code below -

$f \text{ } f \text{ } \uparrow$

Find few case of using function in day to day life

(Hint – Think routine things that you do over and over)

1) Morning Routine (Wakeup, Brush, Eat, Go to School)

2) Make a Sandwich (Bake Bread, Put Cheese, Put veggies, slice)

Practice Exercise



A. Operations

In this section kids learn Boolean and Operations.



Comparison

- > Greater than
- < Less than
- = Equal to

Boolean

- TRUE
- FALSE

Roll the dice; Note the dice score; Circle TRUE or FALSE

Hopsy = 2 Flopsy = 1 Bingo = 3 Disco = 2

example

- 1) Hopsy > Flopsy **TRUE** FALSE
- 2) Flopsy < Bingo **TRUE** FALSE
- 3) Disco > Flopsy **TRUE** FALSE
- 4) Bingo > Disco **TRUE** FALSE
- 5) Hopsy < Disco **TRUE** **FALSE**
- 6) Bingo > Hopsy **TRUE** FALSE

Practice Exercise

Roll

Fill : (Use = , > , <)

- 1) Disco > Flopsy
- 2) Hopsy > Flopsy
- 3) Bingo > Hopsy
- 4) Flopsy < Bingo
- 5) Bingo > Disco
- 6) Bingo > Hopsy

Practice Exercise

Practice Exercise

Create your own TRUE statement
(E.g. if Hopsy = 3, Flopsy = 2: Hopsy > Flopsy)

- 1)
- 2)
- 3)
- 4)



B. Variable

In this section kids learn about variable and string



Variable	Value
<u>Your_Name</u>	"<your_name>"
<u>Bunny_Name</u>	"Hopsy", "Flopsy"
<u>Dice_Roll</u>	0,1,2,3
<u>Number_of_Code_Cards</u>	0,1.....39,40

What's your Name?

1) Your_Name = Tom (or Anna)

What's your Bunnyz name?

2) Bunny_Name = Hopsy

Practice Exercise

What's your Dice Roll?

3) Dice_Roll = 2

Use of Variable/String

Hello_World Tom (or Anna) Practice Exercise !!

You got Hopsy !! Nice Bunny

{"Bunny_Name"}

Your Dice score is 2

{"Dice_Roll"}

Practice Exercise

Variables are like containers that take different values. Think about 3 examples of variables (e.g. days_of_week month_of_year)

- 1)
- 2)
- 3)



C. Loopy-Loop

In this section kids learn nested loops.



Multiple loops

Inner Loop(IL) and Outer Loop(OL)

```
Repeat (2) // Outer Loop (OL)
```

```
  Repeat (4) {Go Forward} // Inner Loop(IL)
```

Inner loop execute 2 times. Total 8 Go Forward

Circle the paths that could be converted to nested repeat loops



Here's an example of converting to nested repeat

```
↑ ↑ = Repeat(2) ↑
```

Convert this sequence to repeat by filling the blank



```
Repeat ( 2 ) // OL
```

```
  [ {Repeat ( 2 ) } ↑ } ↑ ]; // IL
```

```
{Repeat ( 2 ) } ↑ } // rest of the code
```

Here's one more situation in your daily life where you could use nested repeat

```
repeat ( for all day in month ) // OL
```

```
  repeat (morning schedule) // IL
```

Can you think of more?

