# Lesson Plan

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

## Lesson 3: Learning Conditions, if then else

**Before you start** – If you haven't already, please read CoderBunnyz Rulebook page 1-8 to be introduced to playing the game.

### Lesson Overview

Students will do a worksheet that teaches them the concept of conditionals. Then they will play the game level with puddles and fences. They will be introduced to the conditional and decision making concepts in coding. Finally, they will write their code as an algorithm.

### Lesson Objective

- First students will do a worksheet to introduce the concept of conditional statement.
- Then they will play a level of CoderBunnyz with puddles and fences, and arrange their code cards as a sequence of steps.
- Finally they will count the number of code cards used as instructions and write/draw the sequence of cards taken. They will also write the conditional statements used in the play. This is called algorithm writing.

### Materials needed

- Condition worksheet (on the next page), game, algorithm sheet 3.1, pencil

### Getting Started

- Instructor will explain worksheet 3. Players will do the exercise.
- After the worksheet is complete, arrange the game (see Rule Book page 10, Level 1.3), explain the cards, movements, and rules. Choose the destination and get ready to start the game.

### Activity

- Play the level 1.3 of CoderBunnyz game to program bunny to move around the fences and jump over the puddles and reach the destination. Continue till all players reach the destination.
- Then each player will review their code cards. That’s the sequence of code they will write on their algorithm sheet.
- Count the number of cards used to reach the destination and write those on the sheet 3.1. Also write the algorithm of the game played.

**Fun Fact**

The language name C because it succeeds another language called B.

# 3. Conditional

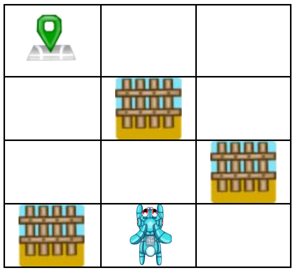
Perform different action(s) based on condition(s).

Read aloud

```
Conditional Operator
 If  (you are hungry) then
     Let’s eat lunch
else
    Eat after an hour
```

Read aloud

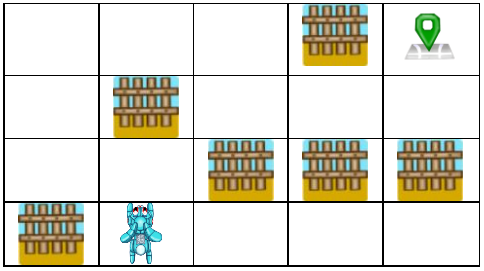
A reason to turn around may be *“if”* there is a fence



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

Read aloud

How many *“if”* would you need to reach 📍 ?

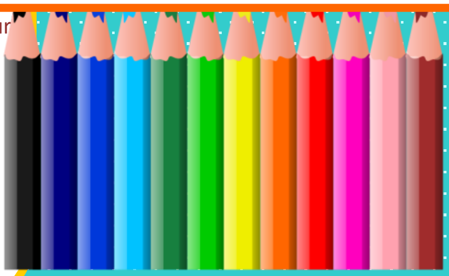


Practice Exercise

Create your own if then else (Hint : think about your 1st and 2nd favorite ice cream flavor)

```
 If  ( the ____________ flavor is available )
    take a scoop
else
     take the _______________  flavor
```

Practice Exercise



Write Algorithm

# 3.1 Conditional

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

_____ MoveForward ( ); ⇧

_____ TurnLeft ( ); ↰

_____ TurnRight ( ); ↱

_____ Jump ( ); ↷

⇧ ↱ ...

MoveForward ( );
TurnRight ( );
..

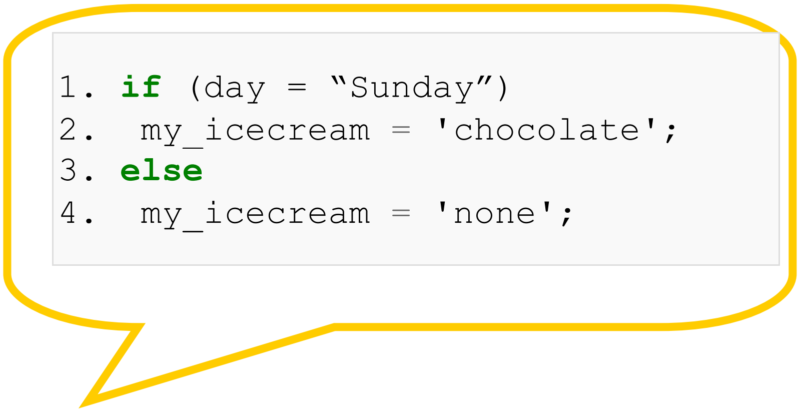
**Using the symbol or text. Write your Code for the game played!!**

**(Bonus - Circle the code movement due to fence, that is conditional 'if')**

## Conditional, “if then else” in real coding languages?

The “if then else” is used by all the coding languages to perform different tasks or action depending on whether a coder specified condition is true or false.

A code example in many programming languages using conditional -



```
1. if (day = “Sunday”)
2.  my_icecream = 'chocolate';
3. else
4.  my_icecream = 'none';
```

If the condition in line 1 is TRUE ( ✓), then line 2 will give the value to “my_icecream”. If the condition in line 1 is FALSE ( ✗), then it will go straight to the else command on line 3 and give “my_icecream” the value of none.

| Day | Condition(Line 1) | my_icecream |
|---|---|---|
| Monday | ✗ (FALSE) | my_icecream=”none” (Line 4) |
| Thursday | ✗ (FALSE) | my_icecream=”none” (Line 4) |
| Sunday | ✓ (TRUE) | my_icecream=”chocolate” (Line 2) |