

# INTRODUCTION TO VISUAL AND INTERACTIVE PROGRAMMING

CT803-4-0-OIVIP

Topic: Control Structure

# Topic Learning Outcomes

At the end of this topic, you should be able to:

- Identify the control structure in visual/block programming.
- Differentiate between IF and IF-ELSE Statement.
- Define the concept of repetition statement
- Differentiate between repeat, forever, repeat until and for repetition statement.
- Implement control structure in visual/block programming.

# Contents & Structure

- Control Structure
- Selection Statement
- Repetition / Looping Statement

# Control Structure

---

Are used to control the flow of program's execution.

---

By default, control structure in a program are executed sequentially.

---

Programs can be much more powerful if we can control the order in which statements are run.

---

The Control Structure let the program decides about what to do next.

# The Control Structure

Control Flow in a program - statements may be executed sequentially, selectively or iteratively.

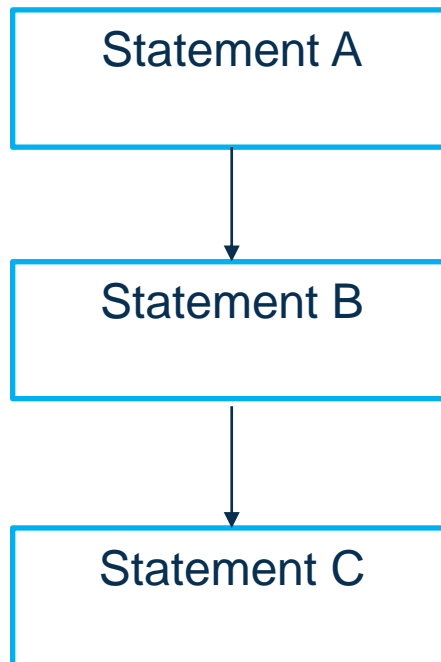
Each programming language provides constructs to support sequence, selection or iteration.

There are three types of programming constructs in visual programming:

- Sequence
  - Functions and Procedures
- Selection
  - If..
  - If.. Else
  - If.. Then.. Else
  - If ... Elself
- Iterative
  - For
  - Repeat
  - Repeat.. Until
  - Forever
  - For.. each

# Sequence Structure

- The sequential structure means the statements are being executed sequentially.
- This represents the default flow of statements.



# Creating Conditions

All conditional structures have a *condition*.

A condition is just an expression that is either true or false.

Test in an IF statement used condition

Comparison operators and logical operators are used in the condition

Example, the condition used in the if structure is  $x < 0$ .

This condition evaluates to either true or false, depending on the value of  $x$ .

If the value of  $x$  is less than zero, then the condition is true. Otherwise, the condition is false.

# Boolean Expression

A Boolean expression is an expression whose value is either True or False.

Examples:

Do you want the coffee? (yes/no)

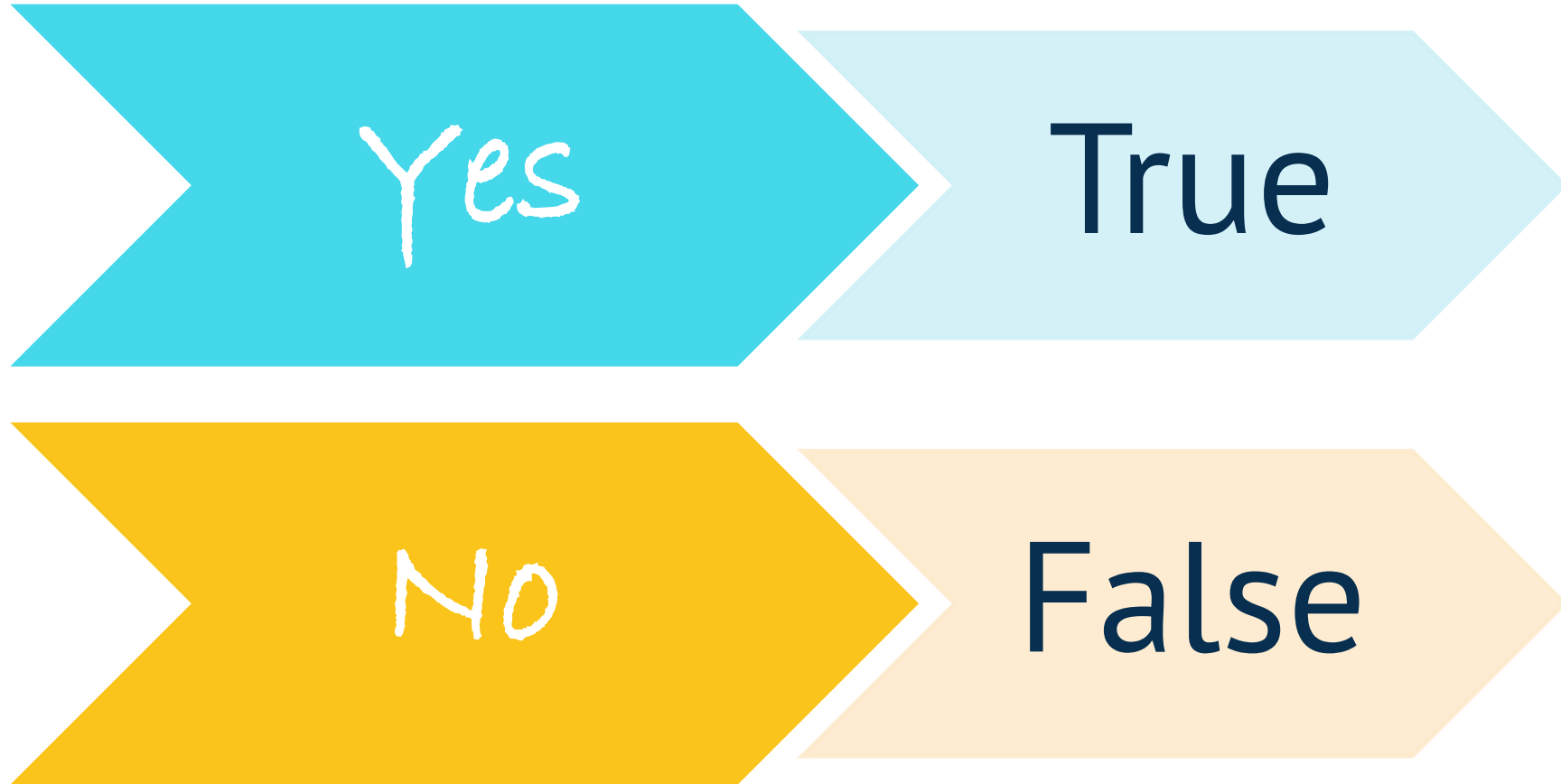
Is x greater than 10? (yes/no)

Have you found the answer? (yes/no)

Does 5 divide 153? (yes/no)



# Boolean Value



# How to get Boolean results

- Comparison Operator

- To get Boolean result, we compare two things

Equal	=
Not Equal	≠
Greater than	>
Greater than or equal	≥
Less than	<
Less than or equal	≤

- Boolean Operator

- To get Boolean results, we combine one or two Boolean results

Boolean Operators	Operators
And	and
Or	or
Not	not

# Selection Structure

- Means the execution of statement(s) depending upon the condition-test.
- If a condition evaluates to true, a course-of-action (a set of statements) is followed otherwise another course-of-action is followed.
- This structure is also called decision structure as it helps in decision making.
- A condition is thus said to "evaluate to true" or "evaluate to false."
- In Snap!, any block whose label says "if," "when," or "until" is a sort of conditional structure.

# Selection Structure – If statement

- If you are 12, raise your hand

```
IF (age = 12)  
    raise your hand  
ENDIF
```

- If you like baseball, stand up

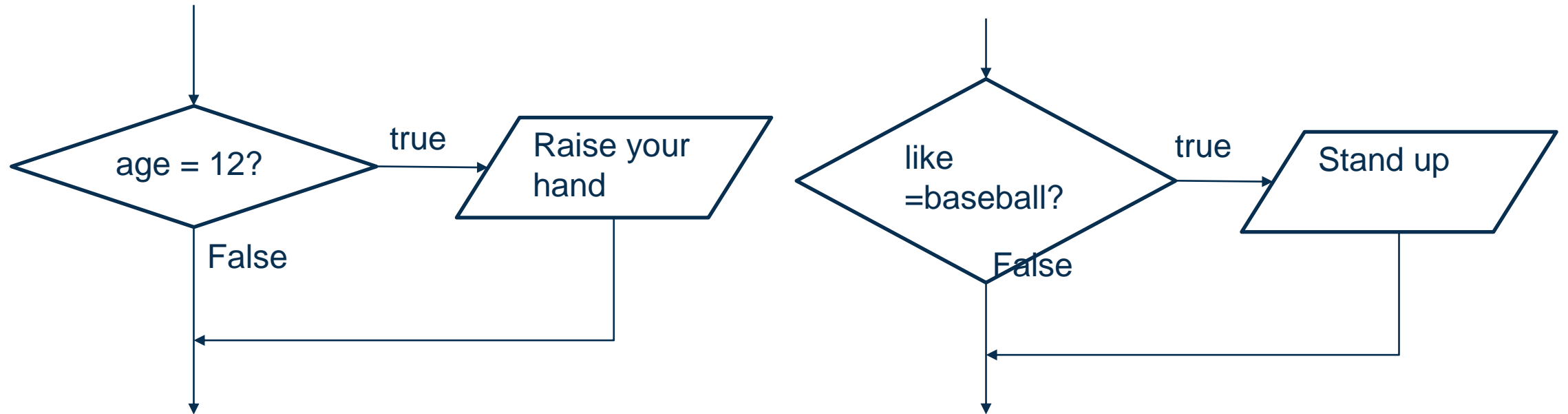
```
IF (like = "baseball")  
    stand up  
ENDIF
```



Pseudocode

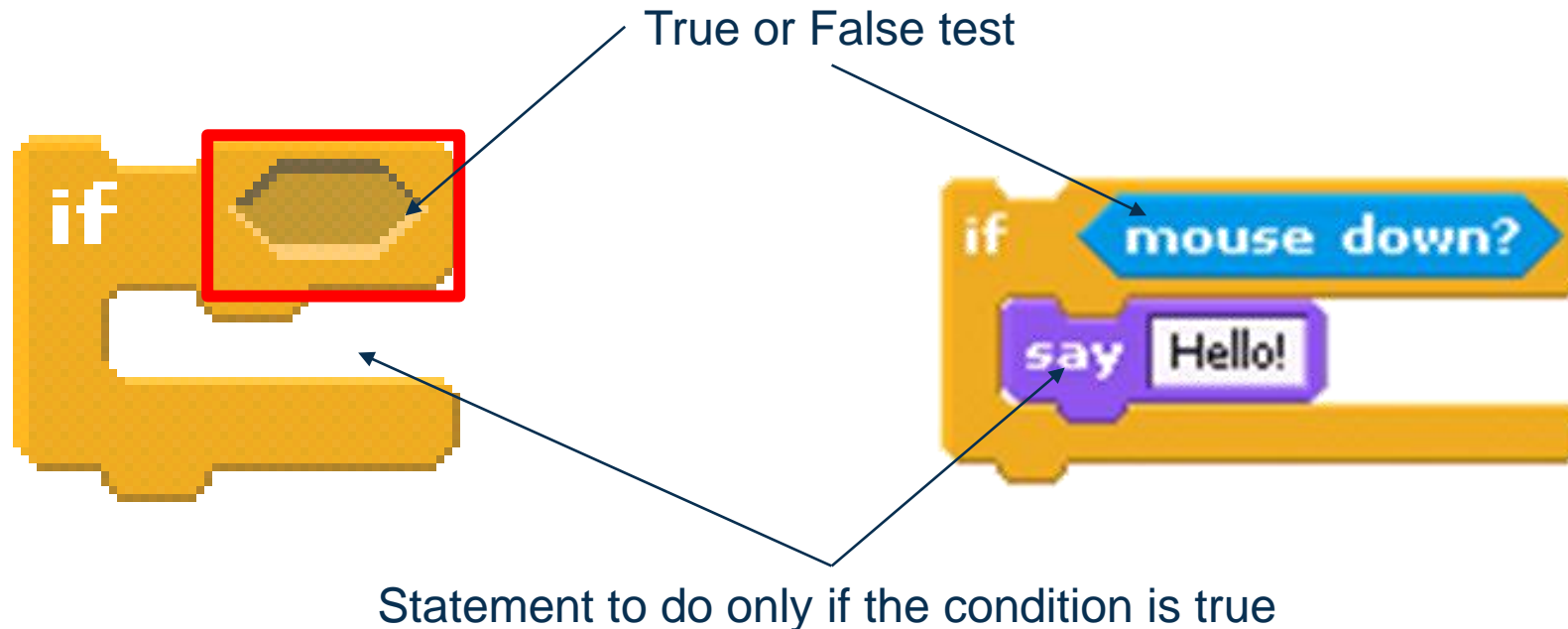
# Selection Structure – IF Statement

- Flowchart



# Selection Structure – IF Statement

- Always evaluates the condition at top of if structure
- Might execute one or nothing



# IF Statement Practice

- Let's create an If Statement
  - Create a variable A
  - Set A to 0
  - Create an If Statement
    - If  $A > 2$
  - Perform some Motion
  - Change A to 3

# Selection Structure – IF.. ELSE Statement

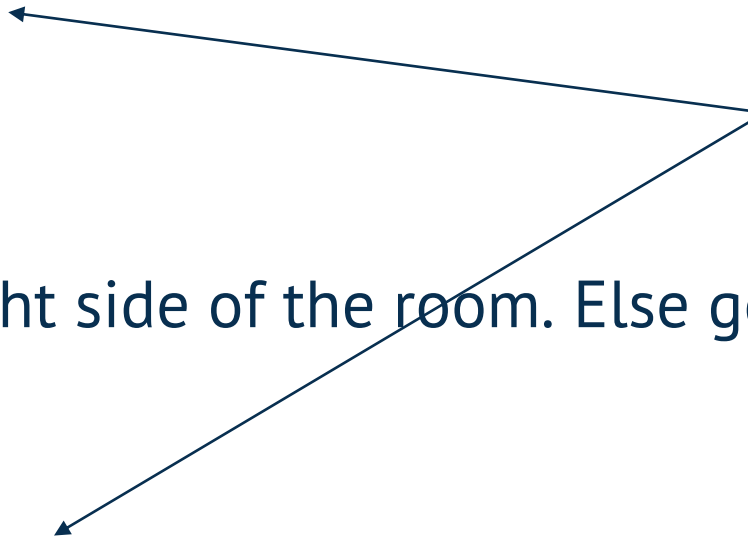
- If you like football, raise your hand. Else stand up

```
IF (like = "football")
    raise your hand
ELSE
    stand up
ENDIF
```

Pseudocode

- If you are 11, go to the right side of the room. Else go to the left side of the room

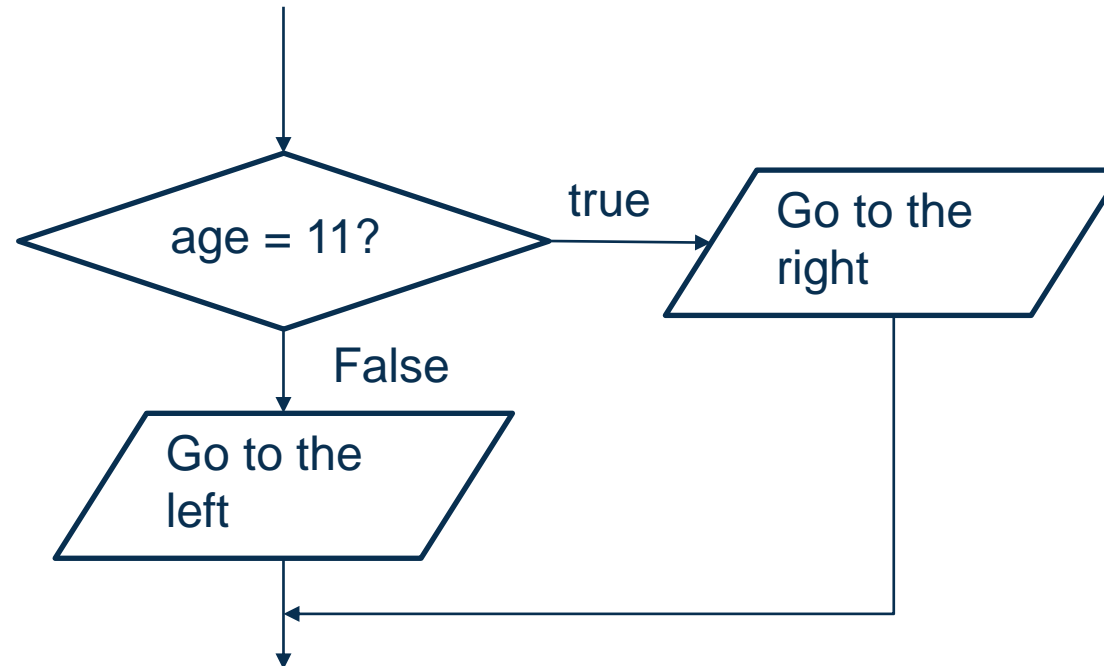
```
IF (age =11)
    go to the right
ELSE
    go to the left
ENDIF
```





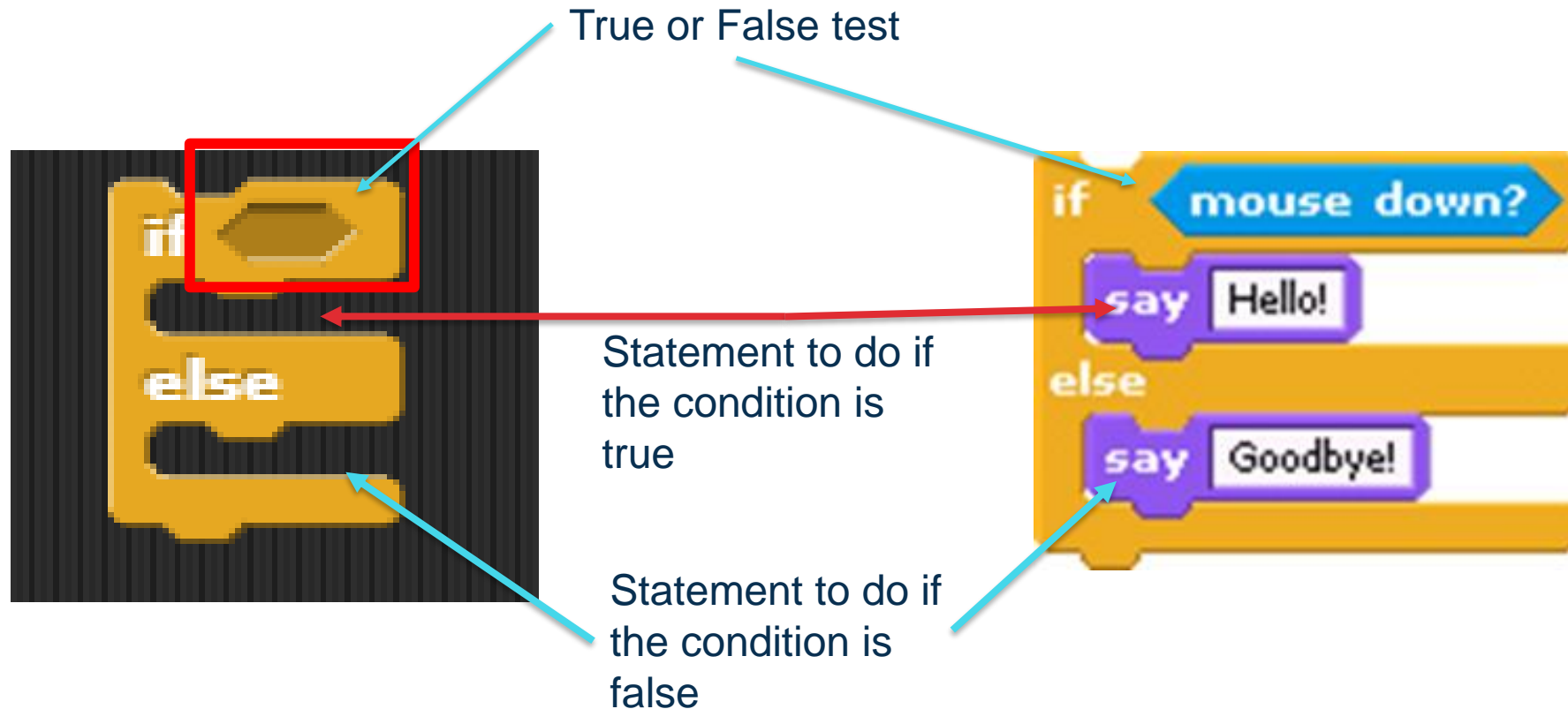
# Selection Structure – IF.. Else Statement

- Flowchart



# If-else structure: conditional

- Always evaluates condition at top of the if-else structure
- Boolean expression: true or false



# IF.. Else Practice

- Loop forever
- If the mouse is clicked
  - Move 10 steps
- Else
  - Turn 15 degrees

# Selection Statement – When and wait until

- **When Construct:** Instruct a sprite to do something when an event occurs:



- **Wait until Construct:** Instruct a sprite to wait to do something until a condition is true:



# Quick Test

- The price of the ice cream is 100 coins. Analyze the scripts presented below and select the correct statement.

A

```
if icecream < 100  
  say I'll buy an ice cream for 2 secs
```

B

```
if icecream > 100  
  say I can't buy ice cream for 2 secs  
else  
  say I'll buy an ice cream for 2 secs
```

# Repetition Structure

- Loops enable blocks of code to be executed more than once
- Loops represent a simple concept that can be used to create extremely dynamic and flexible programs
- Loops are very important for two reasons:
  - Efficiency
  - flexibility

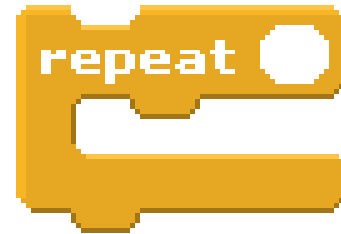
# Repetition Structure

- In programming, a **loop** can induce multiple executions of statements.
- In Snap!, any block whose label begins with "forever" or "repeat" is a looping construct.



# Repetition Structure - Repeat

- The simplest loop is the repeat block.
- It closely resembles the for loop found in other programming languages
- The repeat block expects the programmer to specify a number which will indicate exactly how many times it will repeat



- This block is used when the number of times to repeat is known before the block is encountered, but sometimes, there is no way of knowing exactly how many times a loop must execute.
- In cases of uncertainty, a slightly different loop must be used



# Repetition Structure – Repeat.. Until



- The repeat until loop, as the name implies, repeats until a given Boolean expression is true
- It is very similar to the while loop found in other programming languages; the only difference is the Boolean expression is switched.
- Instead of repeating until a value is true, a while loop repeats until a value is false
- Repeat until loops are perfect when combined with user input
- This allows the user to interact with the program indefinitely; loops such as these are found in almost every program in some way, shape, or form
- Although the two loops mentioned thus far appear to behave differently, they are the same loop. With the use of variables, the repeat until loop can be made to work exactly like a repeat loop

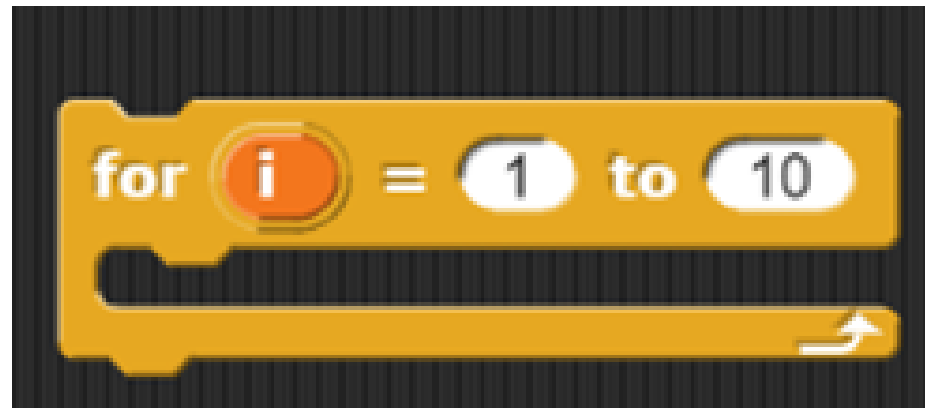
# Repetition Structure - Forever

- In some cases, a loop must continue repeating the entire time the program is running.
- When this is required a forever loop is often used
- Notice that there is no “connector” on the bottom of this block.
- This is because any blocks located below it in the script would be impossible to reach
- Like the repeat until loop, the forever loop is commonly used in conjunction with user input.



# Repetition Structure – For Loop

- Run the script repeatedly
- Keeping count just like repeat loops
- This block is used when the number of times to repeat is known before the block is encountered.



# Exercise Question

1. The expression in the predicate part of an if/else structure evaluates to what kind of value?
2. If a certain block of number of blocks needed to be executed several times, what control structure would you use?
3. If a loop executes 5 times and there was a block inside that moved the sprite 3 steps, how many steps would it move in total?

# Summary / Recap of Main Points

- Control Structure
- Selection Statement
- Repetition / Looping Statement