

CISC 1003 - EXPLORING ROBOTICS



FLOWCHARTS

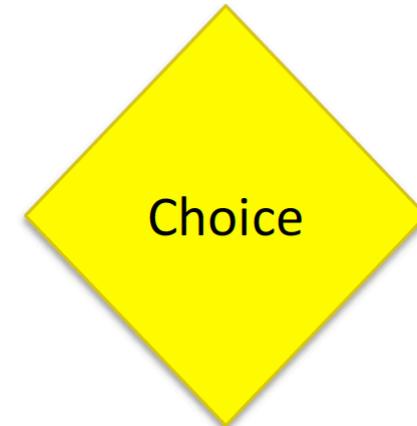
Flowcharts have a standard set of shapes and colors that are universally used so that everyone can understand what they mean.



The start and the stop shapes show where the program starts and stops



These rectangular blocks represent actions in the program

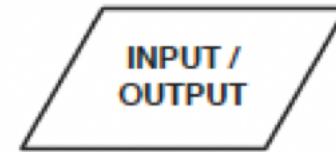


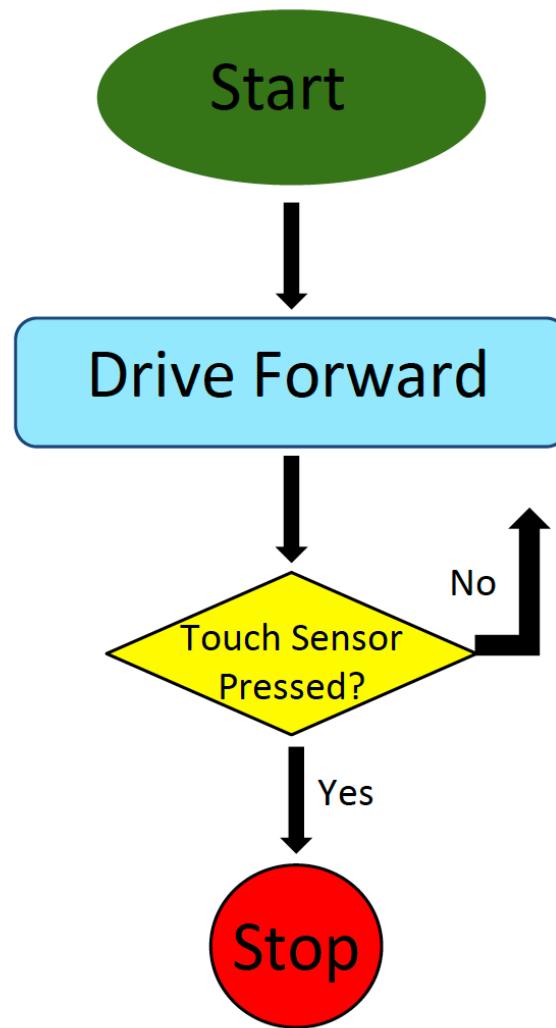
Yellow diamonds represent a choice or decision based on a question. This must be at least a yes/no decision.

Flowcharting

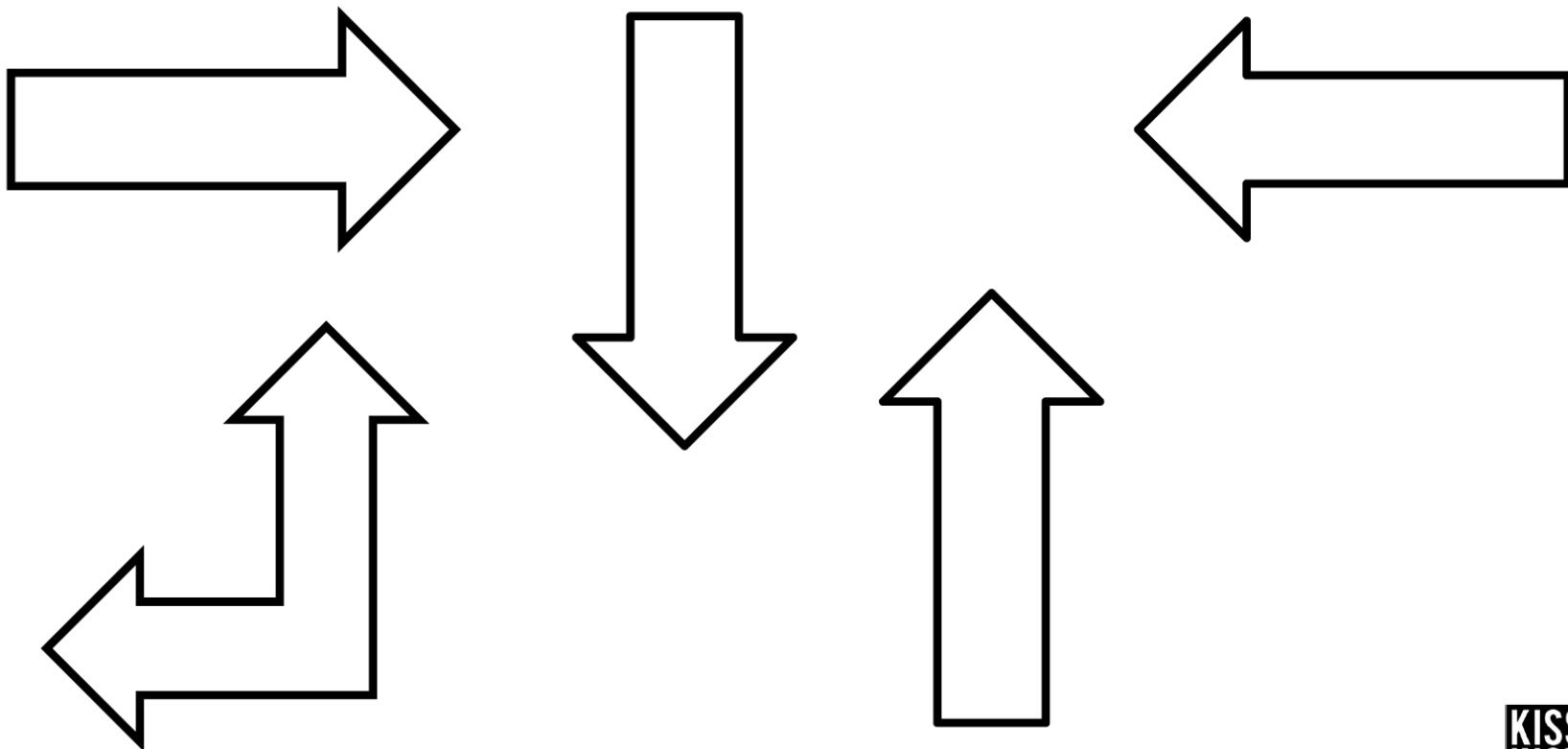
- Input and output:

- The input and output symbol contains data that is used for input (e.g., provided by the user)
 - and data that is the result of processing (output)

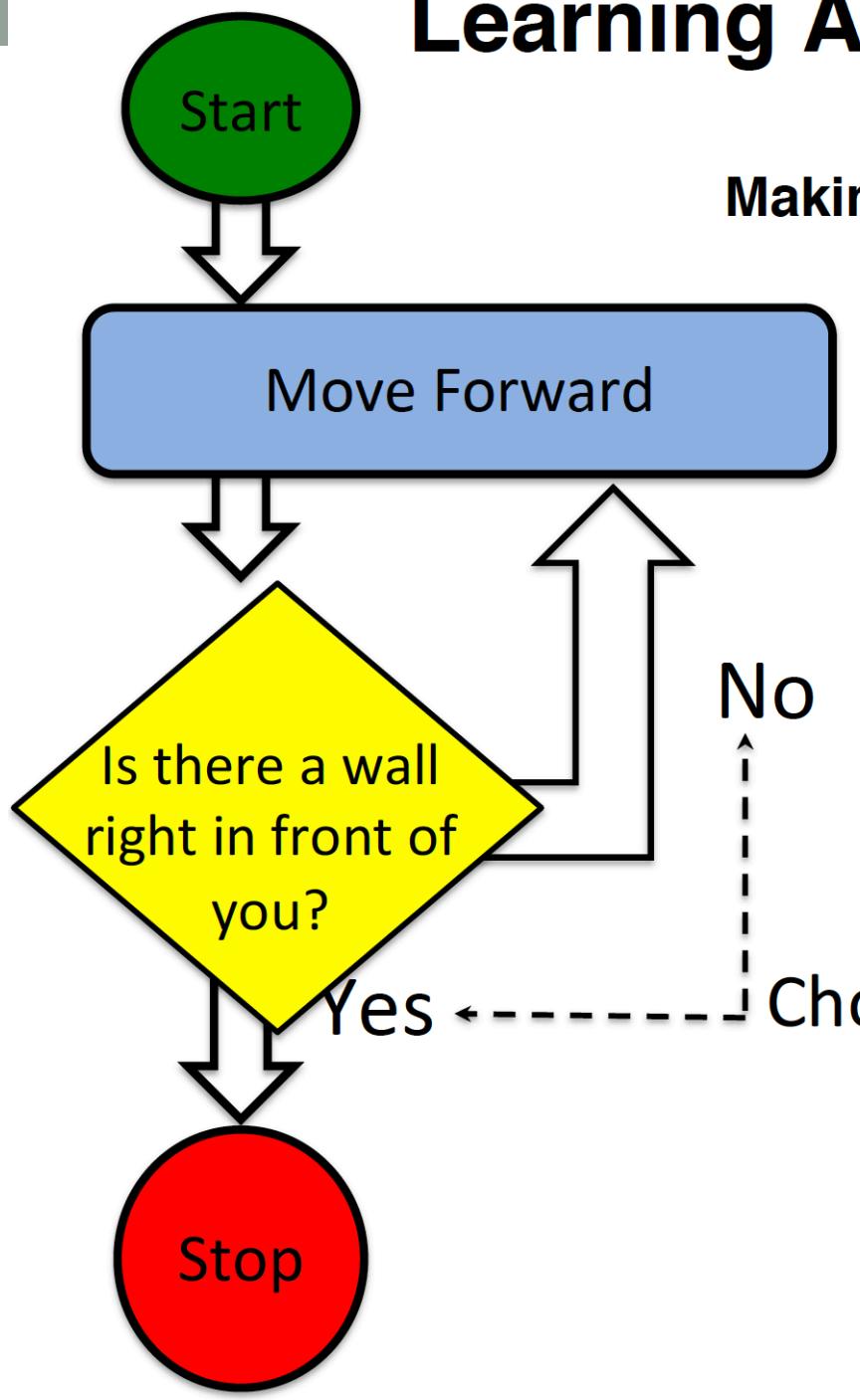




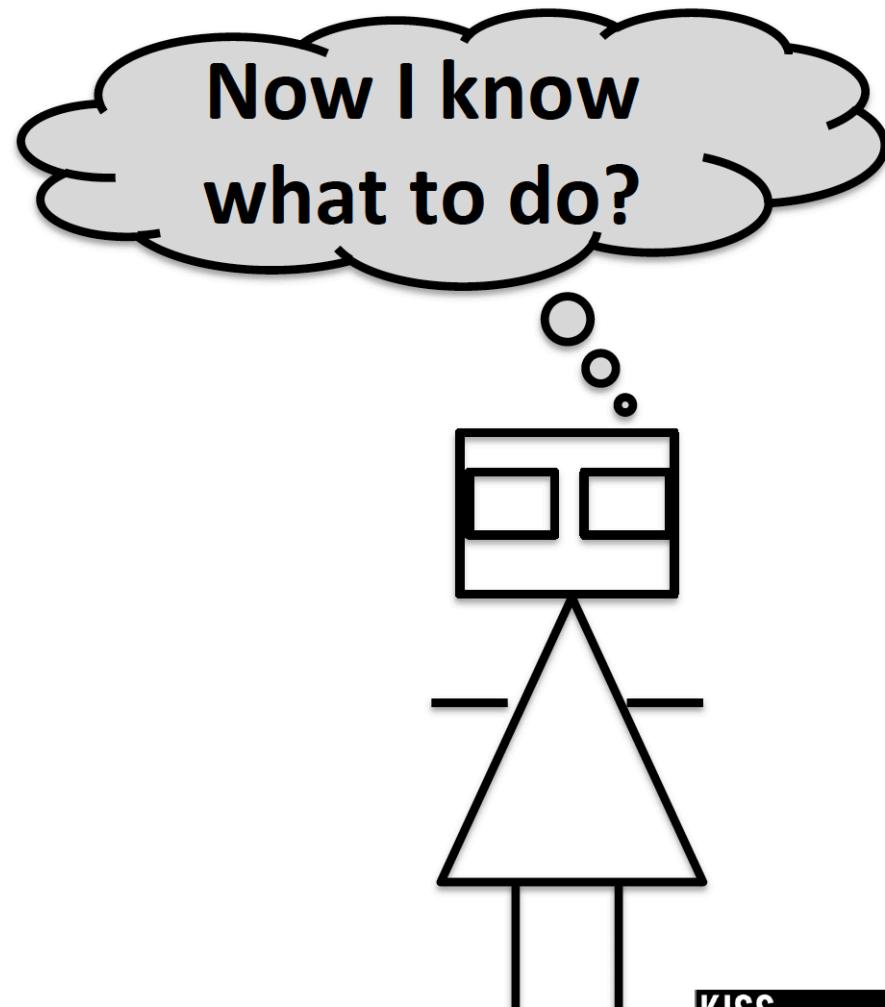
Arrows are used in flowcharts to show the direction, or flow, of the program.



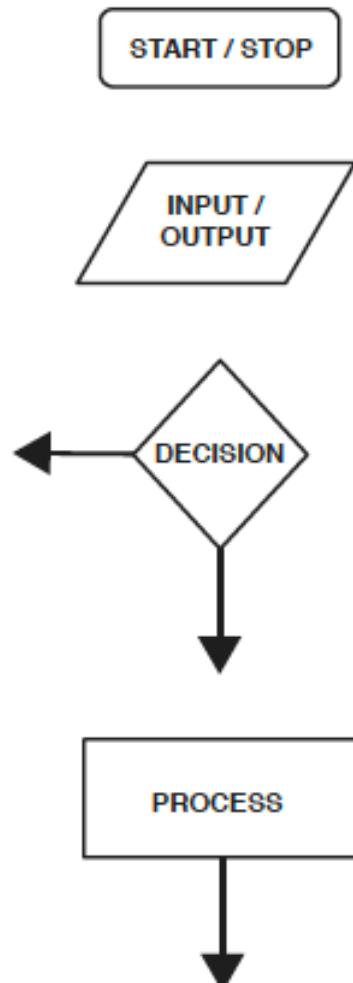
Learning About Flowcharts



Making a Decision:



Common Flowchart Symbols



Program Variables

Do Something

These rectangular
blocks represent actions
in the program

VARIABLES IN FLOWCHARTS

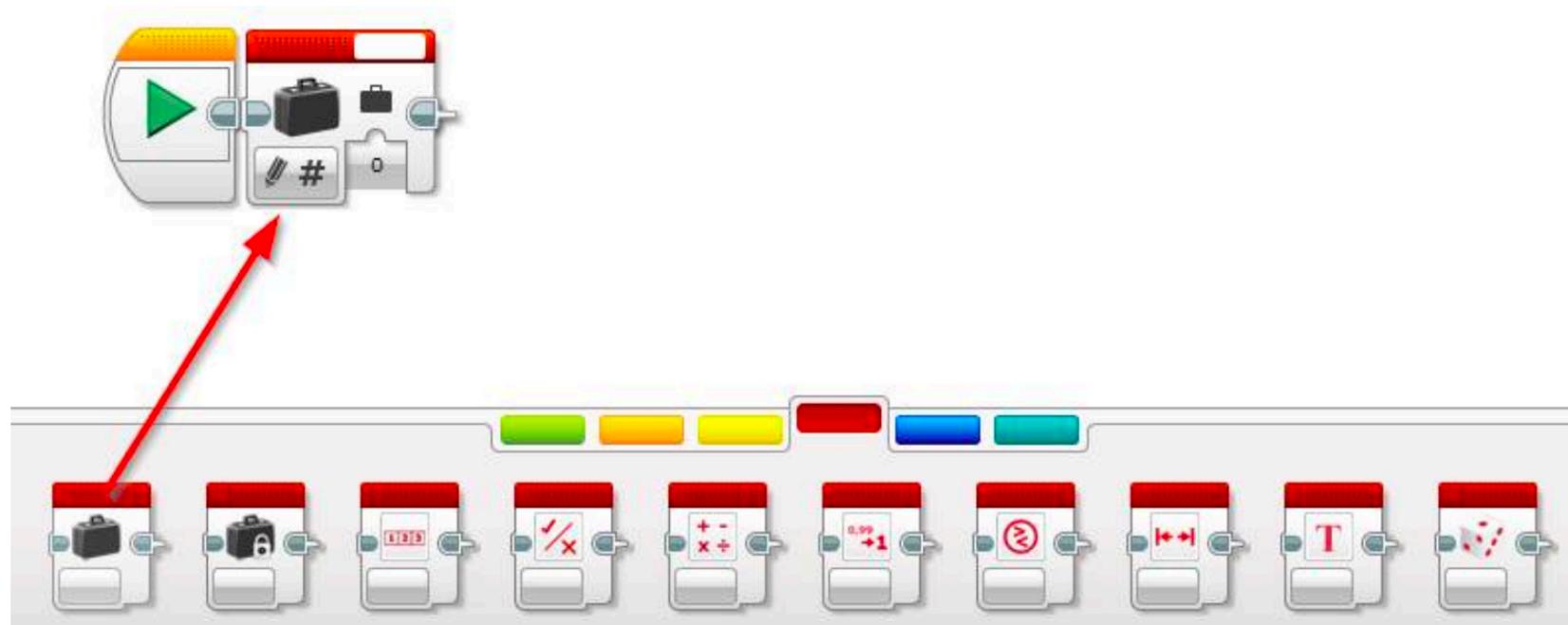
Program Variables

- A variable is a storage location for data which we give a name within a program
- The name provides a way of labelling the data
 - so that we can better understand its purpose within the program's logic
- Variables are typically reused in the program

Variables in EV3

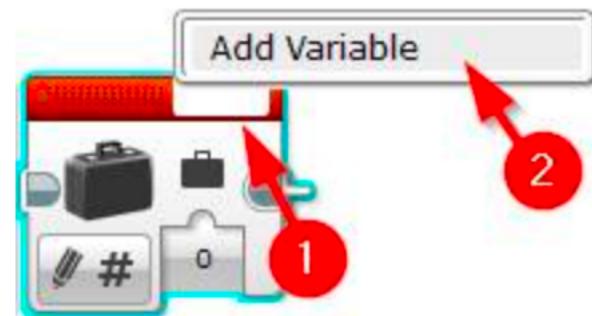
- Using a variable is quite simple as there are only three things we can do with a variable:
 - Define the variable
 - Write data to the variable
 - Read data from the variable

Declaring a Variable in EV3



Declaring a Variable in EV3

- Create a new program
- Drag and drop the **red Variable block** next to the Start block (as per the image above).
 - Update its **mode** to **Write | Numeric**
- Click the **Variable selector box** in the top right and click **Add Variable**



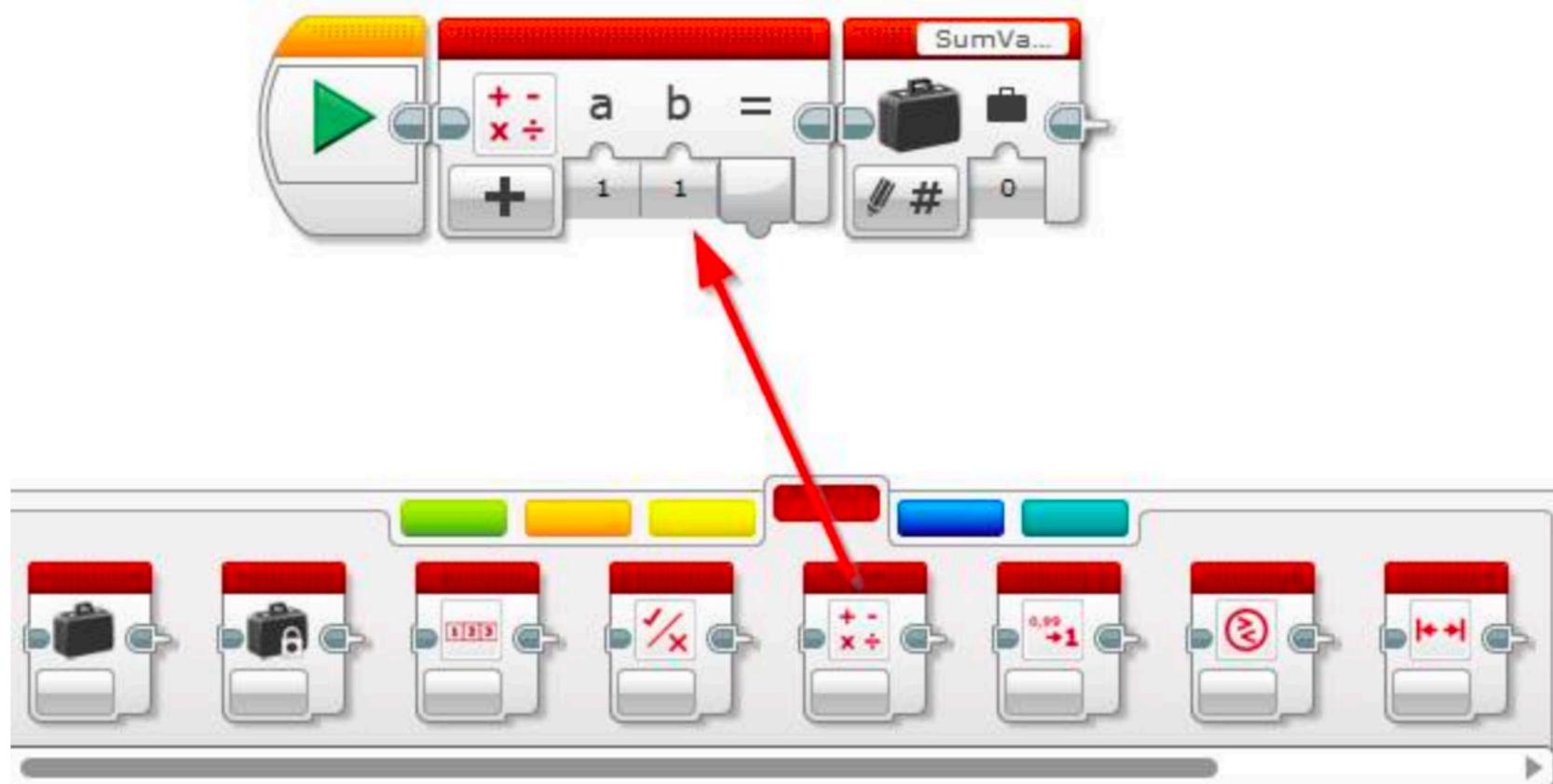
Declaring a Variable in EV3

- Give the variable a descriptive name:
 - Within the New Variable window **type SumValue** and **click Ok**



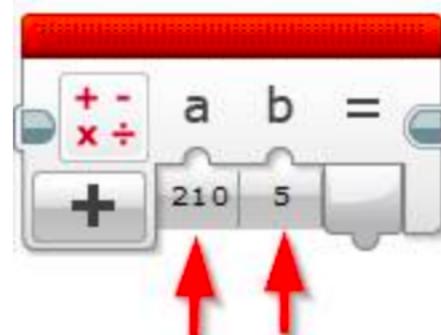
Write Data to the Variable

- Insert a **Math block** in between the start block and the variable block



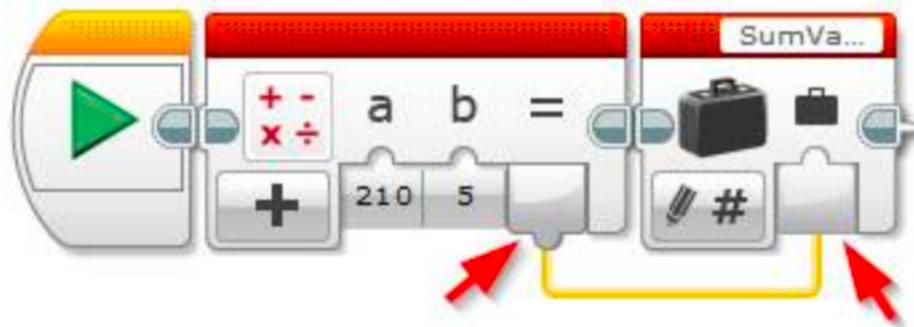
Write Data to the Variable

- Update both **a** and **b** inputs to any number of your choosing.
 - For example: calculate $210 + 5$:



Write Data to the Variable

- Set the variable by wiring the result of the Math block into the Variable block:
 - Ensure the **Variable block**'s mode is **Write | Numeric**
- **Drag the equals sign (=)** from the **Math block** over to the **input** on the **Variable block**:

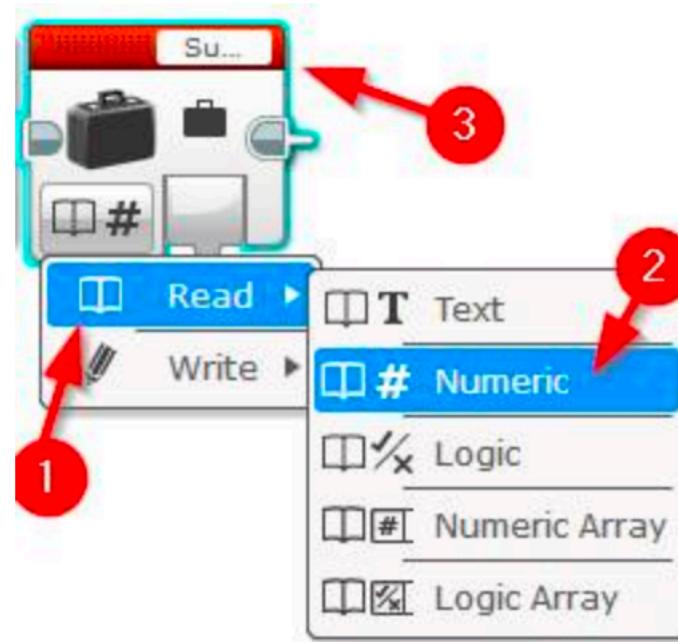


Writing Data to a Variable EV3

- Your data is now saved in a variable SumValue
- You can now use this variable to read back the data as needed

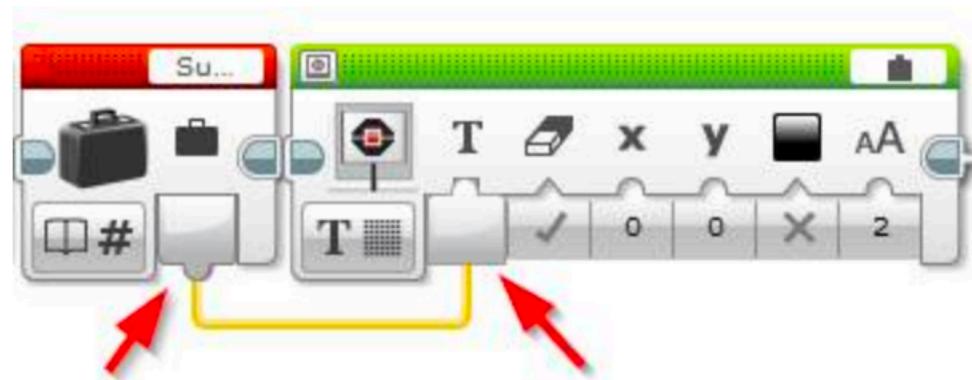
Reading Data from a Variable in EV3

- Drag and drop a red Variable block at the end of the program:
 - Set its **mode** to **Read | Numeric**
 - Select the “**SumValue**” variable from the **variable name list**



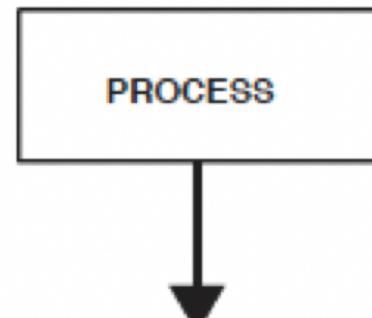
Display the Variable on Screen

- Drag the Value input from the Read Variable block into the Text input of the Display block



Flowcharts and Variables

- Process:
 - The process symbol contains brief descriptions (a few words) of a rule or some action taking place
 - Represents an action that changes value or location of data
 - For example, assign a value to a variable

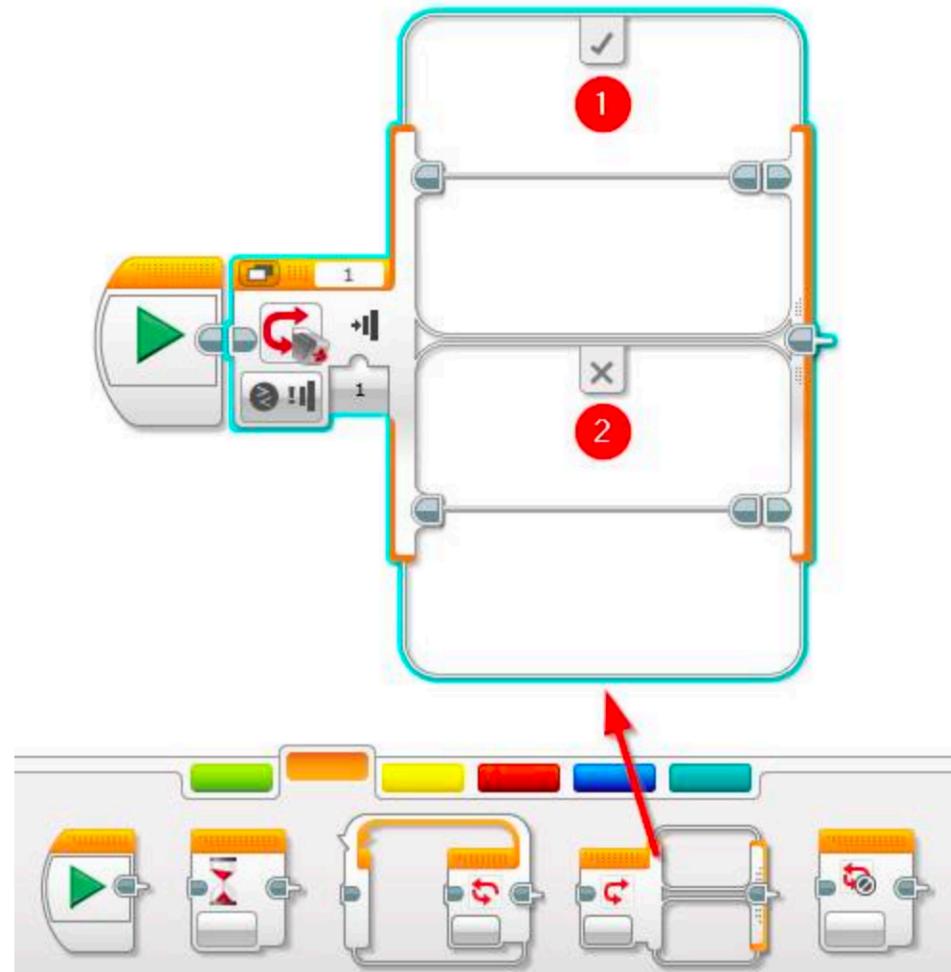


CONDITIONAL STATEMENTS

Conditional Statements

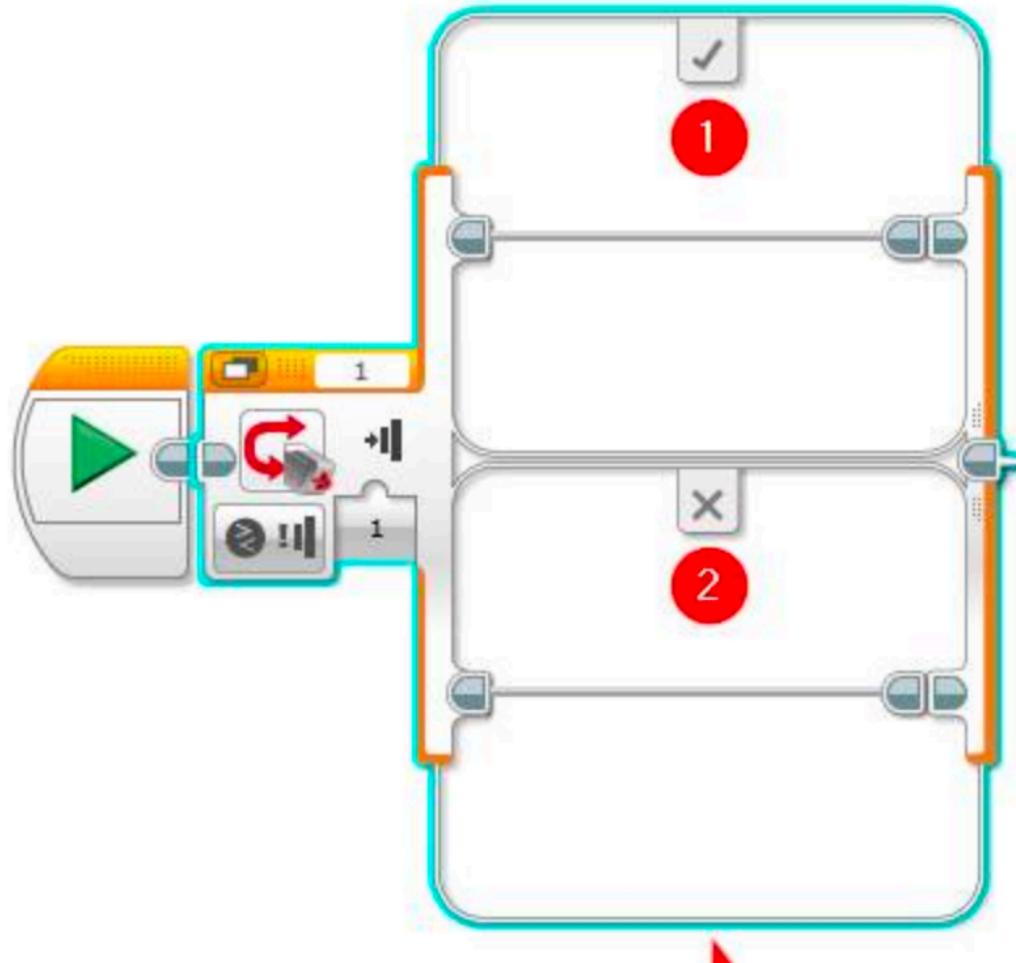
- In many cases actions are conditional
 - Program decides what to do based on the answer to some question or on some condition that is observed
- In EV3 – implemented by a switch block

Switch Block in EV3



Switch block in EV3

- True = 1
- False = 2

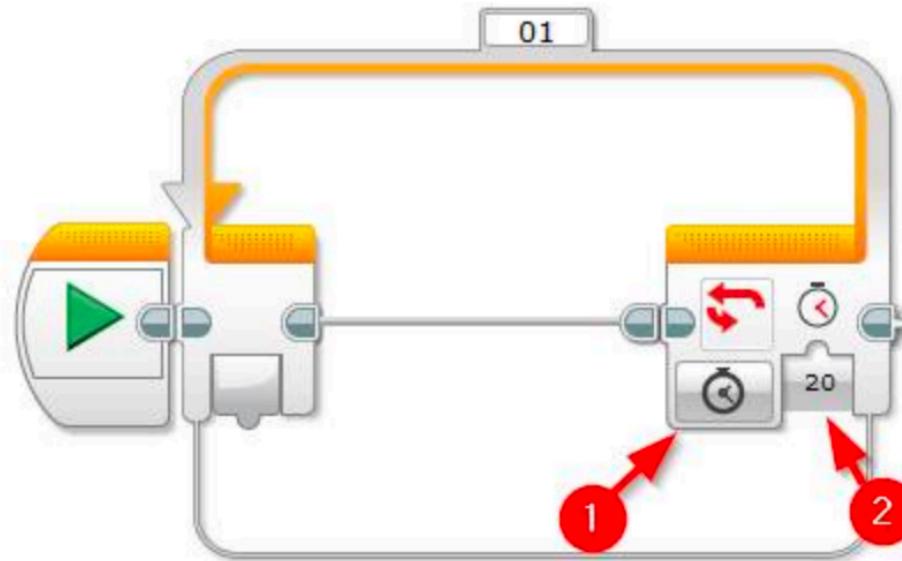


Switch block in EV3

- Example:
 - Use the touch sensor
 - when the touch sensor is pressed the robot will move forward
 - when it's not pressed the robot will stop.
 - Loop this logic for 20 seconds and then stop the program

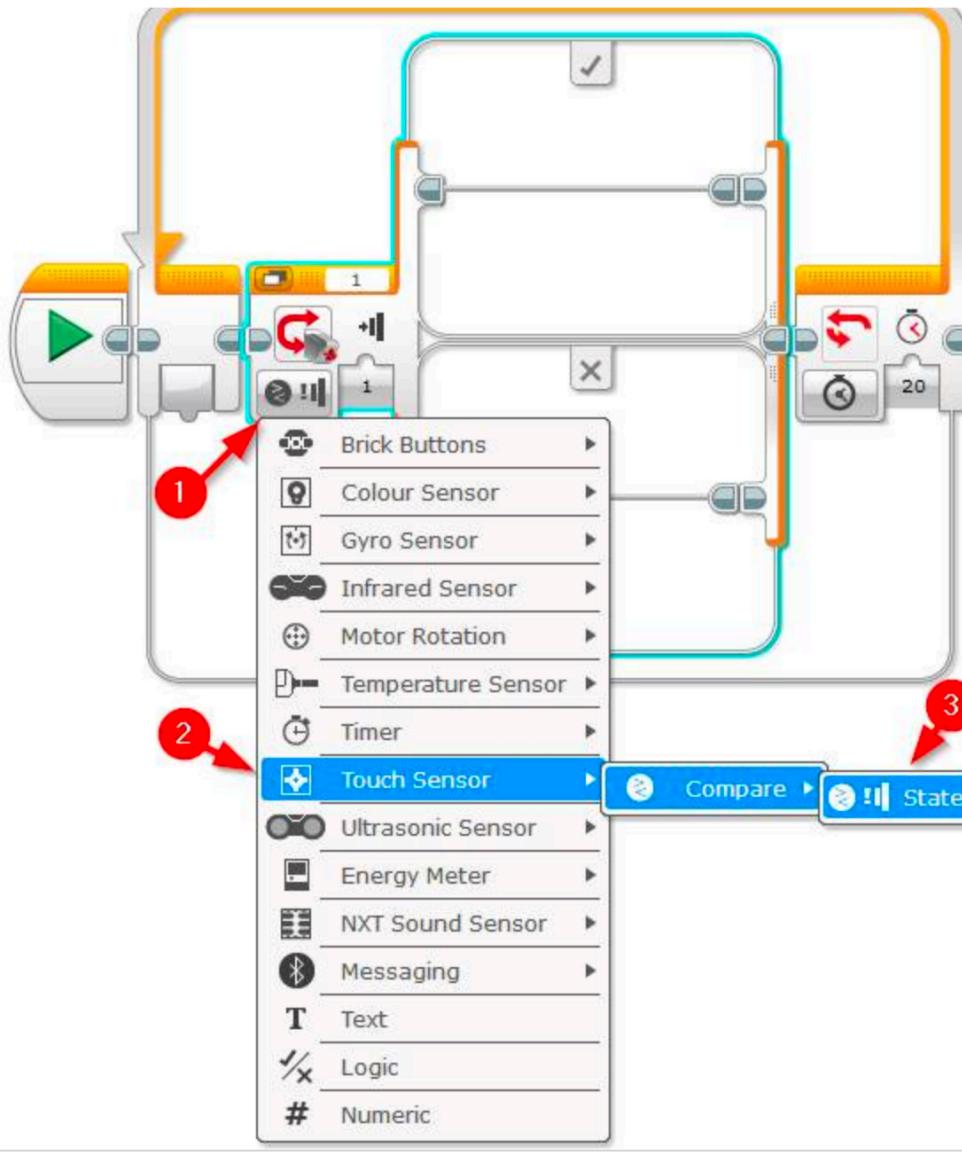
Switch block in EV3

- Loop logic:
 - run the program for 20 seconds



Switch block in EV3

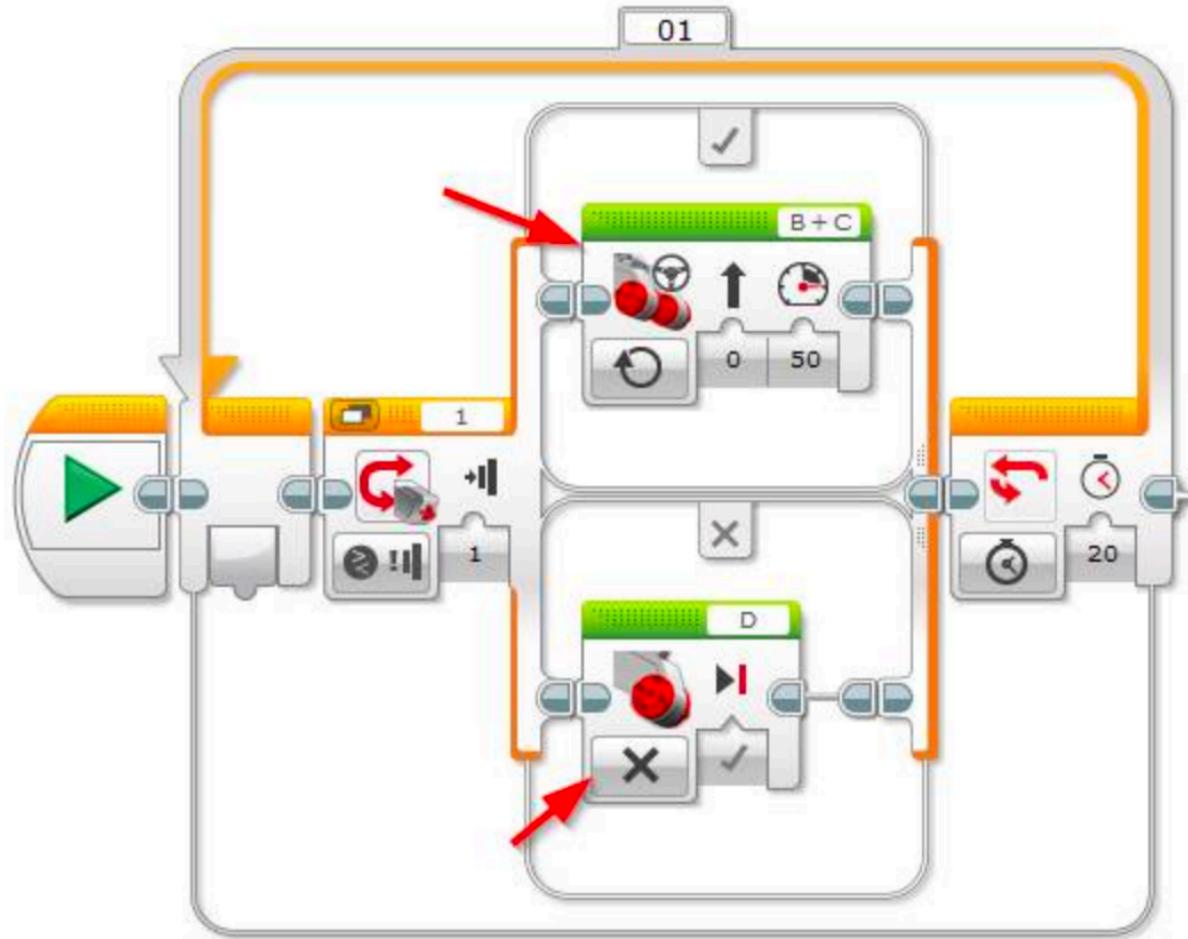
- Next steps:
- Drag and drop an orange **Switch block inside the loop**
- Set mode to **Touch Sensor | Compare | State**



Switch block in EV3

- Move the robot forwards if the button is pressed and stop it once the button is off:
- Move forwards:
 - Drag and drop a green **Move Steering block** into the True (tick) case of the Switch block
 - Set its **mode** to **On**
- Stop the robot
 - Drag and drop a green **Move Steering block** into the false (cross) case of the Switch block
 - Set its **mode** to **Stop**

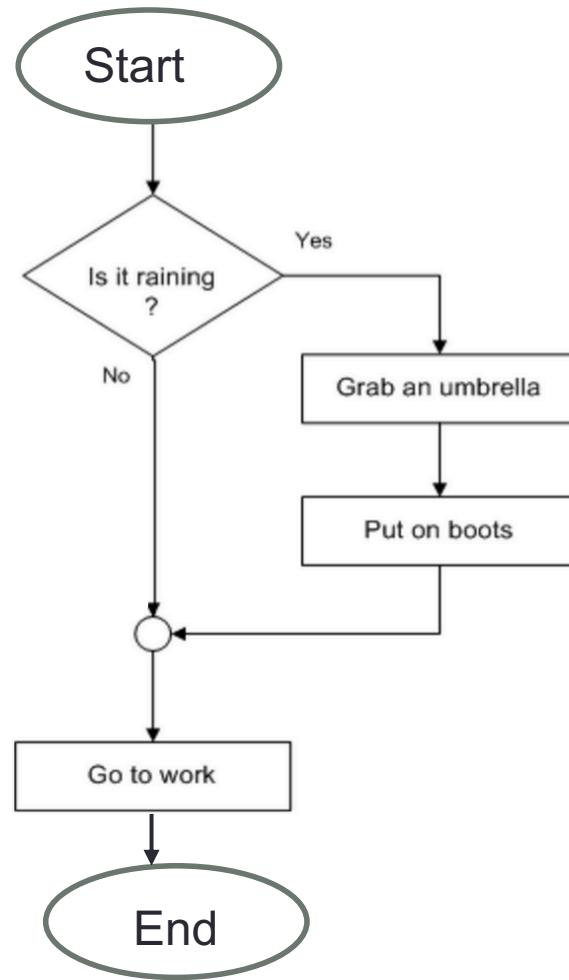
Switch block in EV3



Conditional Statements

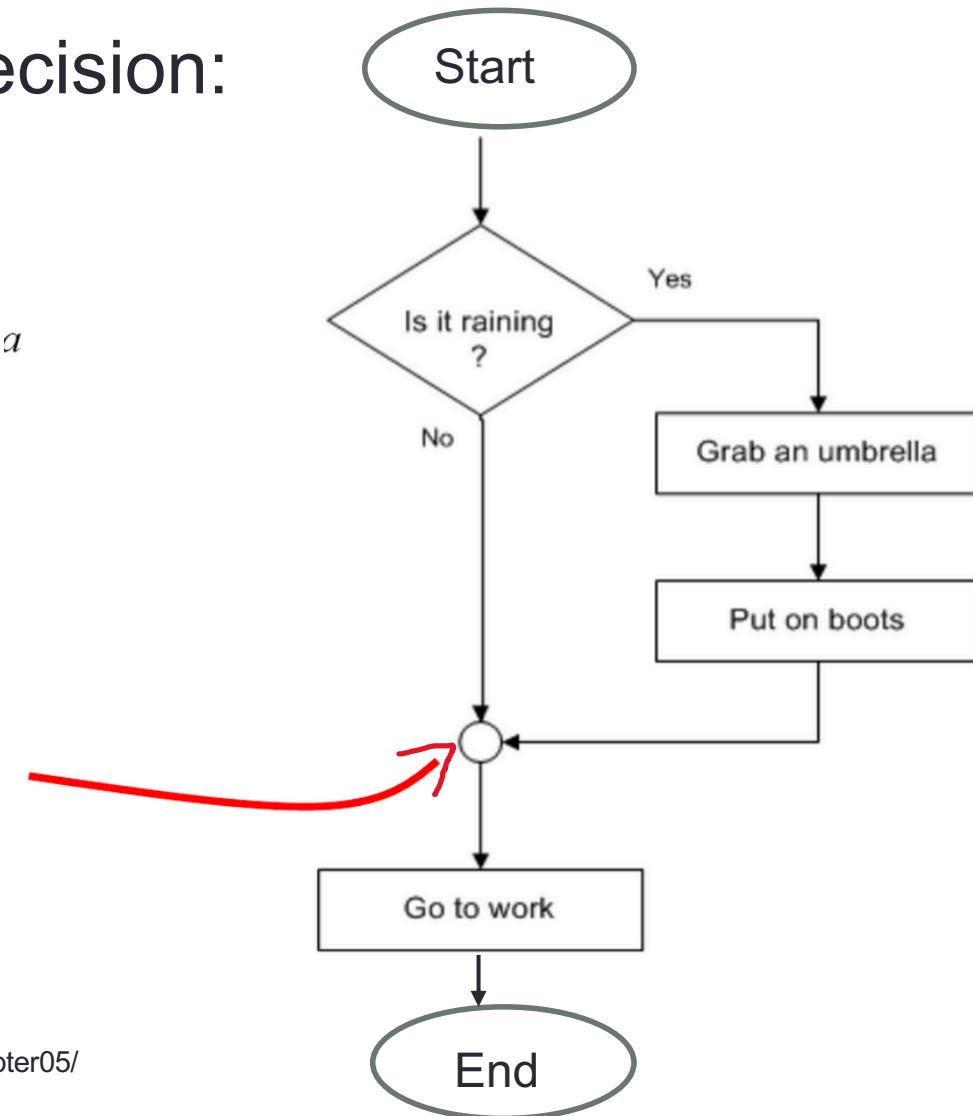
- Design a program: what should I wear today?
- Algorithm for such a program:
 - If it is raining
 - Grab an umbrella
 - Put on boots
 - Go to work
- How would the flowchart look like?

Conditional Statement Flowchart



Conditional Statement Flowchart

- Paths rejoin after decision:



Conditional Statements

- Alternative pseudocode:
 - If it is raining
 - Grab an umbrella
 - Put on boots
 - Otherwise
 - Wear sunglasses
 - Put on sneakers
 - Go to work
- How would the flowchart look like?

Boolean Expressions

- A type that has just two possible values:
 - Typically True and False

Meaning	Mathematical symbol
Less than	<
Greater than	>
Less than or equal to	\leq
Greater than or equal to	\geq
Not equal to	\neq
Equal to	$=$

Pseudocode for Conditional Statement

- Ask the user for his score
- If score bigger than or equal to 65
 - Print “you are passing”
- Otherwise
 - Print score “score is below 65, better study some more”
- Print “bye”
- What would a pseudocode look like?

Pseudocode and Exercise

- Ask the user for his score
- If score bigger than or equal to 65
 - Print “you are passing”
- Otherwise
 - Print score “score is below 65, better study some more”
- Print “bye”
- What would a pseudocode look like?
- What is we want to print the actual score in the second option?

Pseudocode Example

- Ask user for his score
- Create a variable called “scoreVar”
 - assign it the user input
- If score bigger than or equal to 65
 - Print “you are passing”
- Otherwise
 - Print the value of scoreVar
 - Print “ is below 65, better study some more”
- Print “bye”

Pseudocode Example

```
score = int(input("Enter your score:"))
if score >= 65:
    print ("looks like you are passing.")
else:
    print (score, "is below 65.")
    print ("Better study some more.")
print ("Bye!")
```

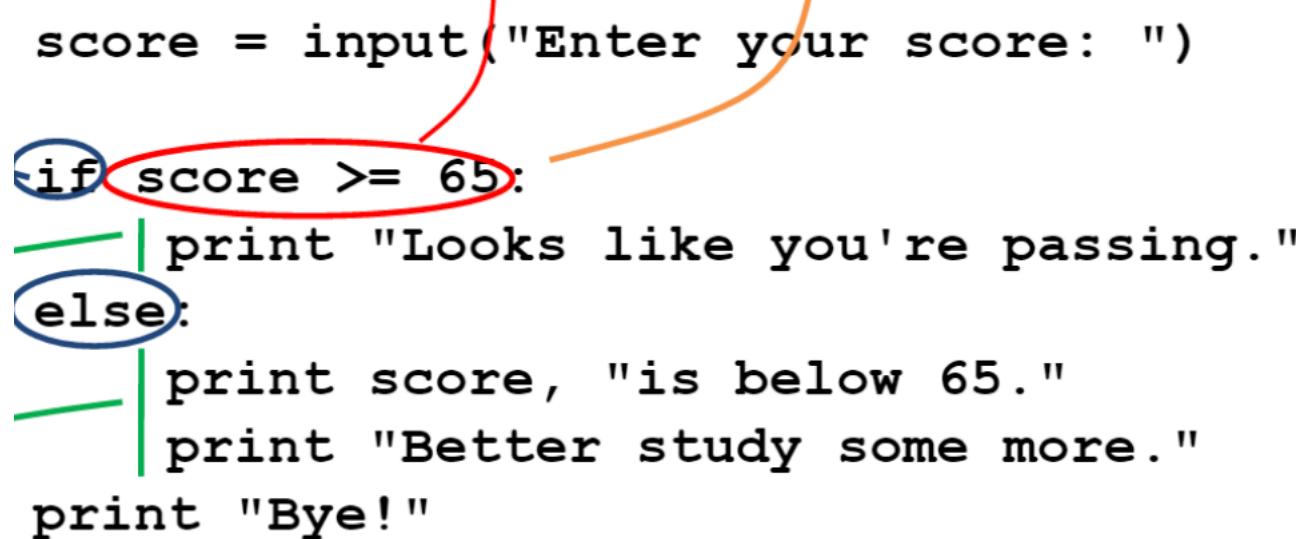
Pseudocode Example

- Boolean condition:

```
keywords    score = input("Enter your score: ")
if score >= 65:
    print "Looks like you're passing."
else:
    print score, "is below 65."
    print "Better study some more."
print "Bye!"
```

Pseudocode Example

```
score = input("Enter your score: ")  
  
if score >= 65:  
    print "Looks like you're passing."  
else:  
    print score, "is below 65."  
    print "Better study some more."  
print "Bye!"
```



The diagram illustrates the control flow of the pseudocode. A red oval highlights the condition `score >= 65`. A blue oval highlights the `if` keyword. An orange curved arrow points from the `if` keyword to the start of the `else` block. A green vertical line with a bracket indicates the scope of the `if` block, covering both the `if` statement and the `print` statement under it. Another green vertical line with a bracket indicates the scope of the `else` block, covering the `print` statement under it.

Pseudocode and Flowchart Example

- Find the grade of a student by reading marks or by taking Percentage.
- Here we are assuming:
 - greater than 80 percentage or marks as grade A
 - if the Marks are between 80-60 Grade is B,
 - if the marks are between 60-40 grade is C,
 - If student got marks below 40 it means he is Failed.
- What would the algorithm be?
 - What would be the pseudocode and flowchart for this program?

Find Max Number

- Write a pseudocode and a flowchart to achieve the following algorithm:
 - Get Inputs firstNum1, firstNum2 from user
 - Find the maximum of the two numbers
 - Output the maximum number

Find Max Number

- Write a pseudocode and a flowchart to achieve the following algorithm:
 - Get Inputs firstNum1, firstNum2 from user
 - Find the maximum of the two numbers
 - Output the maximum number
- Other details:
 - Create a variable called 'larger' and use it to store the max number

FLOWCHART LOOPS

Loops

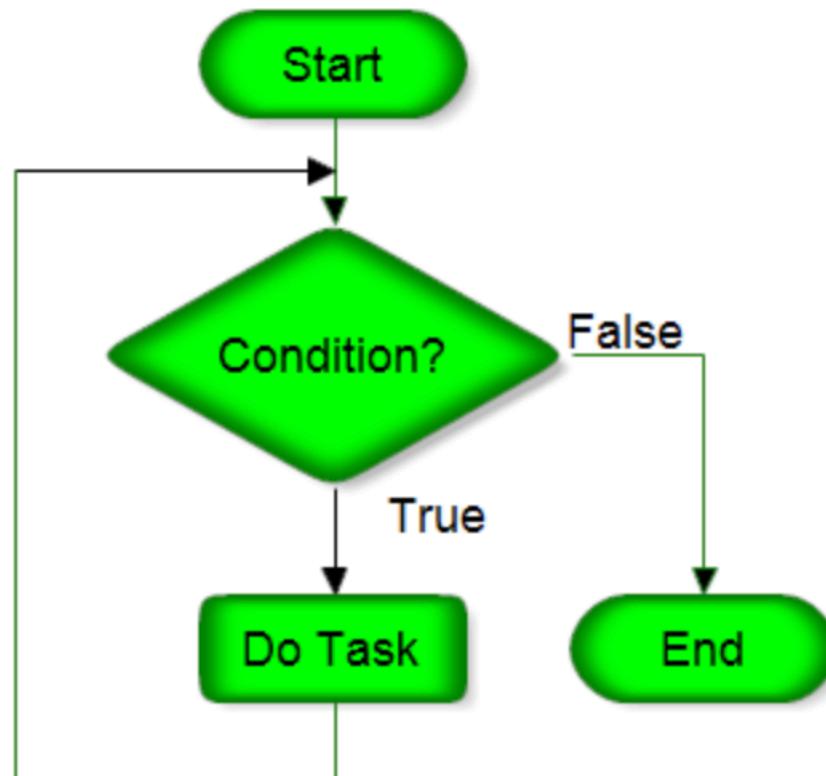
- Loop is an instruction that repeats until a specified condition is reached.
- In a loop structure, the loop asks a question.
 - If the answer requires action, it is executed.
 - The same question is asked again and again until no further action is required.
- Each time the question is asked is called an iteration.

Types of loops

- A **for** loop is a loop that runs for a preset number of times.
- A **while** loop is a loop that is repeated as long as an expression is true. An expression is a statement that has a value.
- A **do while** loop or **repeat until** loop repeats until an expression becomes false.

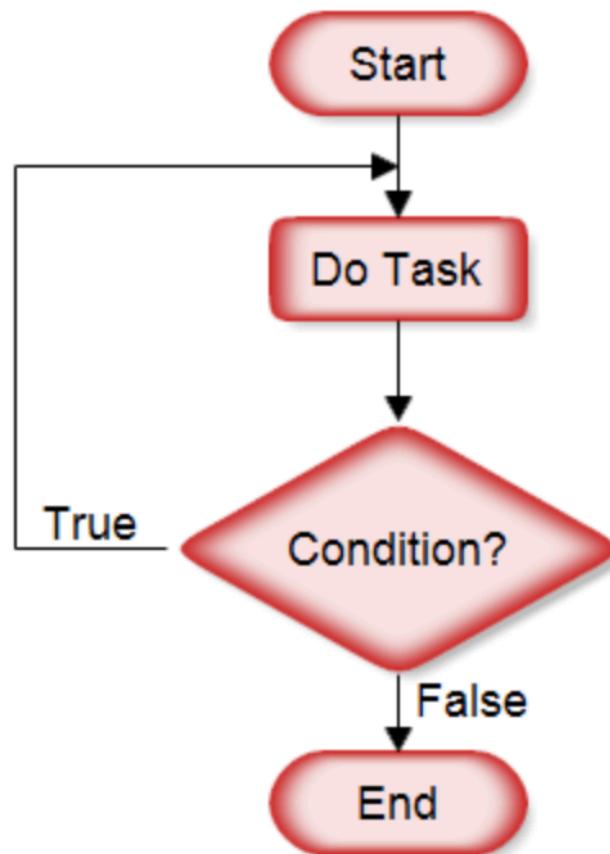
WHILE LOOP

While Loop

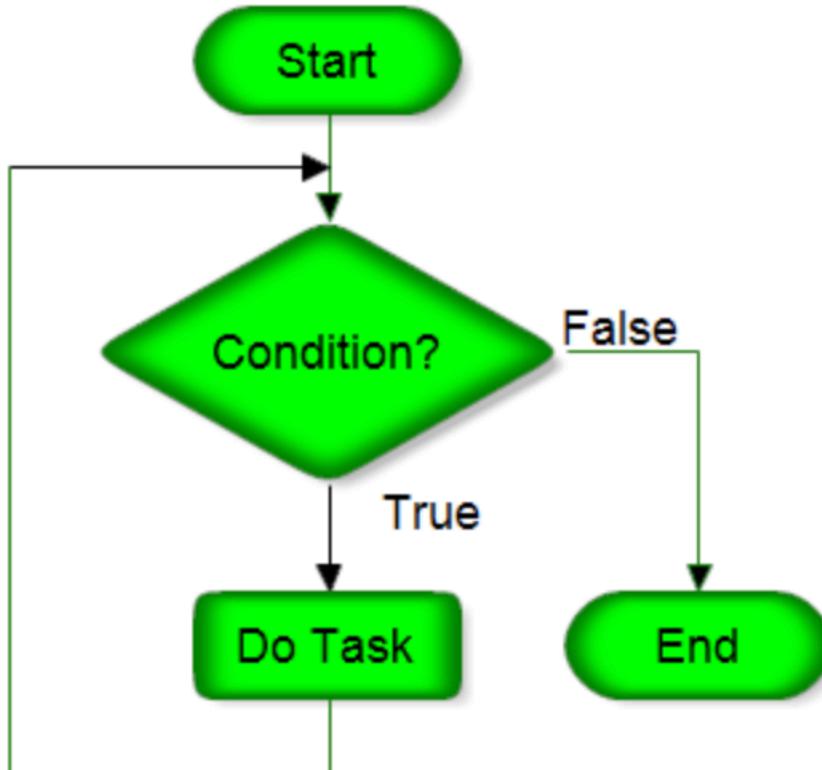


DO WHILE LOOP

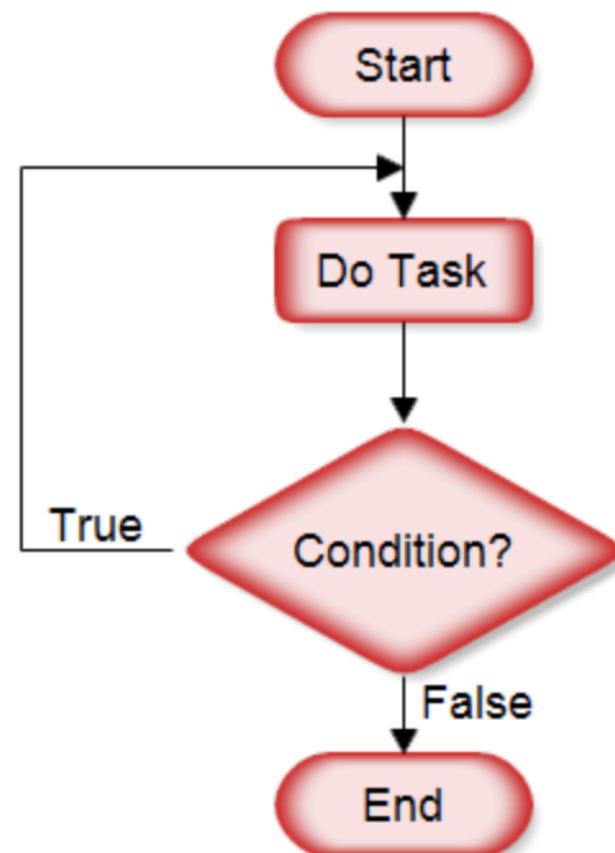
Do While Loop



While Loop



Do While Loop



Types of loops

- An **infinite** or **endless** loop is a loop that repeats indefinitely
 - has no terminating condition
 - the exit condition is never met or the loop is instructed to start over from the beginning.
 - often mistakes made by new programmers.
- A **nested** loop appears inside any other **for**, **while** or **do while** loop.

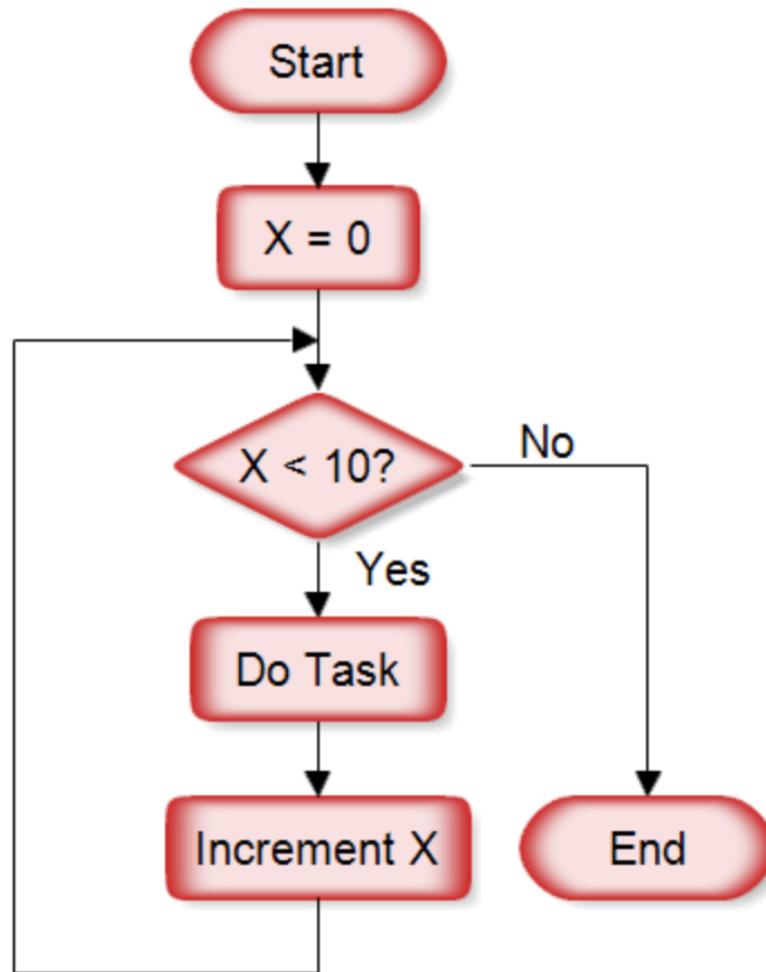
Loops

- Loop is used to save time
 - Allows using the same lines of code many times

LOOP EXAMPLE

- Have a robot perform the task 10 times
 - For example, have the task be moving in a rectangle shape
- Define variable X
- Have X count from 0 to 10

LOOP EXAMPLE

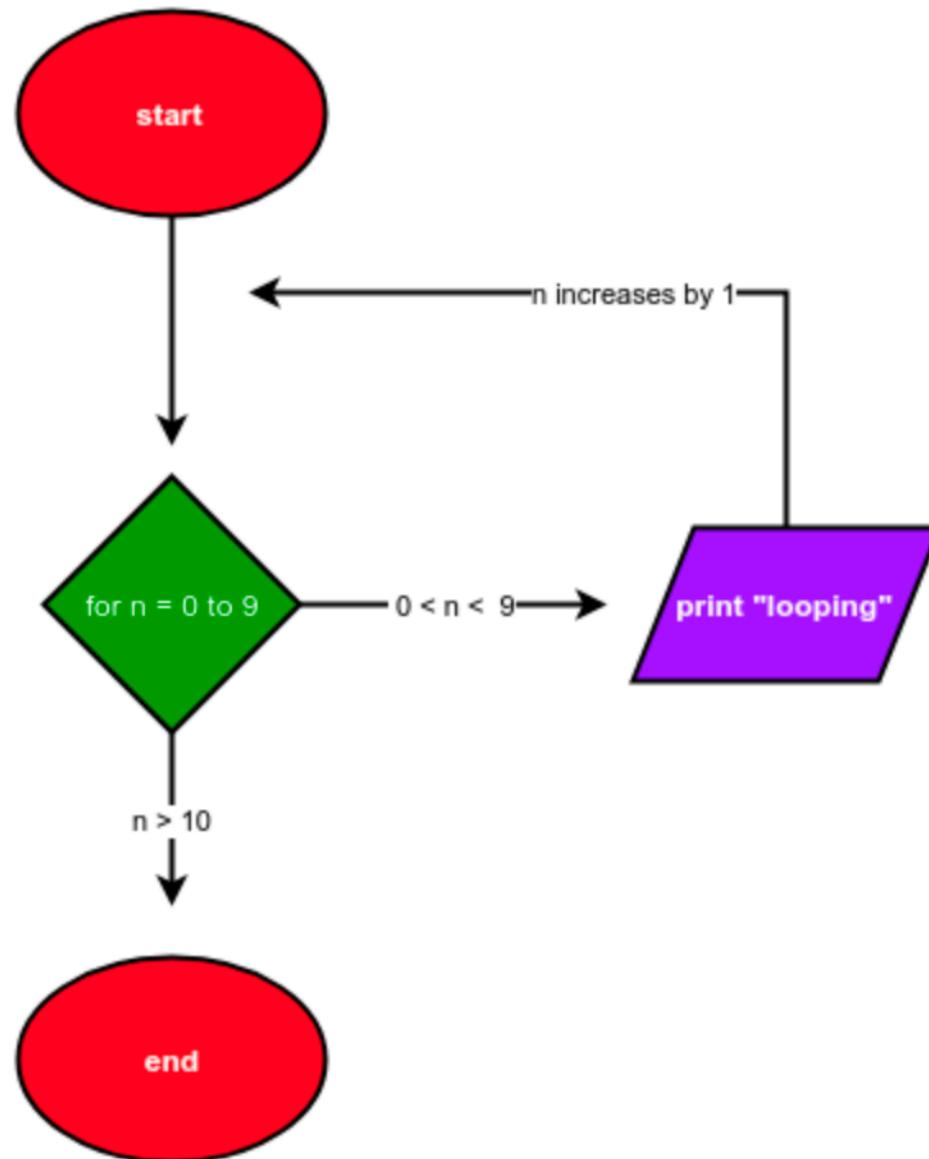


Flowchart Example

- Print the numbers 3,5, and 7 using loops

FLOWCHART LOOPS

- Flowcharts can also be used to describe programs which contain *for loops* and *while loops*
- *Example:*
 - Create a *flowchart that loops 10 times*
 - *Print the word “looping” every time*
 - *Stop when the program ends*

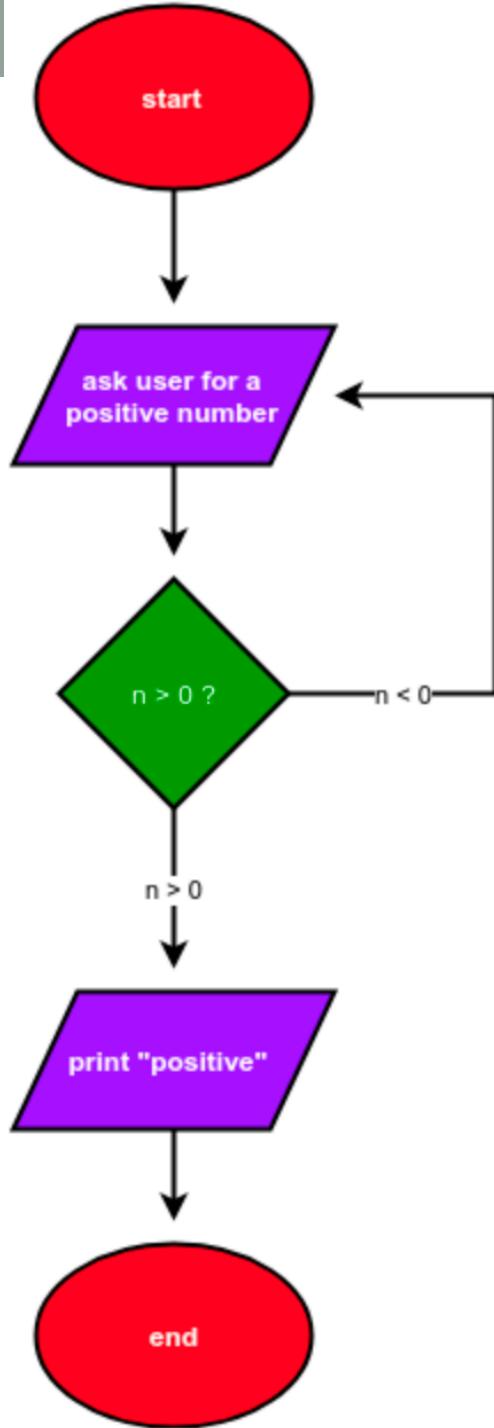


Program Code

- # start
- for i in range(10):
 - print("looping")
- # end

While loop

- Create the flowchart for the following program:
 - The program starts.
 - The program asks the user for a positive number.
 - If the number is negative, the program asks the user for a positive number again.
 - If the number is positive, the program prints "positive".
 - Finally, the program ends



Code

- # start
- num = -1
- while num < 0:
 - num = input("Enter a positive number: ")
 - num = float(num)
- print("positive")
- # end

FLOWCHART EXERCISES

Flowchart exercise

- a. Have your robot START.
- b. Have your robot move forward.
- c. Have your robot make a decision with two choices. (YES or NO)
 - Is the Touch Sensor Pressed?
 - If yes, turn right and stop
- Have your robot STOP after 10 minutes.

Flowchart exercise

- Write a program/directions that will move your “Robot” through an obstacle course.
- To move forward you must tell each wheel to move and how far.

Number addition

- Create a flowchart for adding two numbers entered by the user

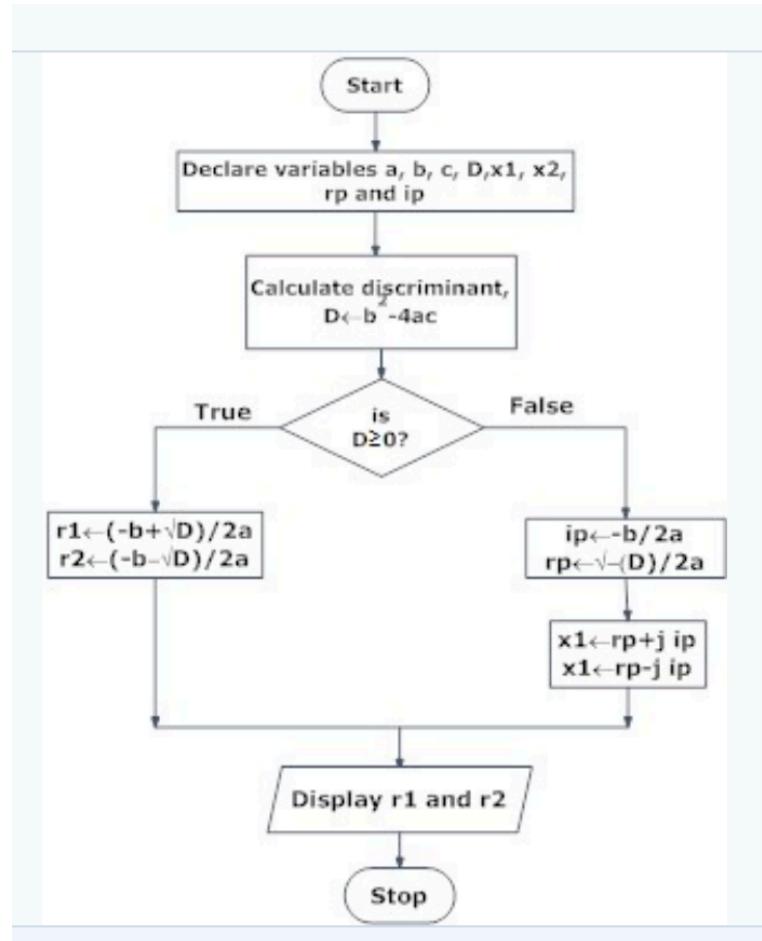
Finding max number

- Find the largest among three different numbers entered by the user

Roots of quadratic equation

- Find all the roots of a quadratic equation
 $ax^2+bx+c=0$

Roots of quadratic equation



Fibonacci Series

- The Fibonacci Sequence is the series of numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...
- The next number is found by adding up the two numbers before it.
 - The 2 is found by adding the two numbers before it (1+1)
 - The 3 is found by adding the two numbers before it (1+2),
 - And the 5 is (2+3),
 - and so on!

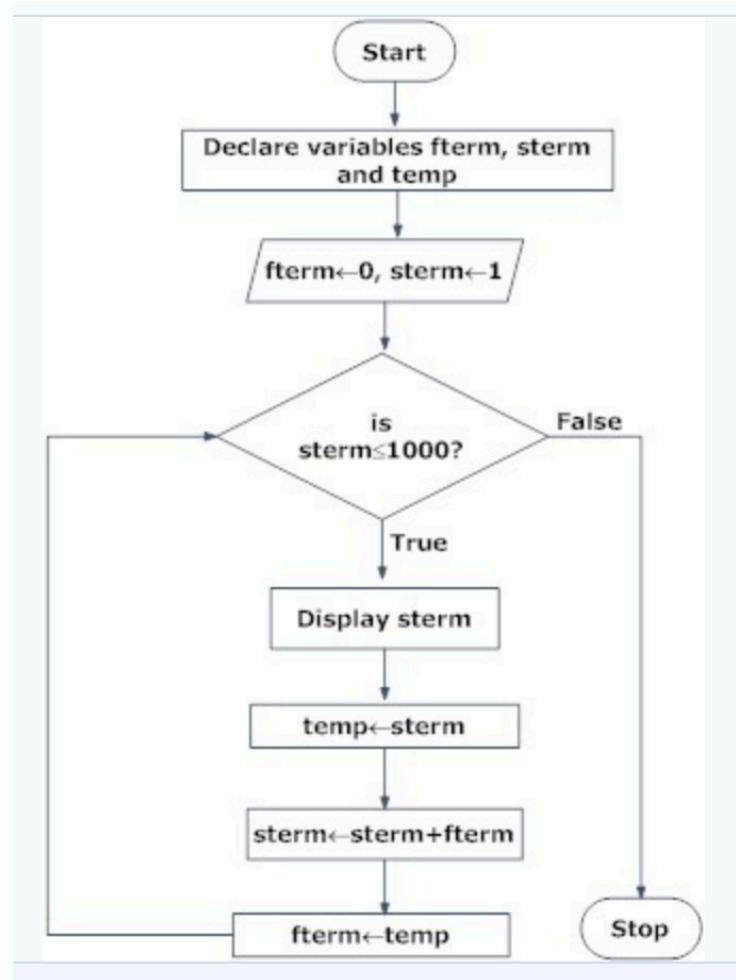
Fibonacci Series

- The Fibonacci Sequence is the series of numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
- Example: What is the next number in the sequence above?
 - the next number in the sequence above is $21+34 = 55$

Flowchart Exercise - Fibonacci Series

- Find the Fibonacci series till term ≤ 1000

Fibonacci Series

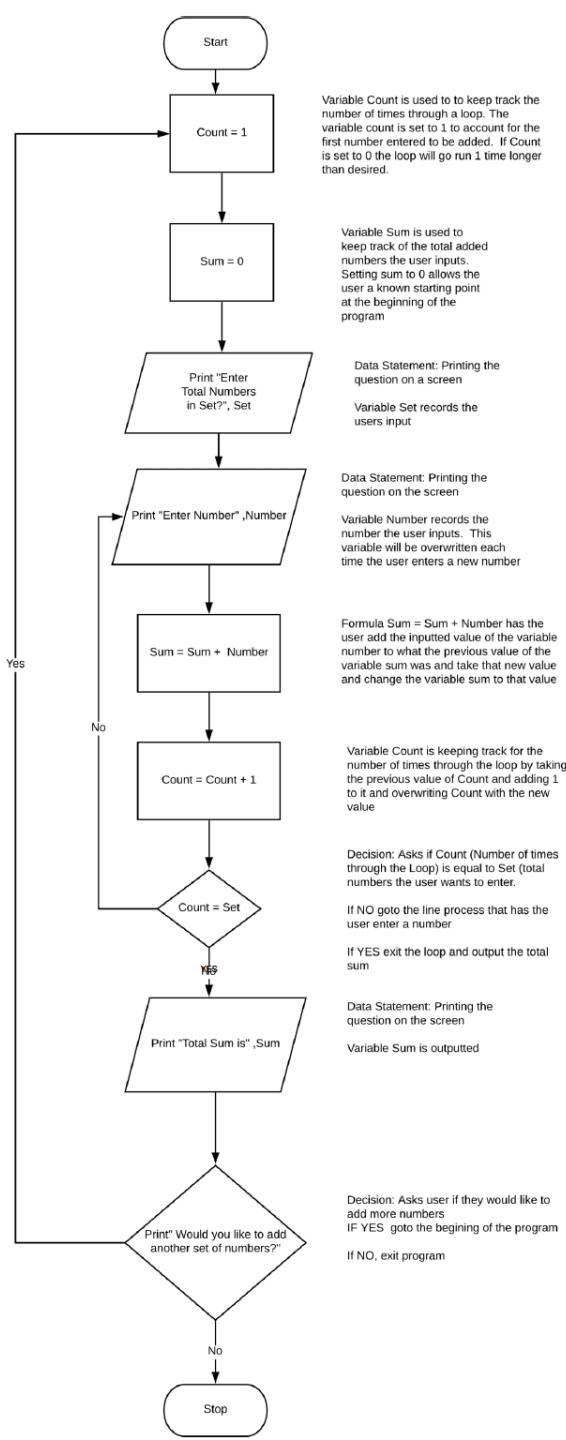


Flowchart exercise - Adding numbers

- User enters total numbers they would like to add together
 - The number of values that the user has in a set of must be greater than 3
- User enters each number to be added one at a time until the Set is full.
- Output the summation of all the numbers added

Flowchart exercise - Adding numbers (2)

- Modify the program to the following:
- Ask user if they would like to add a second set of numbers
 - User decides if they want to enter another set of numbers to be added

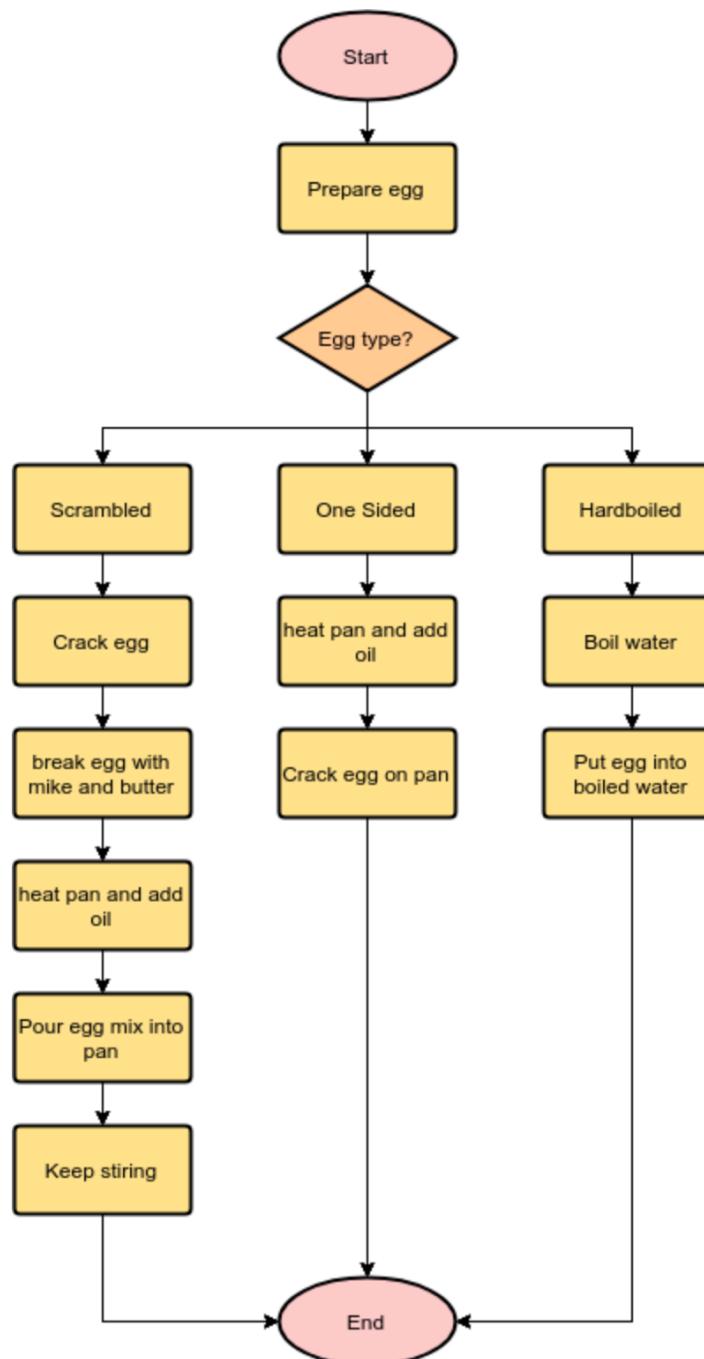


Flowchart Exercise - Average

- Modify the program to do the following:
 - Calculate Average of the set of numbers. Create a variable called Average.

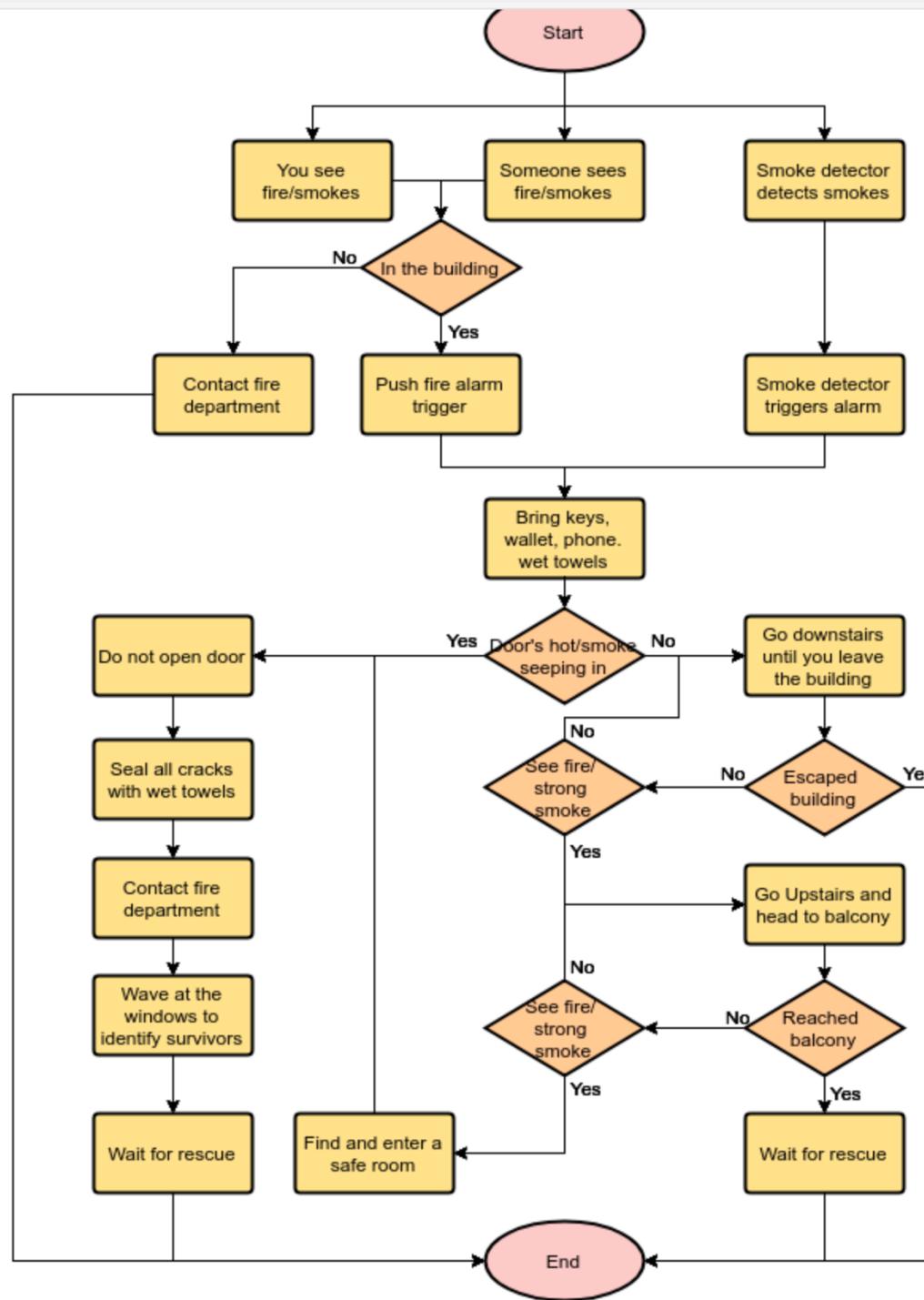
Flowchart Exercise

- Cook an egg:
 - 3 methods are possible:
 - Scrambled
 - One-sided
 - Hardboiled



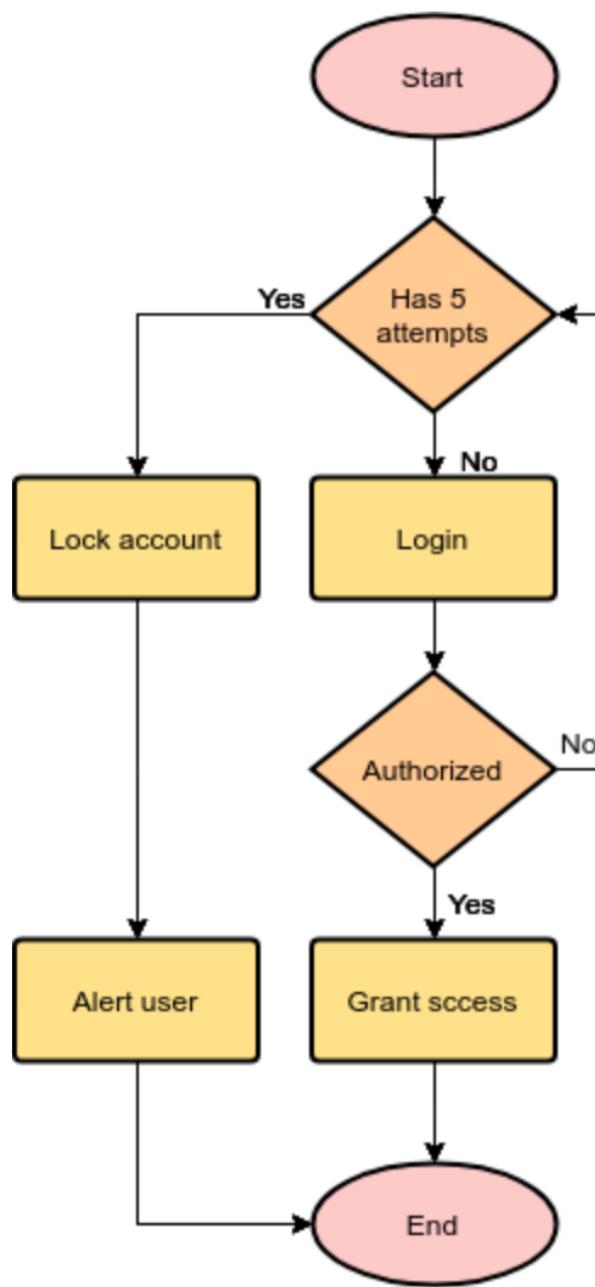
Flowchart Exercise

- Fire evacuation plan
 - What are the variables?
 - Smoke detected
 - Via robot's sensors
 - From external smoke detectors
 - Reporting results to robots
 - User reports
 - People calling 911
 - What could a robot do?



Flowchart Example

- Create a login process
 - User is allowed to attempt to login up to 5 times



- Example: create a flowchart for store check-out

