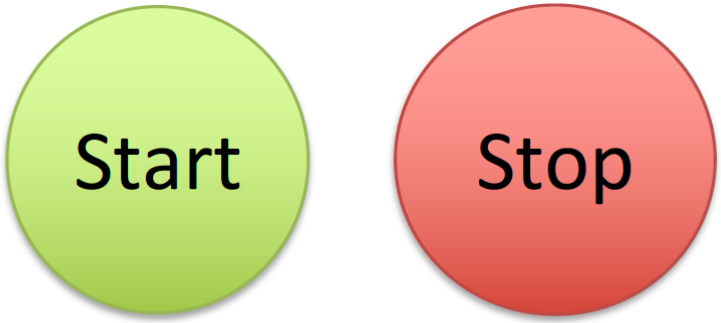
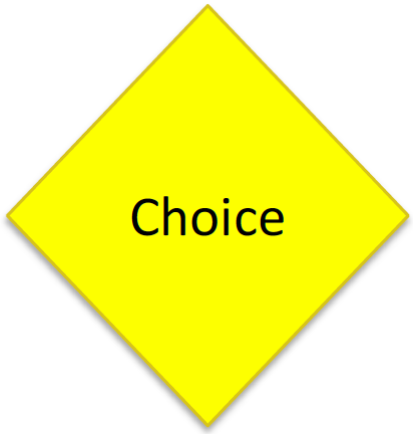
CISC 1003 - EXPLORING ROBOTICS

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

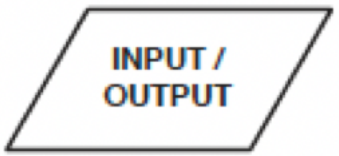


Yellow diamonds represent a choice or decision based on a question. This must be at least a yes/no decision.

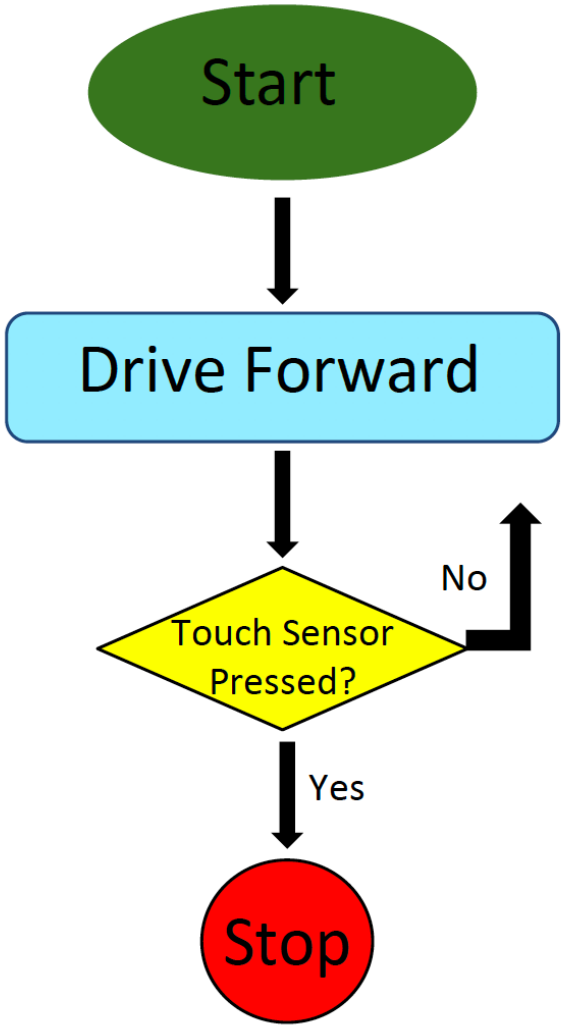
These rectangular blocks represent actions in the program

KISS INSTITUTE FOR PRACTICAL ROBOTICS

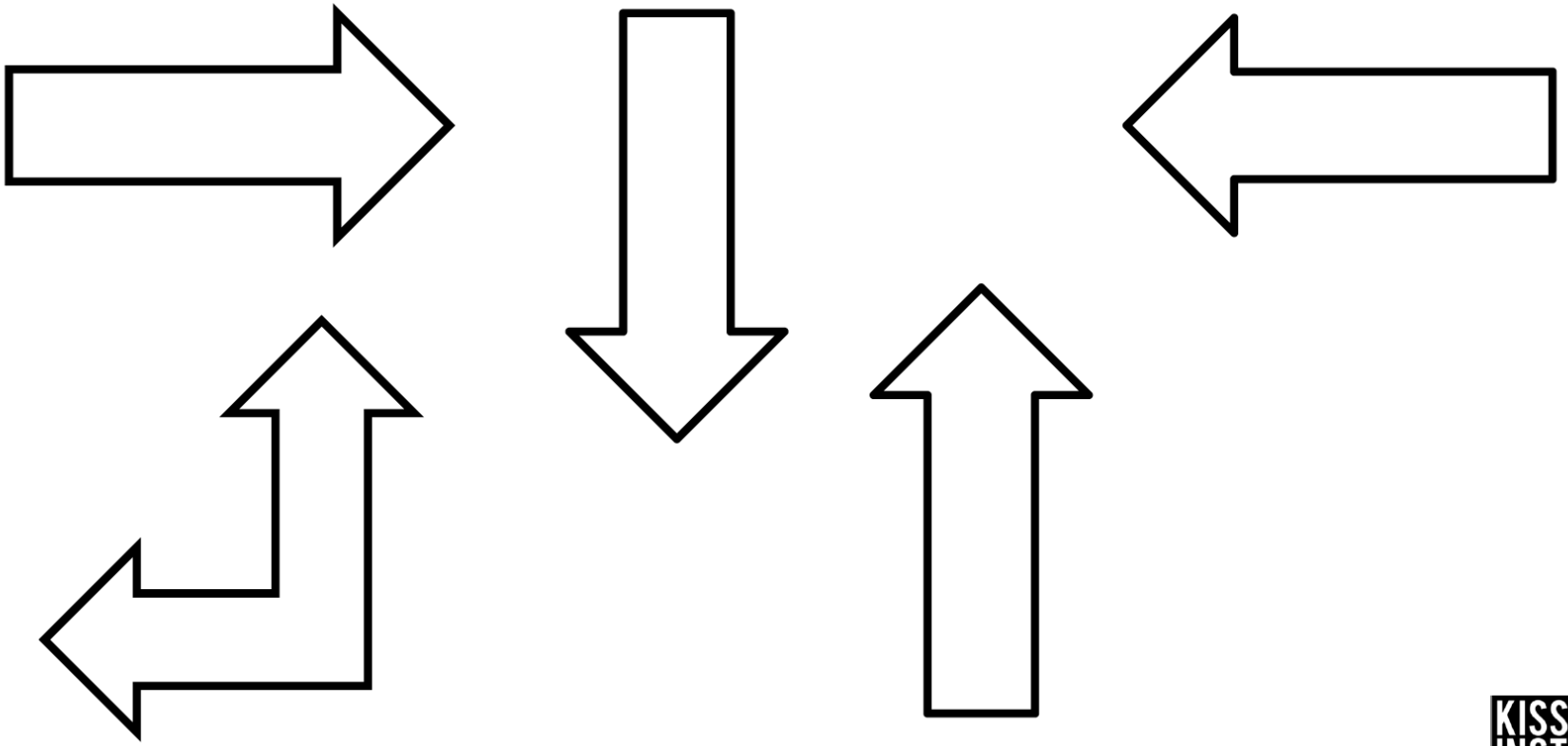
Flowcharting



- 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)

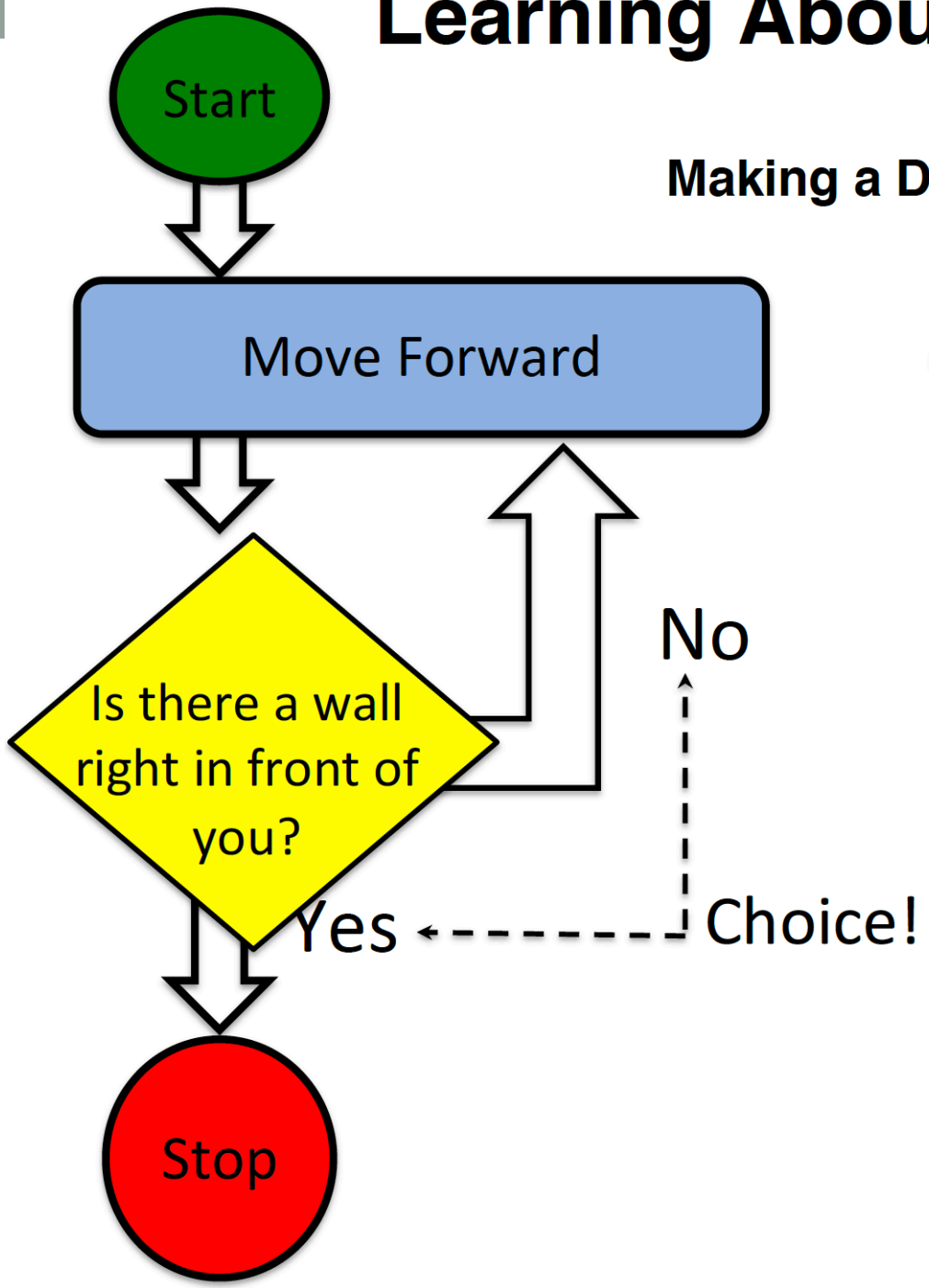

Start
Drive Forward
Touch Sensor Pressed?
No
Yes
Stop

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

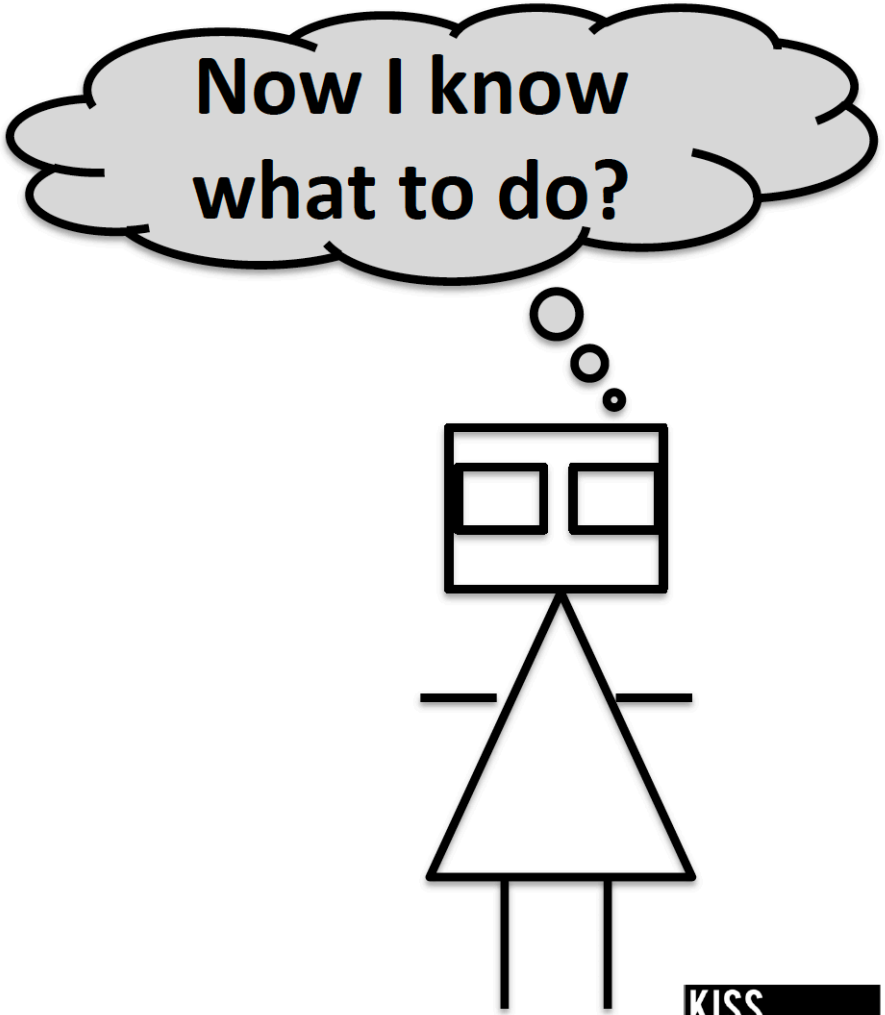


KISS INSTITUTE FOR PRACTICAL ROBOTICS

Learning About Flowcharts

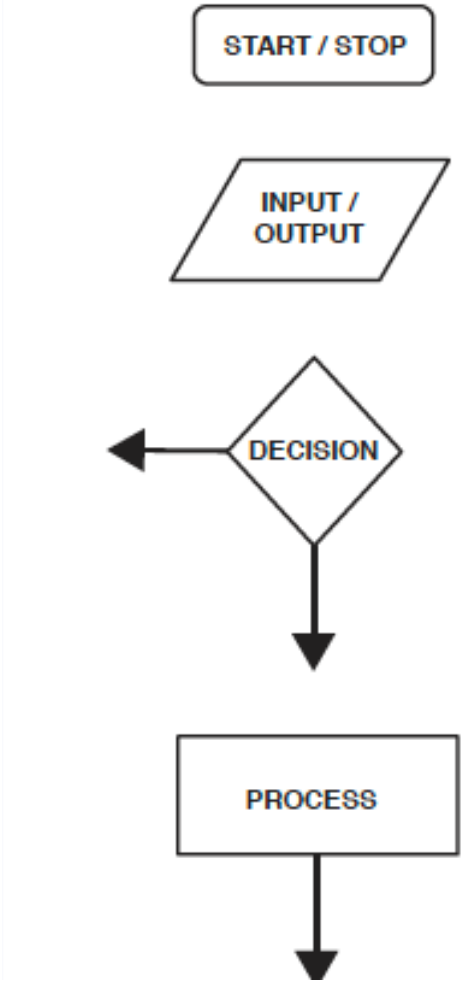


Making a Decision:



KISS INSTITUTE FOR PRACTICAL ROBOTICS

Common Flowchart Symbols



*Robot Programming: A Guide to Controlling Autonomous Robots, C. Hughes and T. Hughes

Program Variables

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

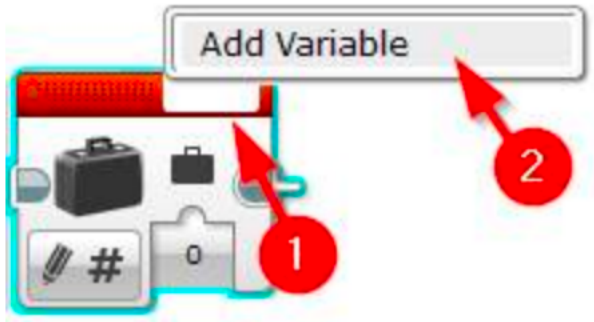
Variables in EV3

- 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**

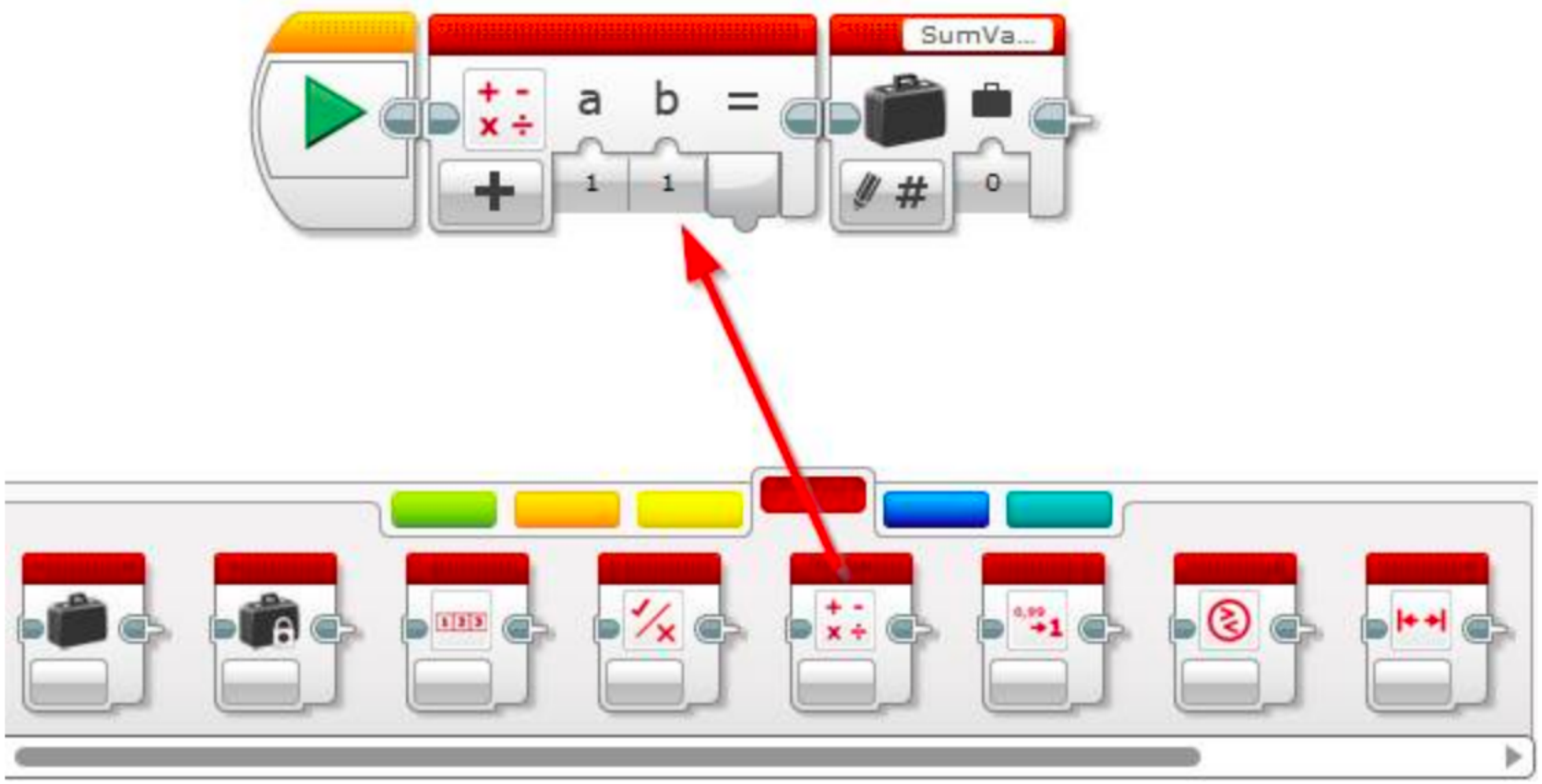


Declaring a Variable in EV3

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

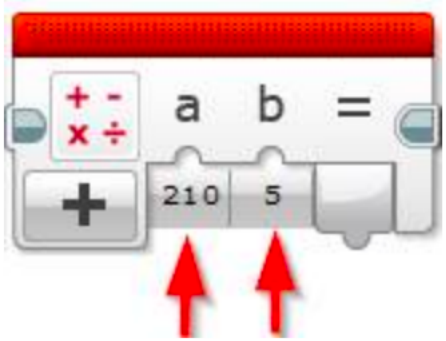
Write Data to the Variable

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



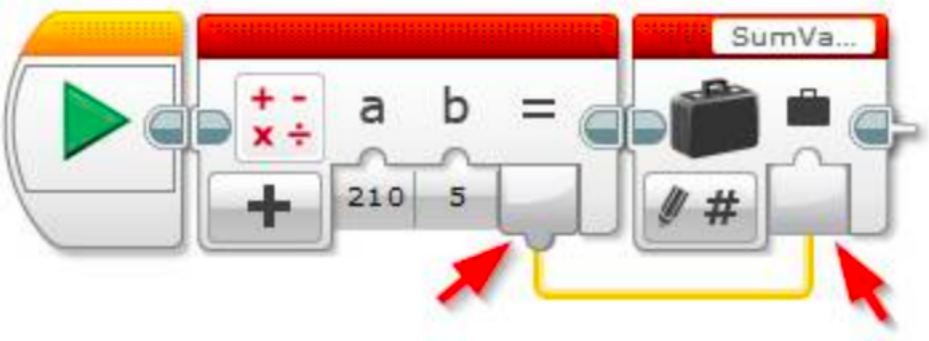
Write Data to the Variable

- 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**:

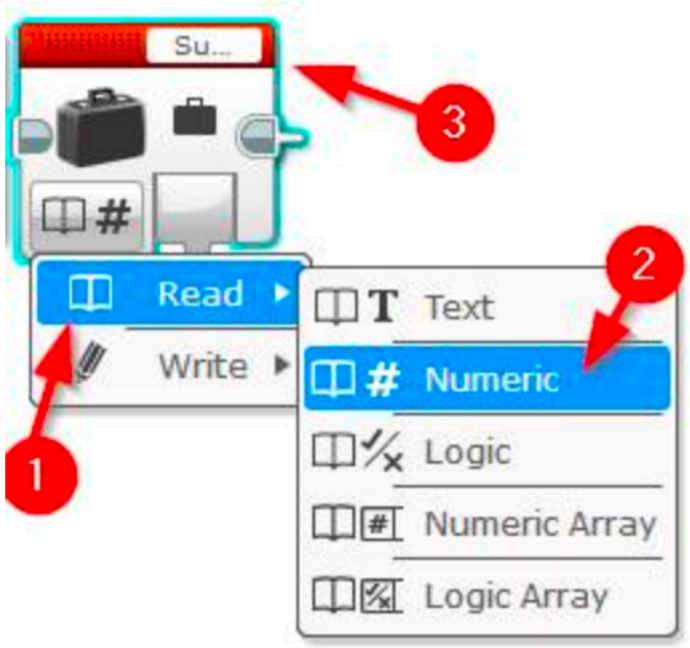


Writing Data to a Variable EV3

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

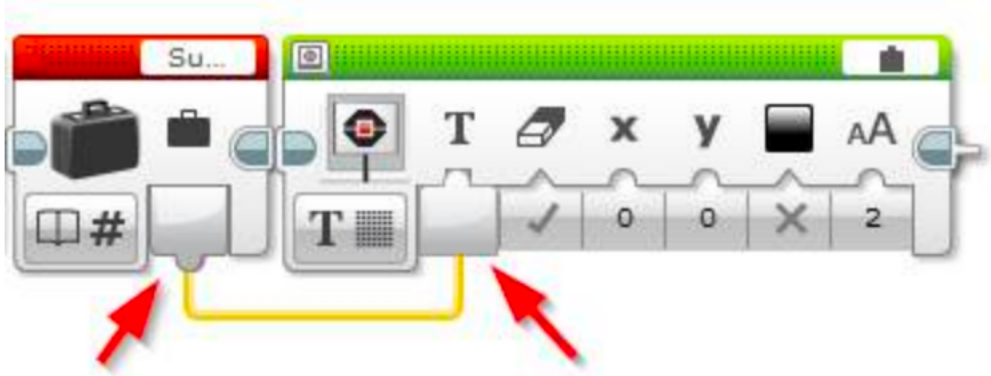
Reading Data from a Variable in EV3

- 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**



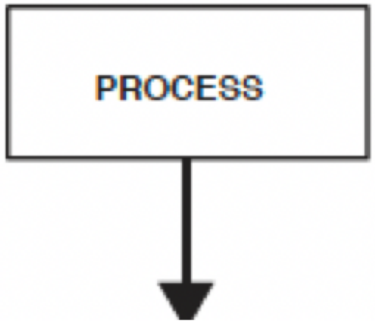
Display the Variable on Screen

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



Flowchars and Variables

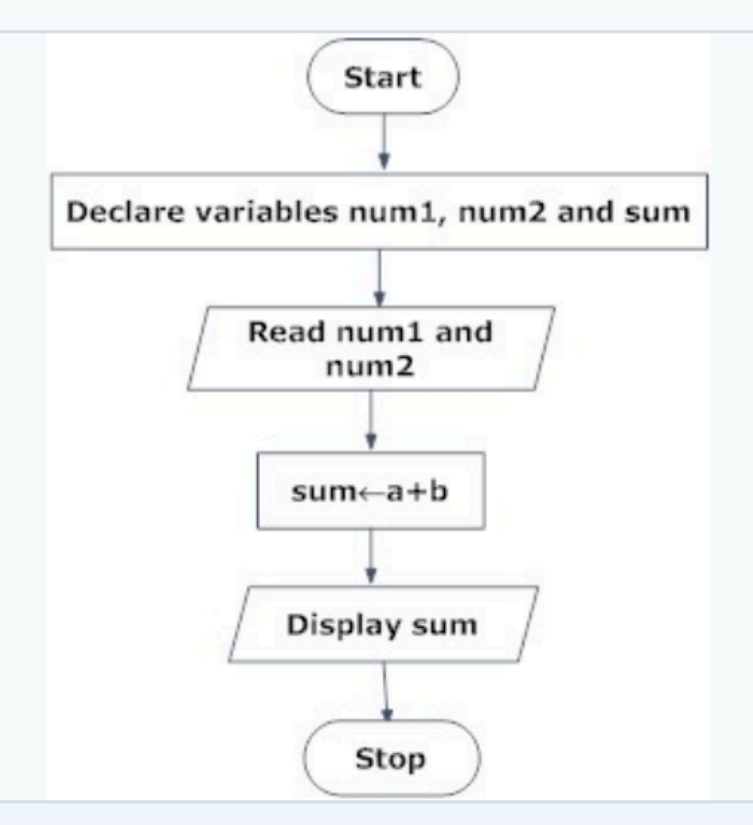
- Process:
  - The process symbol contains brief descriptions (a few words) of a rule or some action taking place
  - Represents an action that changes value or location of data
    - For example, assign a value to a variable



Example: Number addition

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

Add two numbers

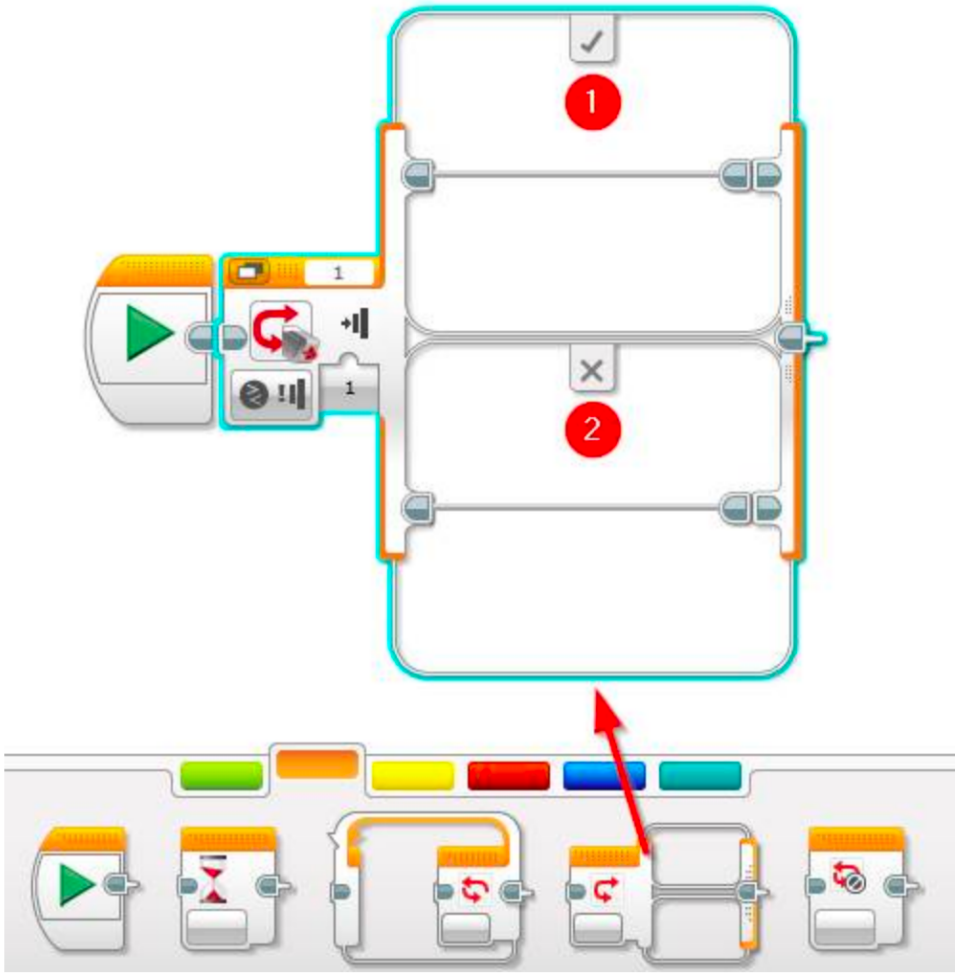


CONDITIONAL STATEMENTS

Conditional Statements

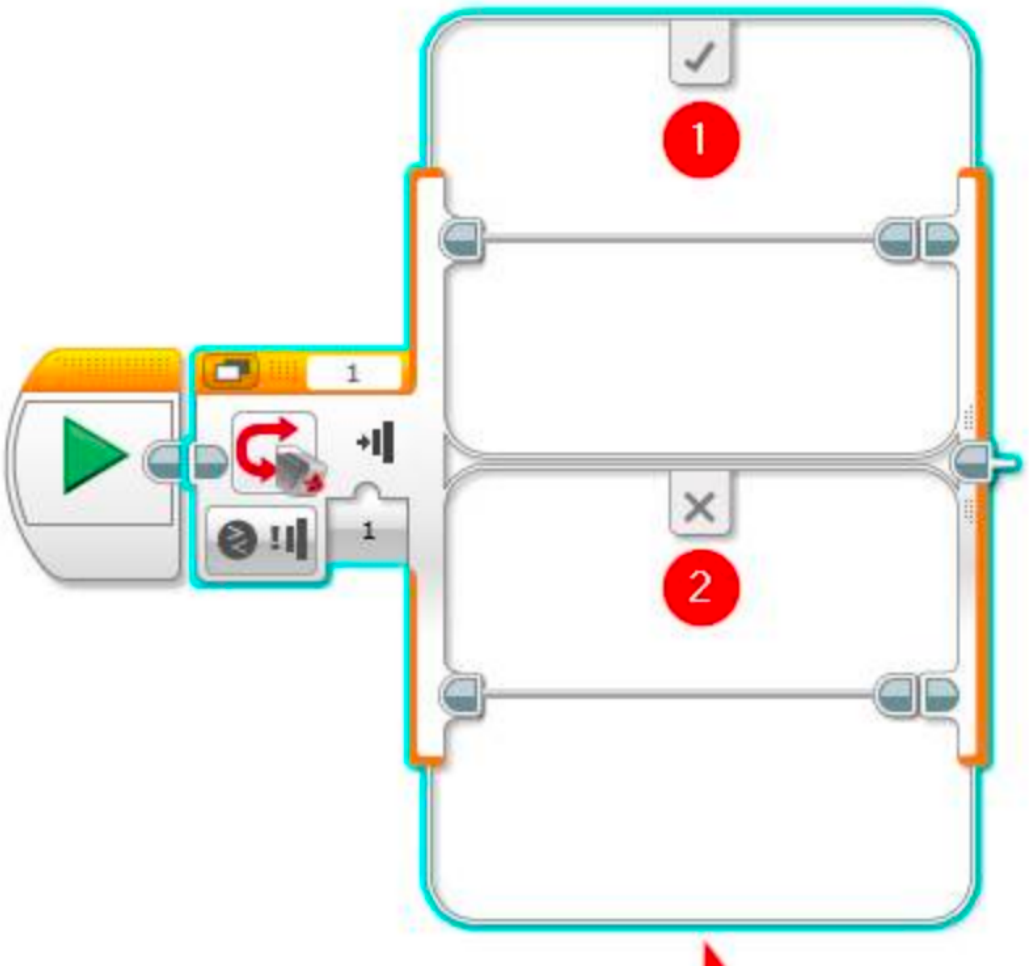
- In many cases actions are conditional
  - Program decides what to do based on the answer to some question or on some condition that is observed
- In EV3 – implemented by a switch block

Switch Block in EV3



Switch block in EV3

- True = 1
- False = 2



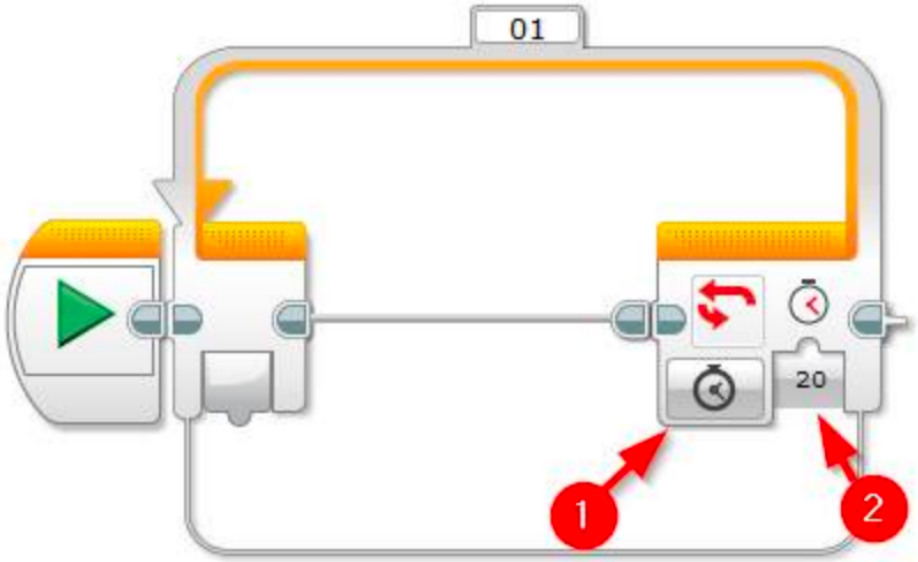
https://www.funcodeforkids.com/lego-mindstorms-ev3-switch-block-in-detail/

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

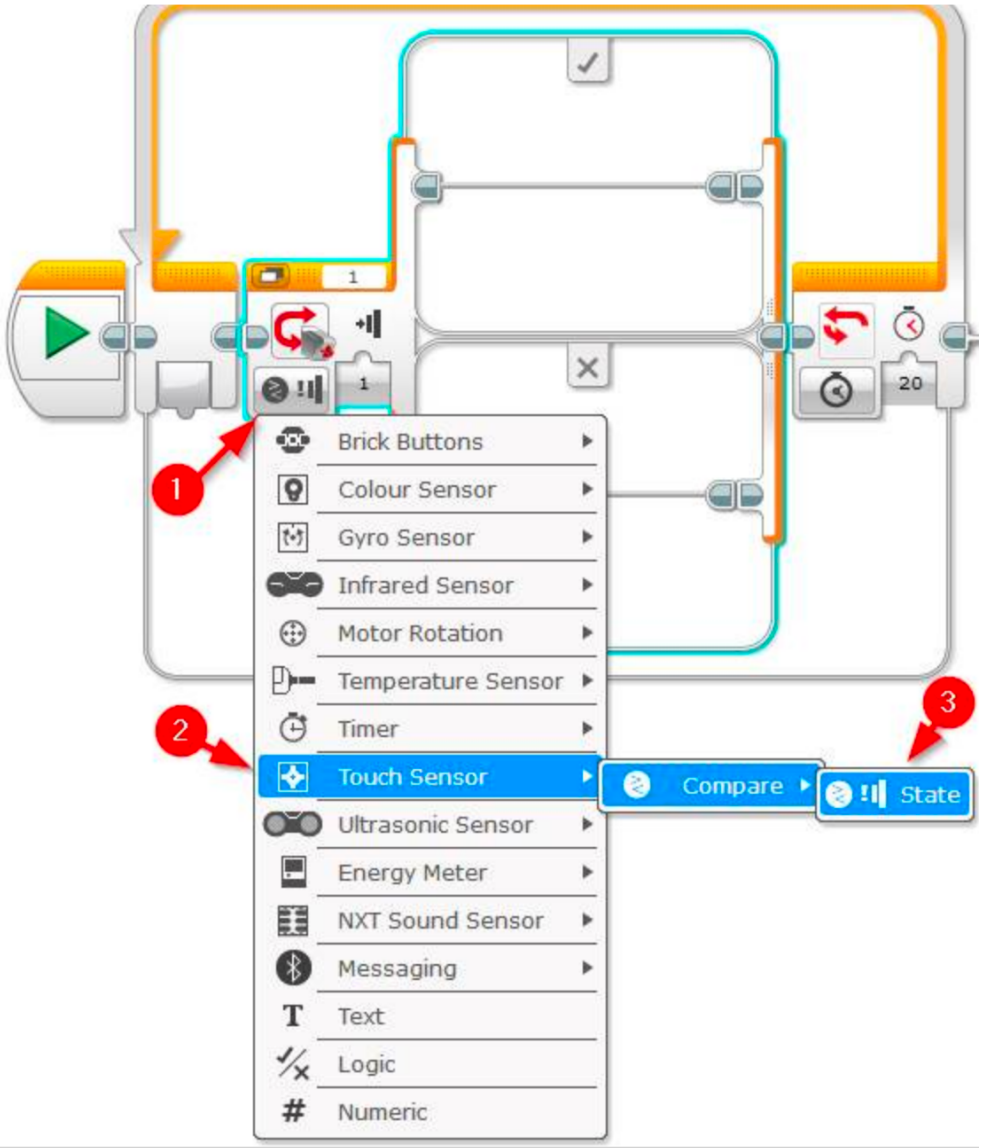
- Loop logic:
  - run the program for 20 seconds



https://www.funcodeforkids.com/lego-mindstorms-ev3-switch-block-in-detail/

Switch block in EV3

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

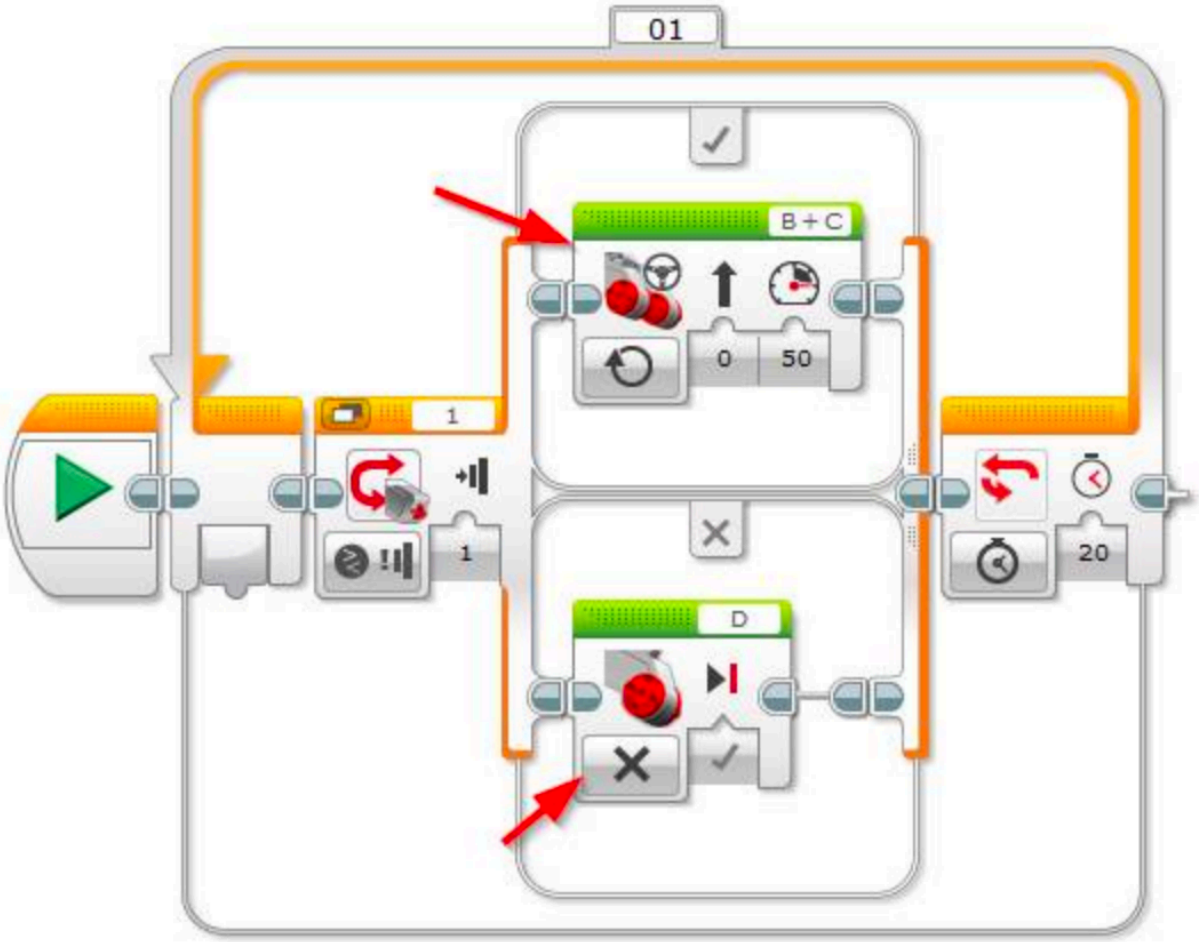


https://www.funcodeforkids.com/lego-mindstorms-ev3-switch-block-in-detail/

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

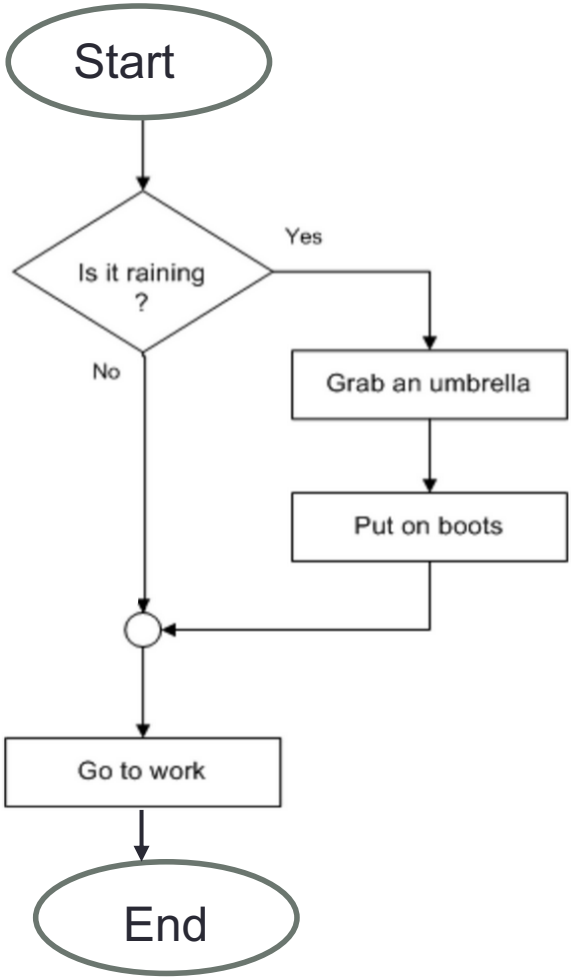


https://www.funcodeforkids.com/lego-mindstorms-ev3-switch-block-in-detail/

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?

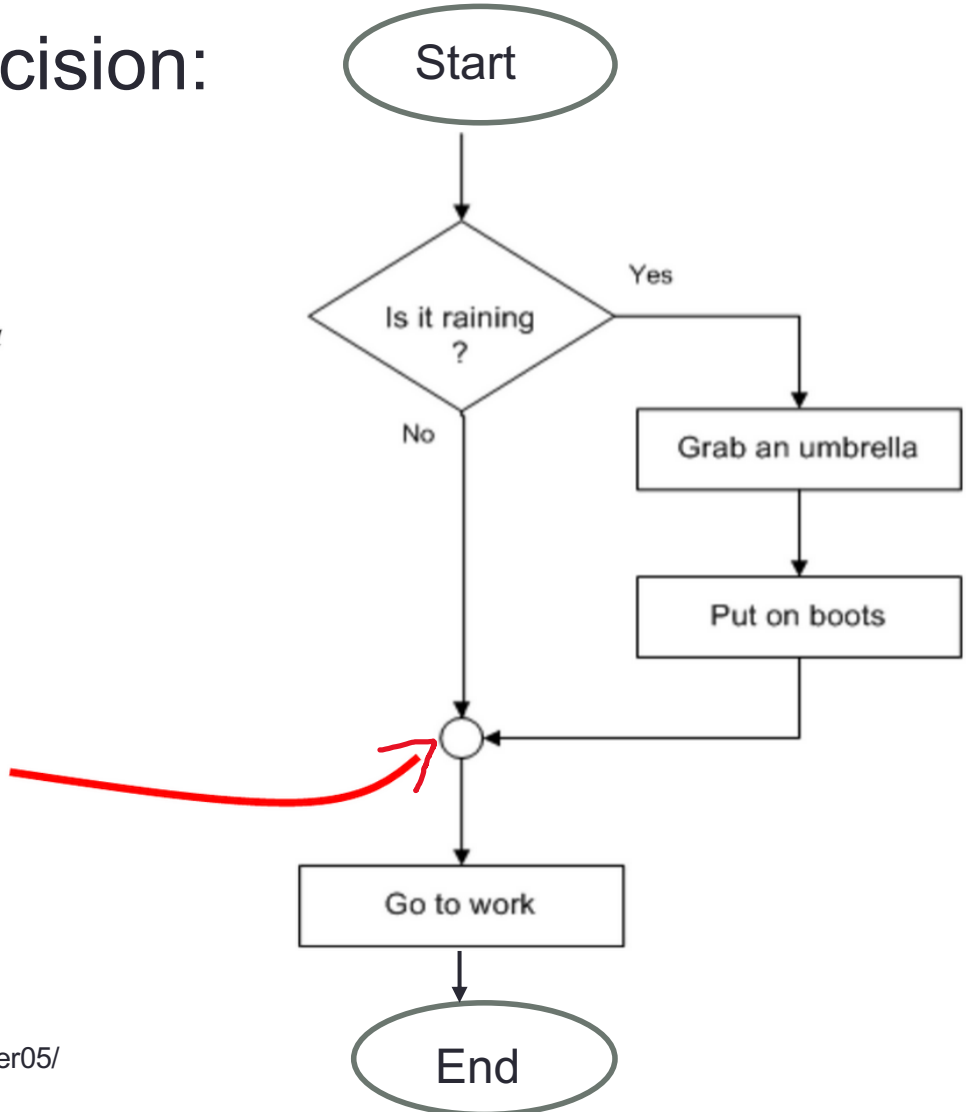
Conditional Statement Flowchart



http://web.cs.iastate.edu/~smkautz/cs127f16/notes/chapter05/

Conditional Statement Flowchart

- Paths rejoin after decision:

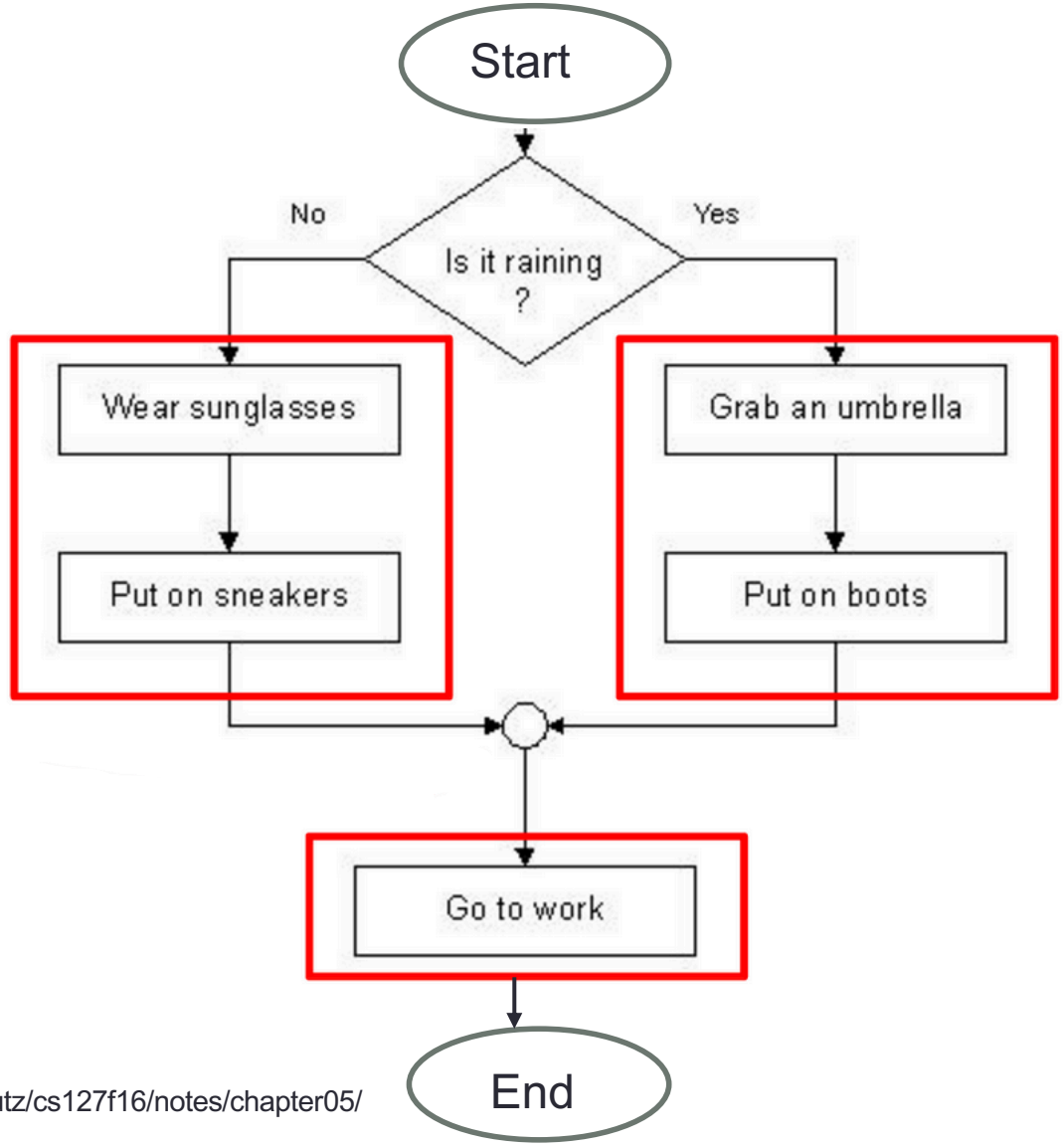


http://web.cs.iastate.edu/~smkautz/cs127f16/notes/chapter05/

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?

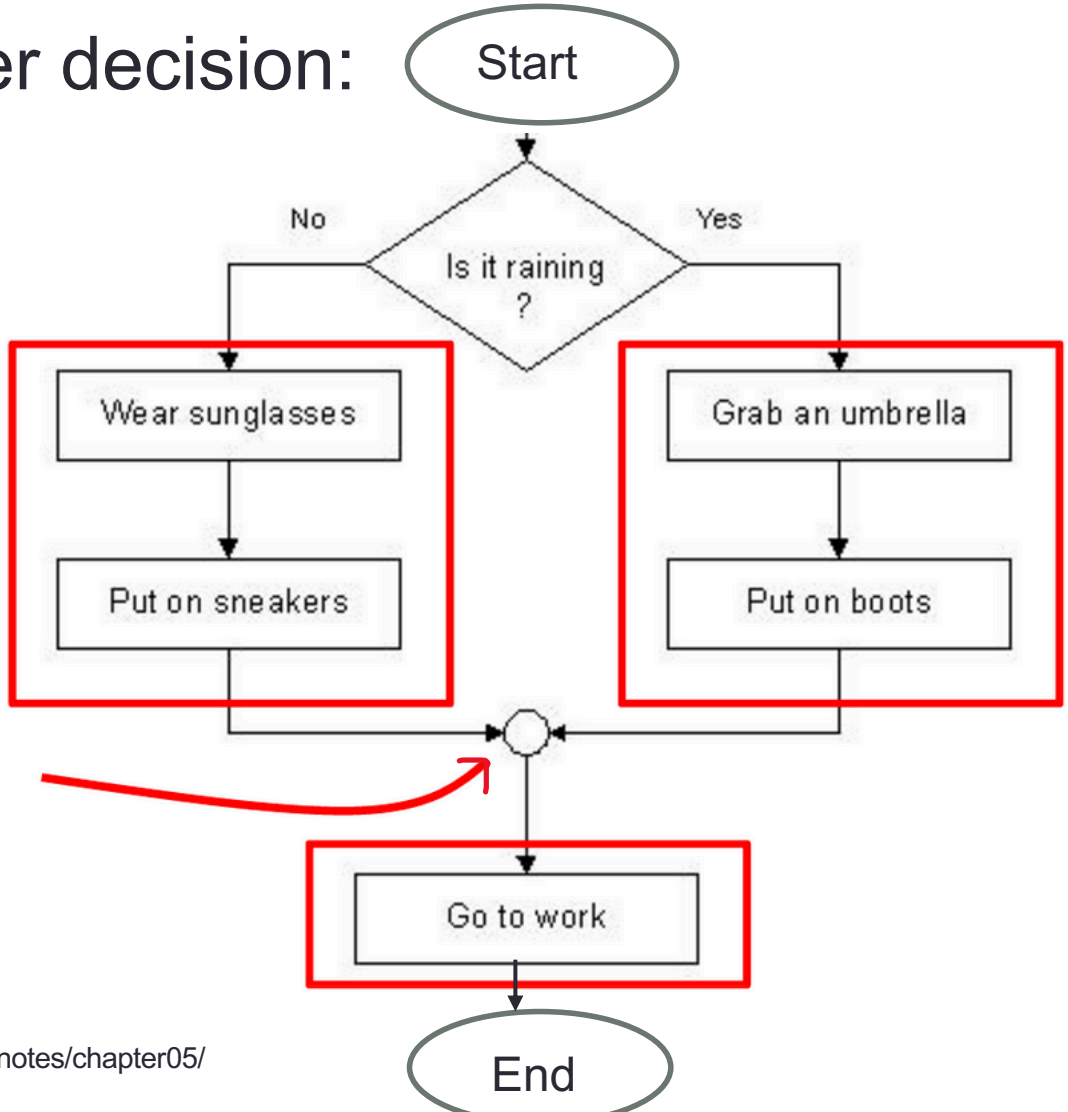
Conditional Statement Flowchart



http://web.cs.iastate.edu/~smkautz/cs127f16/notes/chapter05/

Conditional Statement Flowchart

- Paths rejoin after decision:



http://web.cs.iastate.edu/~smkautz/cs127f16/notes/chapter05/

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”

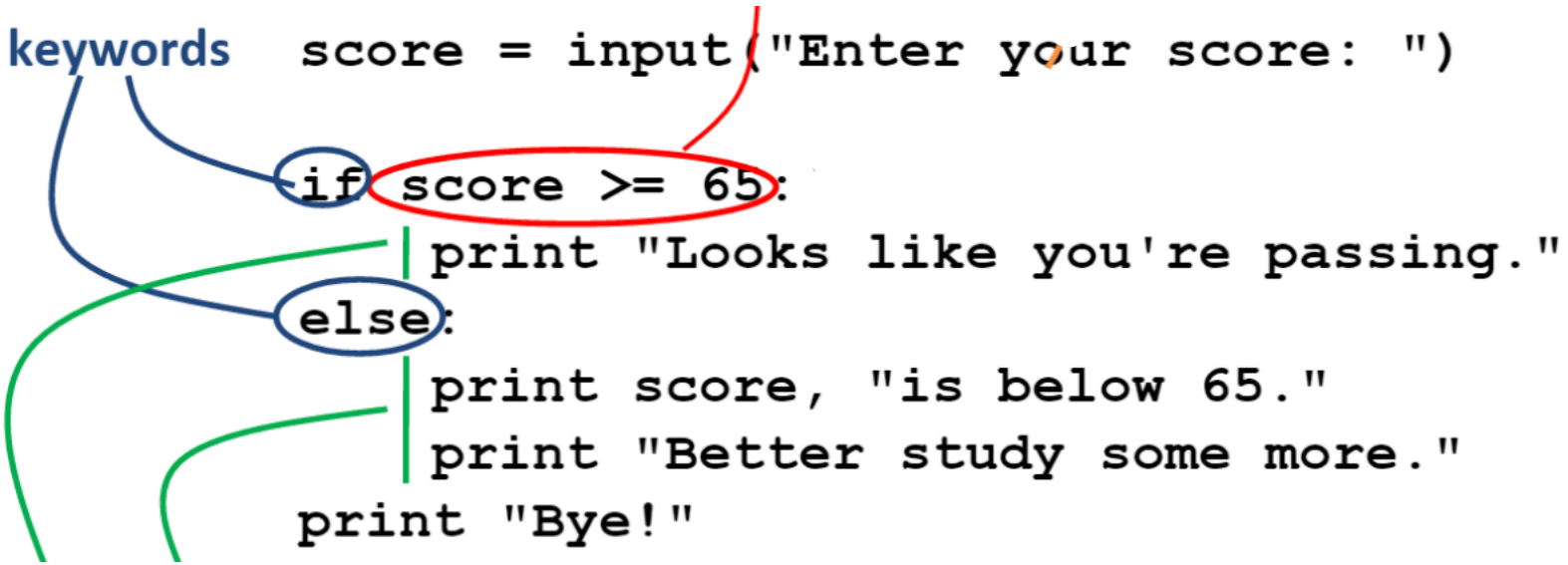
Pseudocode Example

```
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!")
```

http://web.cs.iastate.edu/~smkautz/cs127f16/notes/chapter05/

Pseudocode Example

- Boolean condition:



http://web.cs.iastate.edu/~smkautz/cs127f16/notes/chapter05/

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!"
```

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?

Algorithm

Step 1 : start

Step 2 : read marks or Percentage

Step 3 : if marks >= 80 then grade =A, go to step 7

Step 4 : if marks >= 60 and marks <=80 then grade = B, go to step 7

Step 5 : if marks >=40 and marks <=60 then grade = C go to step 7

Step 6 : display failed

Step 7 : display grade.

Step 8 : stop.

https://sillycodes.com/c-program-to-calculate-grade-of-stude/nt

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

Find Max Number

- Pseudocode:
  - Get Inputs firstNum1, firstNum2 from user
  - If the firstNum1 is larger than firstNum2
    - store it in a variable called ‘larger’
  - Else
    - store firstNum2 in ‘larger’
  - Output the ‘larger’ value

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.

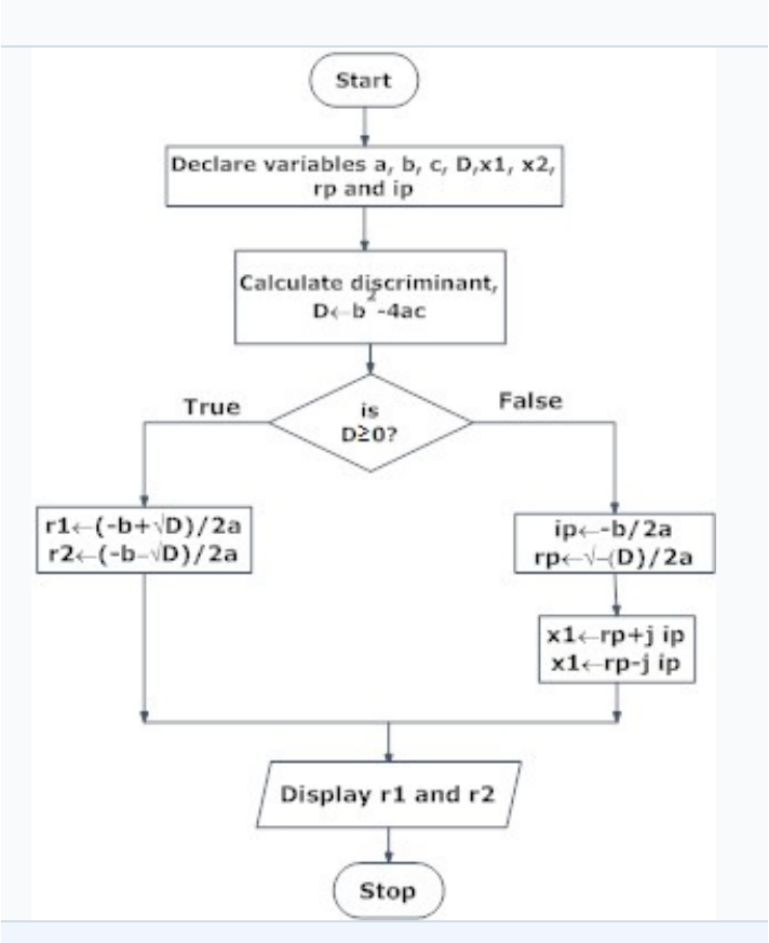
Finding max number

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

Roots of quadratic equation

- Find all the roots of a quadratic equation ax2+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

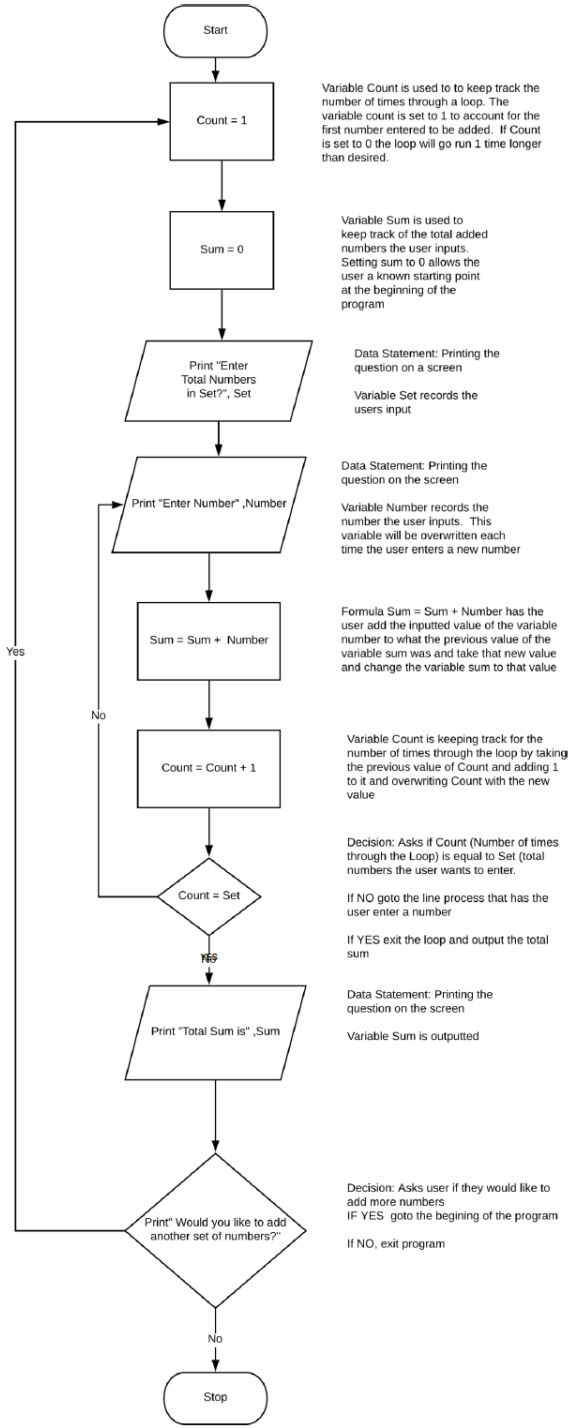
- Find the Fibonacci series till term≤1000

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


Start
Count = 1
Variable Count is used to to keep track the number of times through a loop. The variable count is set to 1 to account for the first number entered to be added. If Count is set to 0 the loop will go run 1 time longer than desired.
Sum = 0
Variable Sum is used to keep track of the total added numbers the user inputs. Setting sum to 0 allows the user a known starting point at the beginning of the program
Print "Enter Total Numbers in Set?", Set
Data Statement: Printing the question on a screen
Variable Set records the users input
Print "Enter Number" ,Number
Data Statement: Printing the question on the screen
Variable Number records the number the user inputs. This variable will be overwritten each time the user enters a new number
Sum = Sum + Number
Formula Sum = Sum + Number has the user add the inputted value of the variable number to what the previous value of the variable sum was and take that new value and change the variable sum to that value
Count = Count + 1
Variable Count is keeping track for the number of times through the loop by taking the previous value of Count and adding 1 to it and overwriting Count with the new value
Count = Set
Decision: Asks if Count (Number of times through the Loop) is equal to Set (total numbers the user wants to enter.
If NO goto the line process that has the user enter a number
If YES exit the loop and output the total sum
No
Yes
Print "Total Sum is" ,Sum
Data Statement: Printing the question on the screen
Variable Sum is outputted
Print" Would you like to add another set of numbers?"
Decision: Asks user if they would like to add more numbers
IF YES goto the begining of the program
If NO, exit program
Yes
No
Stop

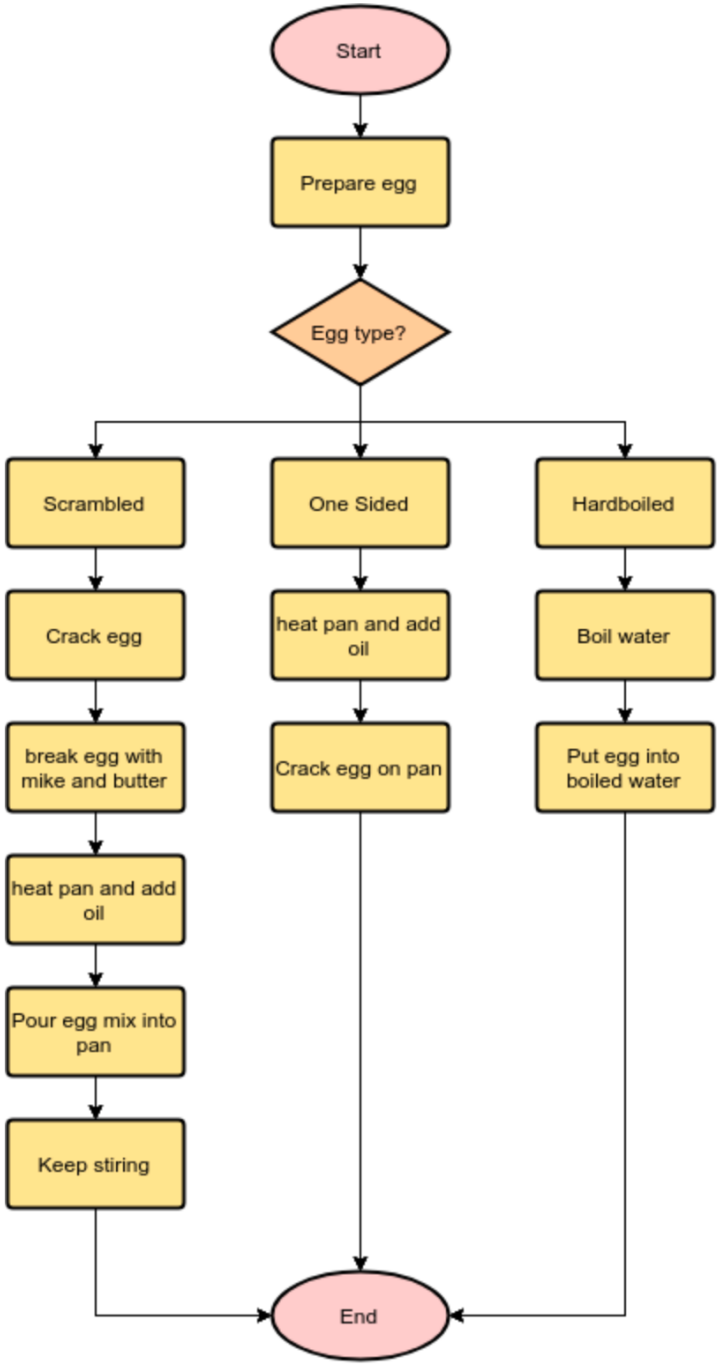
Flowchart Exercise - Average

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

Flowchart Exercise

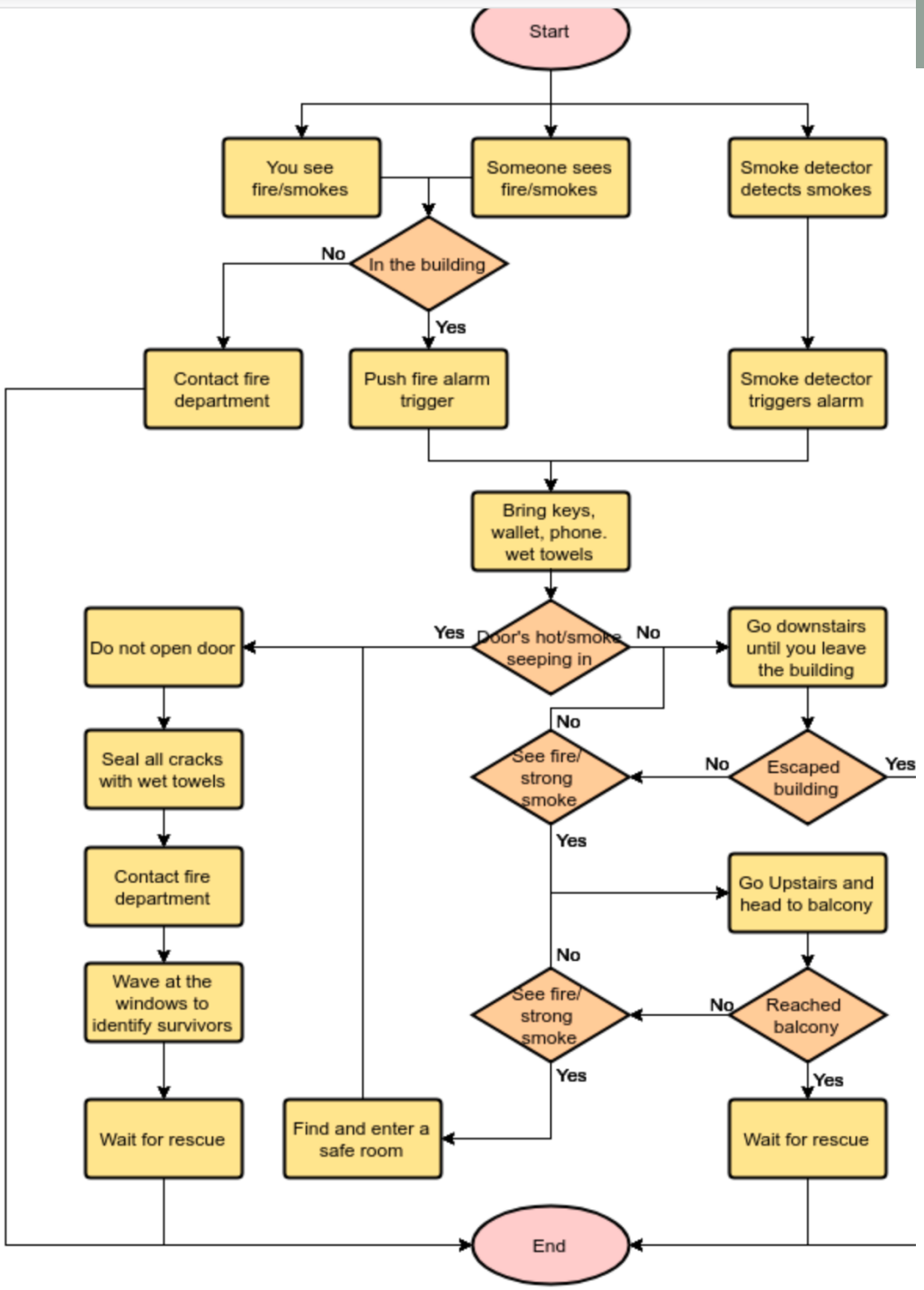
- Cook an egg:
  - 3 methods are possible:
    - Scrambled
    - One-sided
    - Hardboiled

https://medium.com/@warren2lynch/a-comprehensive-guide-for-flowchart-over-50-examples-785d6dfdc380


Start
Prepare egg
Egg type?
Scrambled
One Sided
Hardboiled
Crack egg
heat pan and add oil
Boil water
break egg with mike and butter
Crack egg on pan
Put egg into boiled water
heat pan and add oil
Pour egg mix into pan
Keep stiring
End

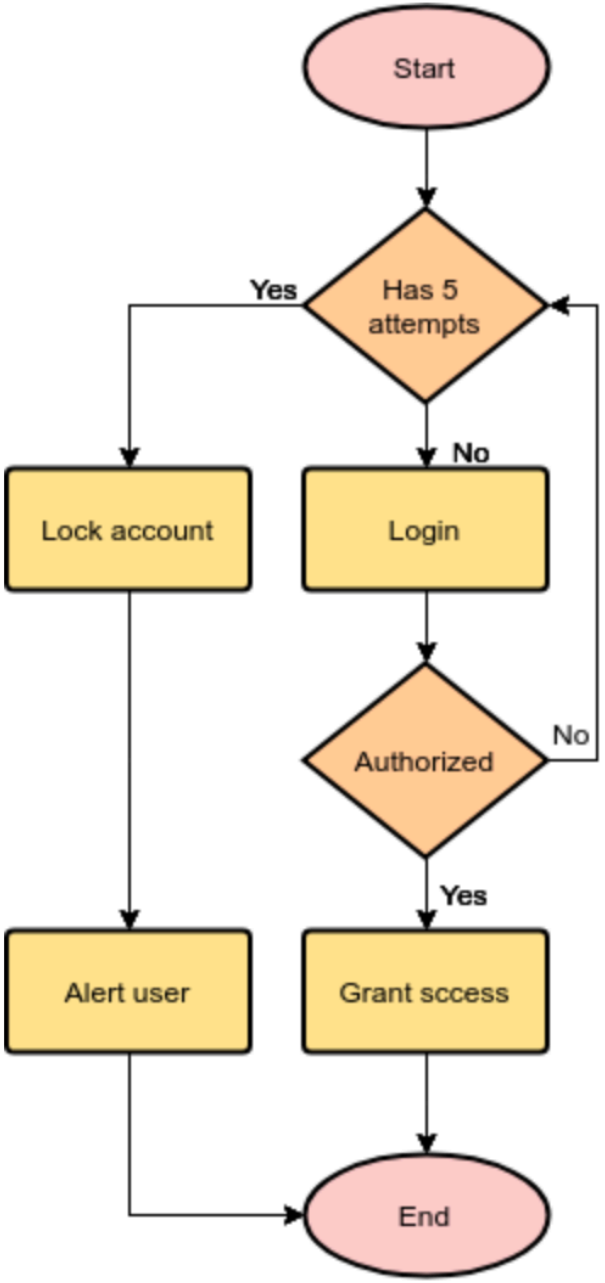
Flowchart Exercise

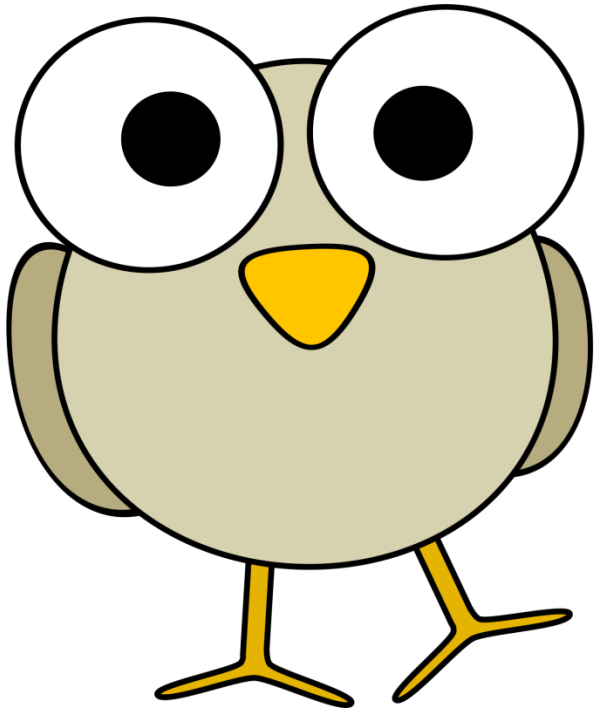
- Fire evacuation plan
  - What are the variables?
    - Smoke detected
      - Via robot's sensors
      - From external smoke detectors
        - Reporting results to robots
      - User reports
        - People calling 911
  - What could a robot do?


Start
You see fire/smokes
Someone sees fire/smokes
Smoke detector detects smokes
No
In the building
Yes
Contact fire department
Push fire alarm trigger
Smoke detector triggers alarm
Bring keys, wallet, phone. wet towels
Yes
Door's hot/smoke seeping in
No
Do not open door
Go downstairs until you leave the building
Seal all cracks with wet towels
No
See fire/ strong smoke
No
Escaped building
Yes
Contact fire department
Yes
Go Upstairs and head to balcony
Wave at the windows to identify survivors
No
See fire/ strong smoke
No
Reached balcony
Yes
Yes
Wait for rescue
Find and enter a safe room
Wait for rescue
End

Flowchart Example

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


Start
Has 5 attempts
Yes
No
Lock account
Login
Authorized
No
Yes
Alert user
Grant sccess
End



??