



SESSION - 5

CALCULATOR



Learning Outcomes:

- **Remember:** The students will list different types of blocks being learnt in the session .
- **Understand:** - They will focus on understanding the conditional statement - IF-ELSE
 - They will also learn more about arithmetic operators
- **Apply:** They will apply If-else structure to build a calculator
- **Analyze:** They will check their understanding by developing a code.
- **Create:** They will create the code in EduBlocks

- In this activity we have to design a basic calculator using basic arithmetic operations.
- Make a program to add, subtract, multiply, divide and find reminder
- Let the student try to make code by himself/herself.
- Take user input as 1st and second number
- Select an option what to do Add, Sub, Mul, Div

TASK 01:-

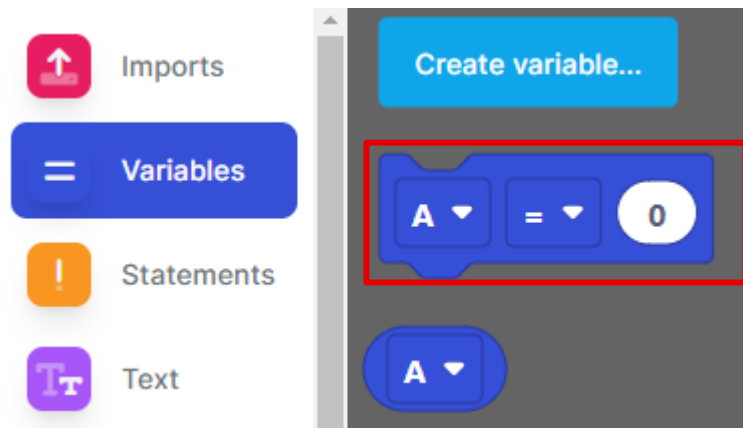
**</> WRITE A PROGRAM TO MAKE A
CALCULATOR**

Program Step 1:-

The screenshot shows the app.edublocks.org interface. On the left, there's a sidebar with 'Restless Rice' at the top, followed by 'Imports', 'Variables', and 'Statements'. The 'Variables' button is highlighted. In the center, a 'Create variable...' button is highlighted with a red box. A dialog box is open on the right, displaying 'app.edublocks.org says' and 'New variable name:' with a text input field containing the letter 'A'. The dialog has 'OK' and 'Cancel' buttons.

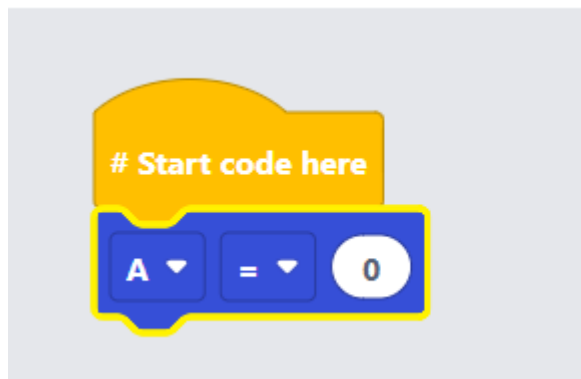
Create a variable and name it as 'A'

Program Step 2:-



The image shows the Scratch IDE interface. On the left, the 'Variables' menu is selected. In the script area, a 'Create variable...' block is at the top, and below it, a block containing 'A = 0' is highlighted with a red rectangular box. The 'A' has a dropdown arrow, and the '0' is in a white circle.

Initialize the "A" variable

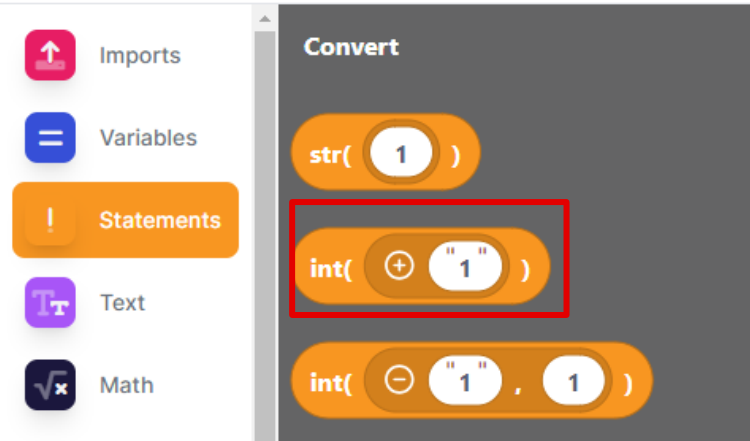


The image shows a Scratch script area. At the top is a yellow 'Start code here' block. Below it is a blue block containing 'A = 0', which is highlighted with a yellow border.

Code

```
1 #Start code here
2 A = 0
3
```

Program Step 3:-



Imports

Variables

Statements

Text

Math

Convert

str(1)

int(+ " 1")

int(- " 1" , 1)

Store an int block to specify that input will be stored as an integer



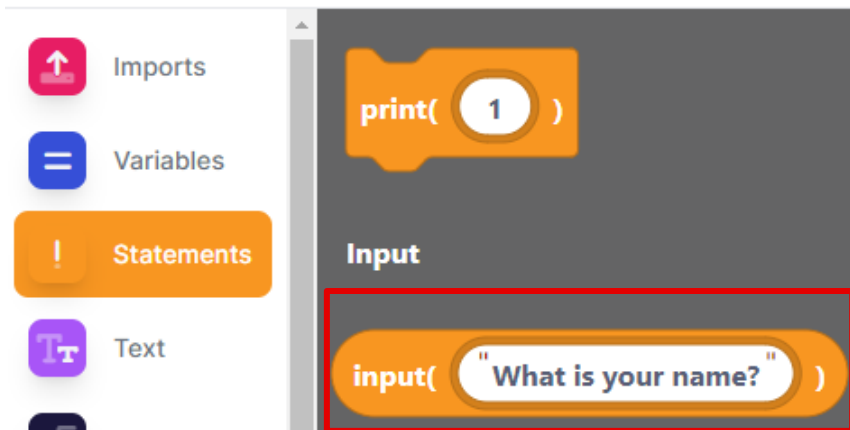
Start code here

A = int(+ " 1")

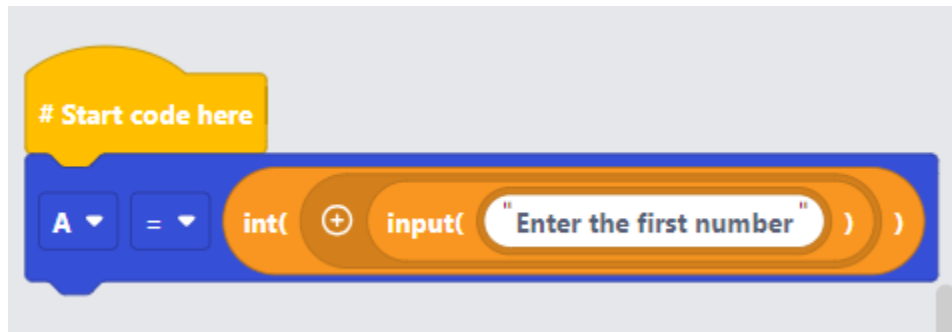
Code

```
1 #Start code here
2 A = int("1")
3
```

Program Step 4:-



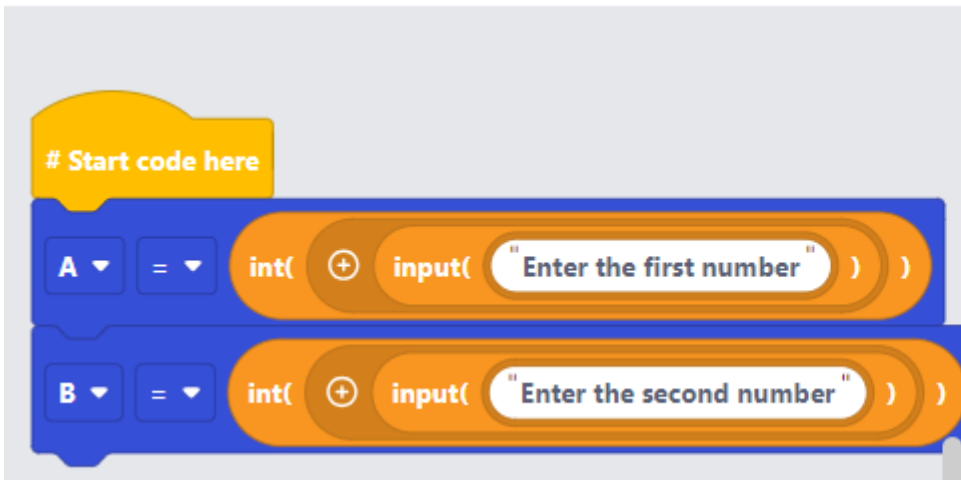
Add an input block to take input from the user



Code

```
1 #Start code here
2 A = int(input("Enter the first number"))
3
```


Program Step 5:-



Code

```
1 #Start code here
2 A = int(input("Enter the first number"))
3 B = int(input("Enter the second number"))
4
```

Create another variable with name 'B' the same way to store the second number

Program Step 6:-

Add a print "Hello World" block and change the "Hello World" with the operation names that calculator can perform

Imports

Variables

Statements

Output

print(Hello World)

print(1)

Start code here

A = int(input(Enter the first number))

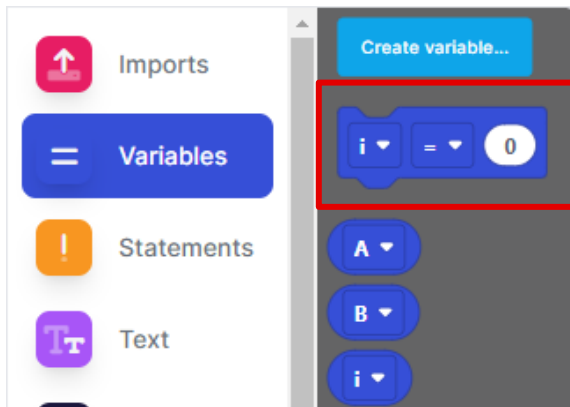
B = int(input(Enter the second number))

print(Add, Sub, Mul, Div)

Code

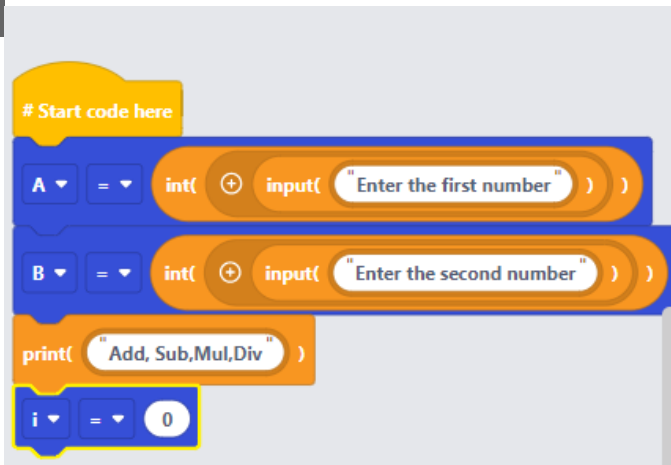
```
1 #Start code here
2 A = int(input("Enter the first number"))
3 B = int(input("Enter the second number"))
4 print("Add, Sub,Mul,Div")
5
```

Program Step 7:-



The image shows the Scratch variable palette on the left with categories: Imports, Variables, Statements, and Text. On the right, the 'Create variable...' button is highlighted. Below it, a variable block for 'i' is shown, set to 0. The variable block is highlighted with a red rectangle.

Create another variable 'i' to store the operation user wants the calculator to perform

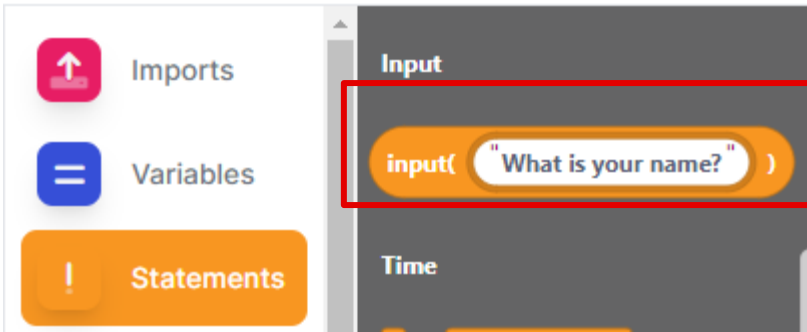


The image shows a Scratch code area with the following blocks: a yellow 'Start code here' block, two blue 'Set variable to' blocks for 'A' and 'B' (both set to 'int(+ input("Enter the first number"))'), a blue 'Set variable to' block for 'i' (set to 0), and a blue 'Print' block with the text 'Add, Sub, Mul, Div'.

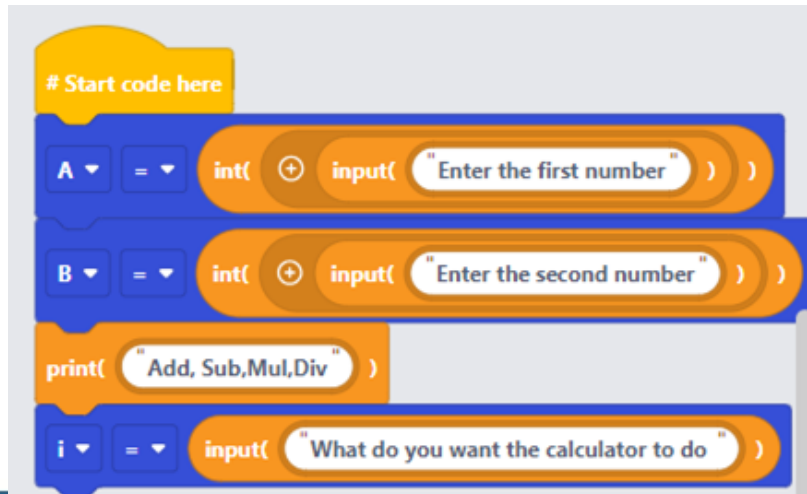
Code

```
1 #Start code here
2 A = int(input("Enter the first number"))
3 B = int(input("Enter the second number"))
4 print("Add, Sub,Mul,Div")
5 i = 0
6
```

Program Step 8:-



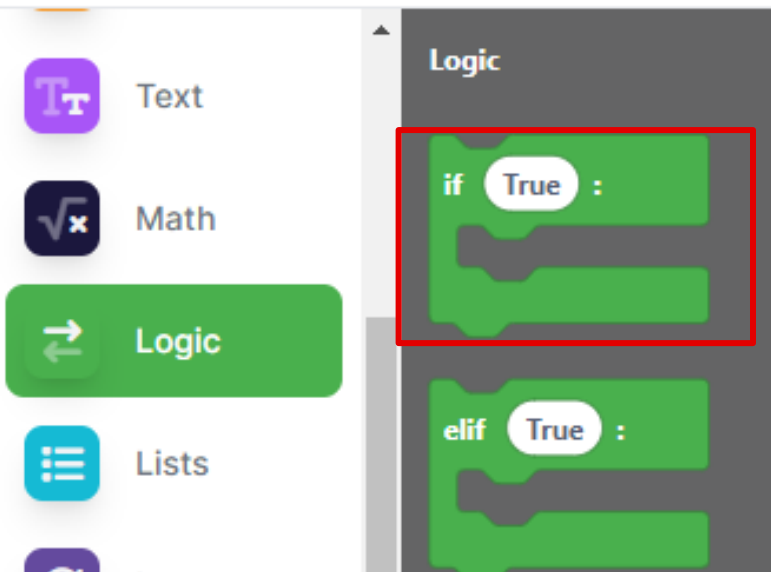
Add an input block to take input from the user for the operation user wants to do



Code

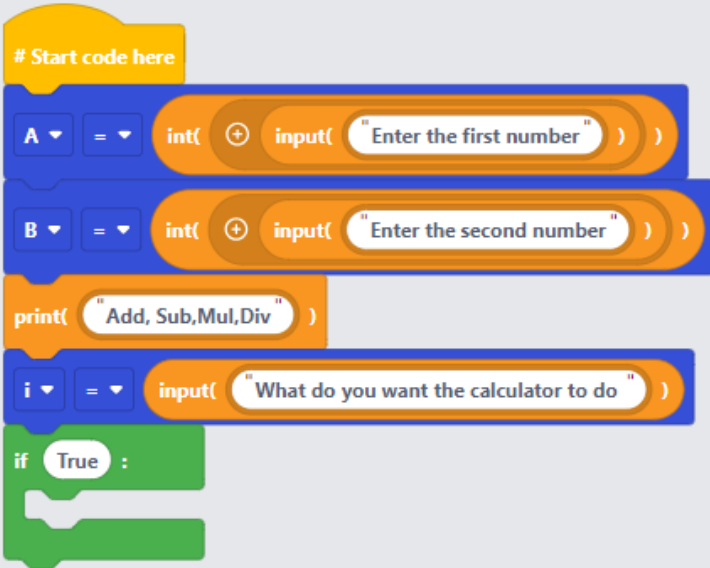
```
1 #Start code here
2 A = int(input("Enter the first number"))
3 B = int(input("Enter the second number"))
4 print("Add, Sub,Mul,Div")
5 i = input("What do you want the calculat
6
```

Program Step 9:-



Add if statement from the logic block

Program Step 10:-



Code

```
1 #Start code here
2 A = int(input("Enter the first number"))
3 B = int(input("Enter the second number"))
4 print("Add, Sub,Mul,Div")
5 i = input("What do you want the calculator to do ")
6 if True:
7     pass
8
```

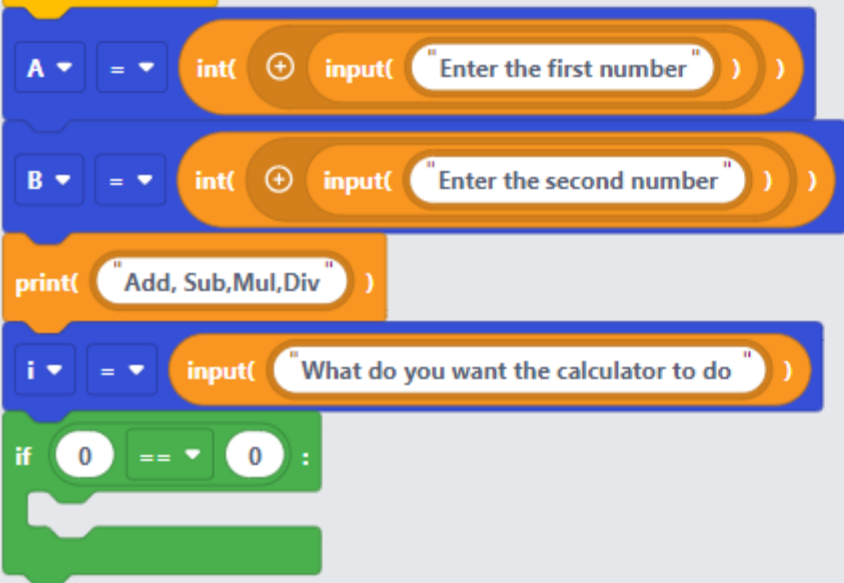
Program Step 11:-

The image shows the Scratch Logic block editor. On the left is a sidebar with categories: Text, Math, Logic (highlighted), Lists, Loops, Definitions, Turtle, and Graphics. The main workspace shows a Logic block with an 'if' statement. The 'if' block contains a 'True' condition. Below it is an 'elif' block with a 'True' condition. At the bottom is an 'else:' block. A red box highlights the bottom of the 'else:' block, showing a comparison block with the value '0' on the left, an equals sign '==' in the middle, and another '0' on the right.

Add equal to from the logic block

Program Step 12:-

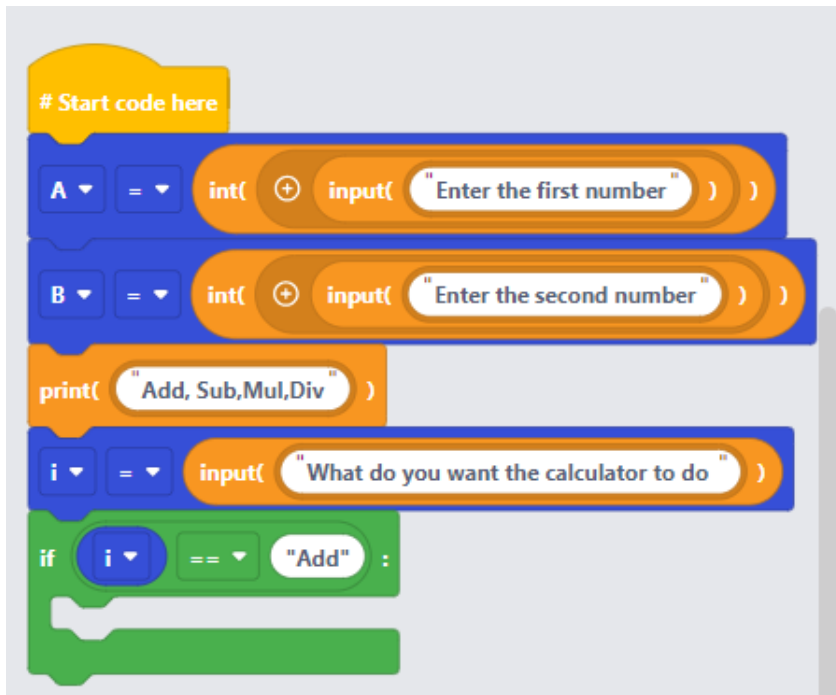
Start code here



Code

```
1 #Start code here
2 A = int(input("Enter the first number"))
3 B = int(input("Enter the second number"))
4 print("Add, Sub,Mul,Div")
5 i = input("What do you want the calculator to do ")
6 if 0 == 0:
7     pass
8
```


Program Step 13:-

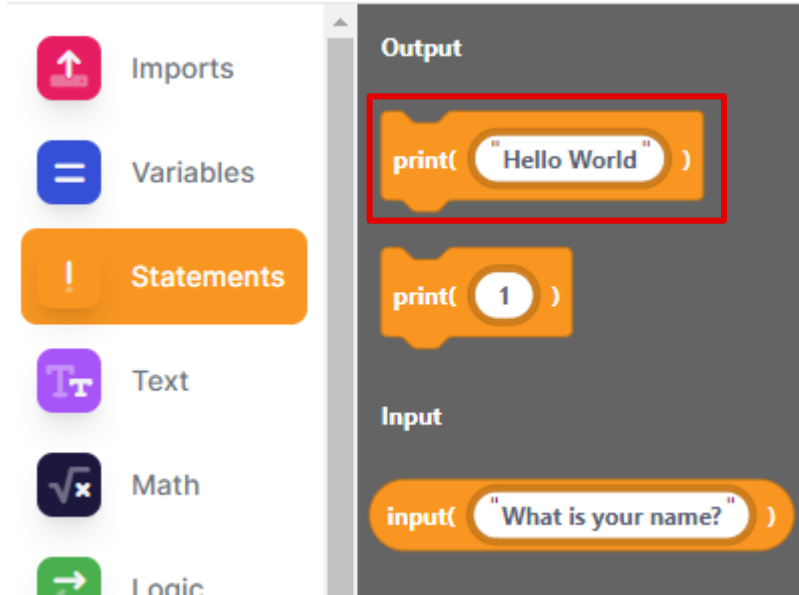


Code

```
1 #Start code here
2 A = int(input("Enter the first number"))
3 B = int(input("Enter the second number"))
4 print("Add, Sub,Mul,Div")
5 i = input("What do you want the calculator to do ")
6 if i == "Add":
7     pass
8
```

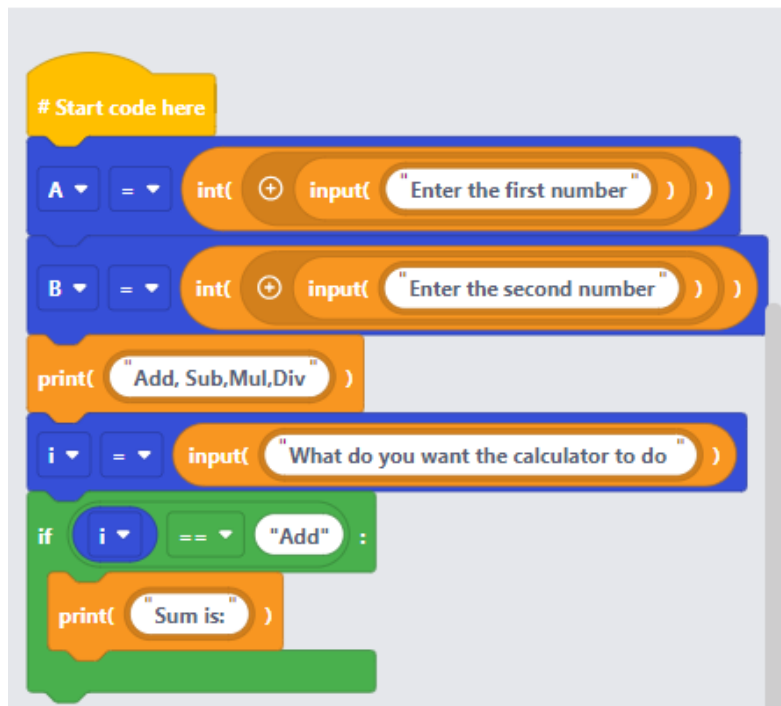
Add the 'i' value block from the variables to the left of the comparison block to compare the input given by user

Program Step 14:-



Take the print “Hello World” block from the statement

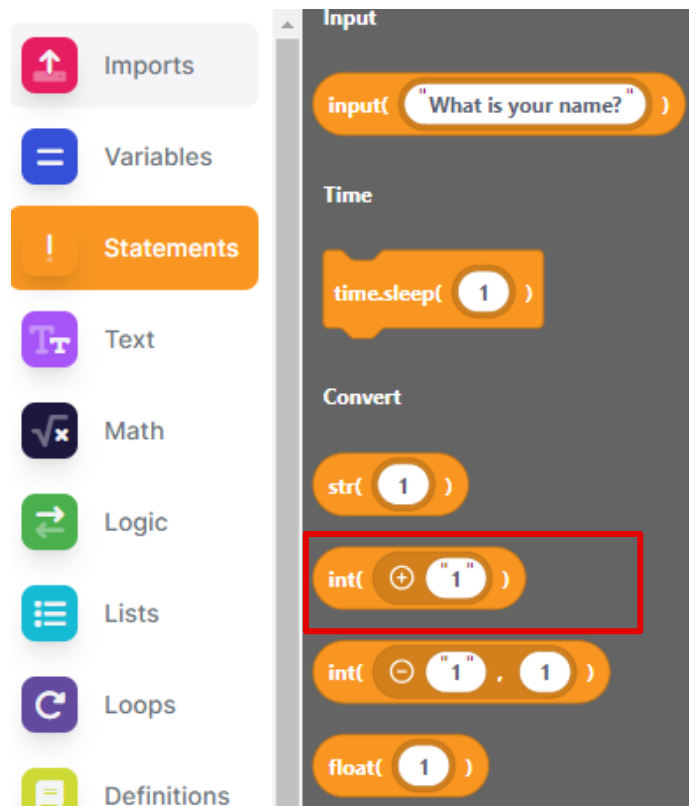
Program Step 15:-



Code

```
1 #Start code here
2 A = int(input("Enter the first number"))
3 B = int(input("Enter the second number"))
4 print("Add, Sub,Mul,Div")
5 i = input("What do you want the calculator to do ")
6 if i == "Add":
7     print("Sum is:")
8
```

Program Step 16:-



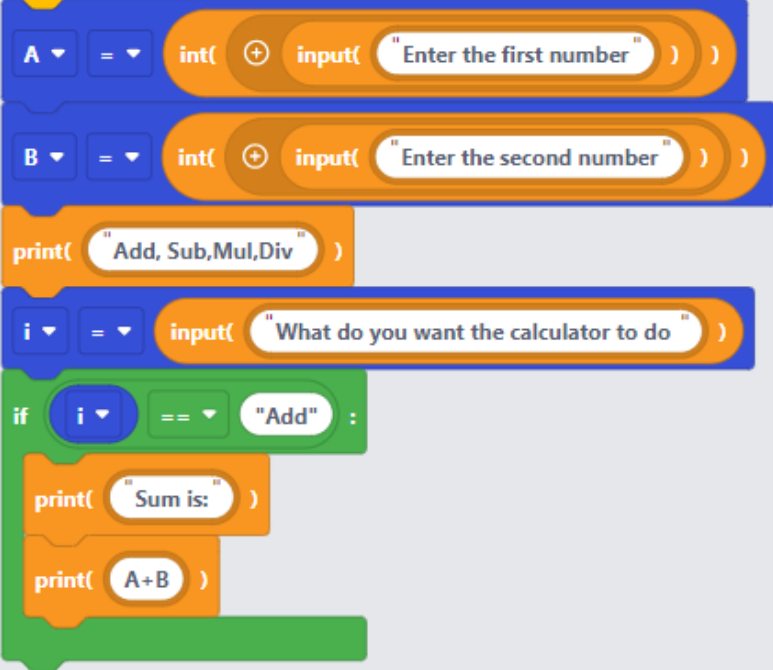
The image shows a block palette on the left and a workspace on the right. The palette includes categories: Imports, Variables, Statements, Text, Math, Logic, Lists, Loops, and Definitions. The workspace contains the following blocks:

- Input:** `input("What is your name?")`
- Time:** `time.sleep(1)`
- Convert:**
 - `str(1)`
 - `int(+ "1")` (This block is highlighted with a red rectangle)
 - `int(- "1" , 1)`
 - `float(1)`

Take the int block from the statement

Program Step 17:-

Start code here



Code

```
1 #Start code here
2 A = int(input("Enter the first number"))
3 B = int(input("Enter the second number"))
4 print("Add, Sub,Mul,Div")
5 i = input("What do you want the calculator to do ")
6 if i == "Add":
7     print("Sum is:")
8     print(A+B)
9
```

Program Step 18:-

```
# Start code here

A = int( input( "Enter the first number" ) )
B = int( input( "Enter the second number" ) )

print( "Add, Sub,Mul,Div" )

i = input( "What do you want the calculator to do" )

if i == "Add" :
    print( "Sum is:" )
    print( A+B )
```



```
if i == "Sub" :
    print( "Difference" )
    print( A-B )

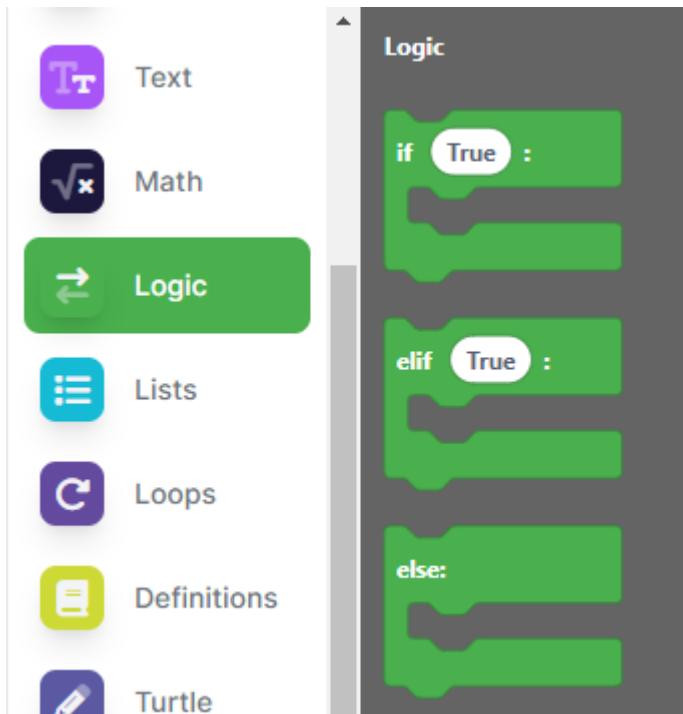
if i == "Mul" :
    print( "Product is" )
    print( A*B )

if i == "Div" :
    print( "Quotient is" )
    print( A/B )
```

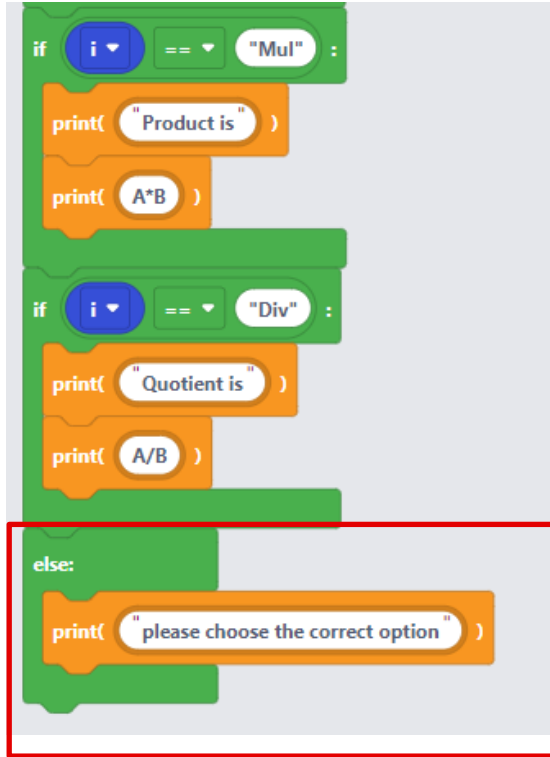
The same way make the code for all the other three operations

Program Step 19:-

Take the Else block from the logic



Program Step 20:-



```

7   print("Sum is:")
8   print(A+B)
9   if i == "Sub":
10    print("Difference")
11    print(A-B)
12  if i == "Mul":
13    print("Product is")
14    print(A*B)
15  if i == "Div":
16    print("Quotient is")
17    print(A/B)
18  else:
19    print("please choose the correct option")
20
    
```


Program Step :-

```

1  #Start code here
2  A = int(input("Enter the first number"))
3  B = int(input("Enter the second number"))
4  print("Add, Sub,Mul,Div")
5  i = input("What do you want the calculator to do ")
6  v if i == "Add":
7      print("Sum is:")
8      print(A+B)
9  v if i == "Sub":
10     print("Difference")
11     print(A-B)
12 v if i == "Mul":
13     print("Product is")
14     print(A*B)
15 v if i == "Div":
16     print("Quotient is")
17     print(A/B)
18 v else:
19     print("please choose the correct option")
20

```

Output

Powered by  trinket

Enter the first number 45

Enter the second number 56

Add, Sub,Mul,Div

What do you want the calculator to do Add

Sum is:

101

Powered by  trinket

Enter the first number 56

Enter the second number 67

Add, Sub,Mul,Div

What do you want the calculator to do Sub

Difference

-11

Powered by  trinket

Enter the first number 78


Enter the second number 67

Add, Sub,Mul,Div

What do you want the calculator to do Mul

Product is

5226

Powered by  trinket

Enter the first number 67

Enter the second number 34

Add, Sub,Mul,Div

What do you want the calculator to do Div

Quotient is

1.97058823529

ACTIVITY SHEETS

Question 1:

If x be the input given by user, how do we compare it with an operation?

- A. `x="add"`
- B. `x=="add"`
- C. `x==add`
- D. `x=add`

Question 2:

What does an int block do?

- A. Check if the value is number
- B. check if the value is decimal
- C. convert the value to integer
- D. convert the value to float

Question 3:

How do we add values of two variables?

- A. $A+=B$
- B. $A+B$
- C. $A++$
- D. $A=+$

Question 4:

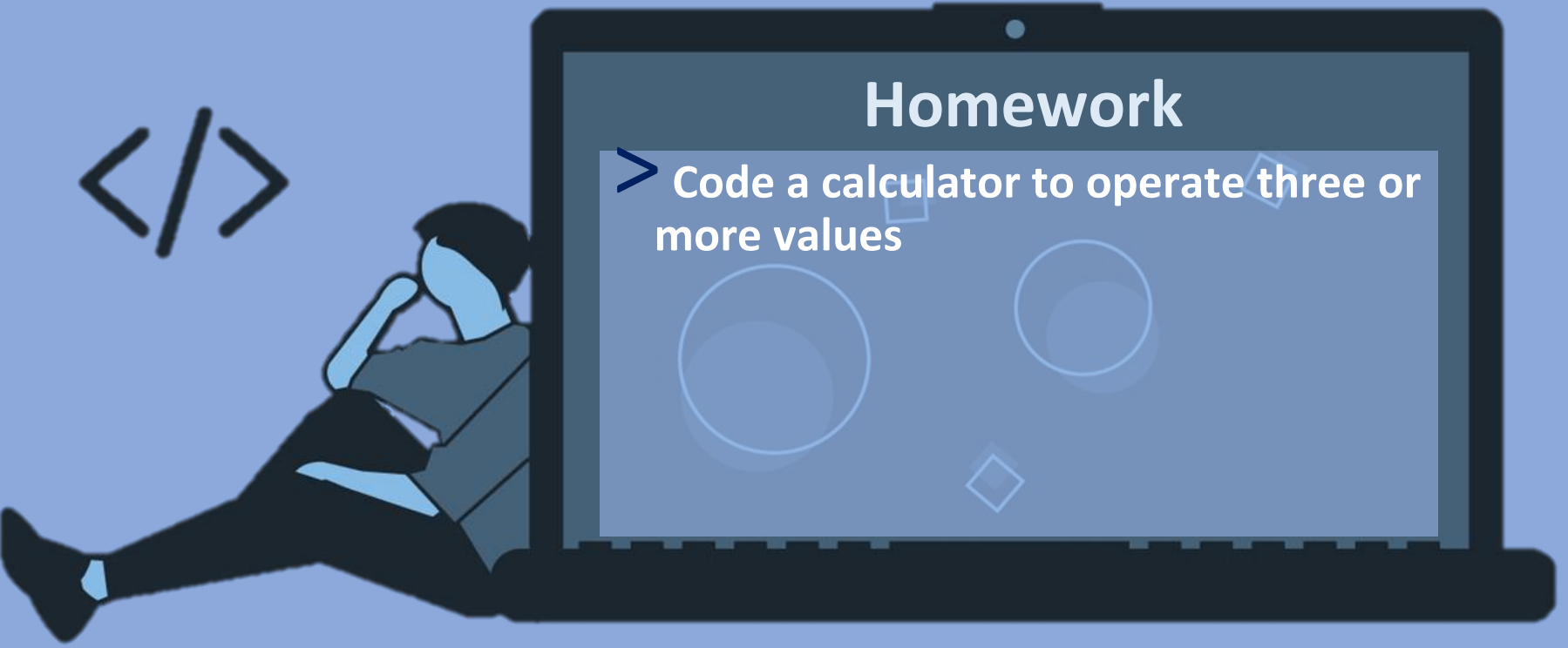
Which statement is correct?

- A. `if x=10:`
- B. `if x==10:`
- C. `if x+=10:`
- D. `if x>=10:`

Question 5:

If we want the code to behave in a way that if one condition isn't correct it should follow the other command, what do we do?

- A. add an else condition
- B. add an if else condition
- C. add elif condition
- D. add while loop



Homework

- > Code a calculator to operate three or more values