Snap! Computer Programming Tutorial: Loops

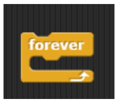
By: Matthew Jourden

Brighton High School

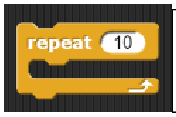
Brighton, MI

Loops

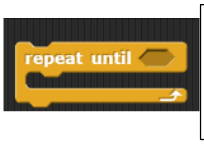
Loops: allow the user to do a serious of commands multiple times given certain conditions.
Snap! Has the following Loop Types



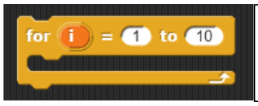
Forever Loop: Will run the Code forever until the user presses the Red Stop button in the top right corner



Repeat Loop: Will run as long as the condition is true, once the condition is false the loop will be . Think of the repeat like an If/Then, but instead of the if it is replaced with the repeat which turns



Repeat Until Loop: Will run as long as the condition is false, once the condition turns true then the loop will be exited. Think of the repeat like an If/Then, but instead of the if it is replaced with the repeat which turns the if/then into a loop



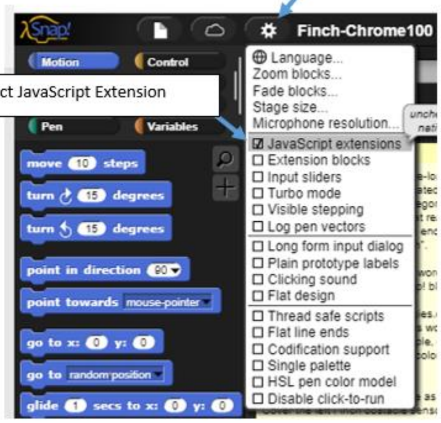
For Loop: Repeats a loop a known number of times.

Setup Java Script Extension

Script allows user to input/output text and graphic images

1. Select Settings Gear at the Top Left Of Screen > Select Check Mark next to Java Script Extentions

1, Select Settings Gear

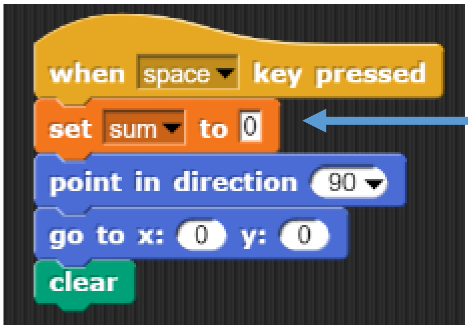


2. Select JavaScript Extension

Part 1: For Loop Adding multiple numbers

Objective have the user tell program how many numbers they would like to add

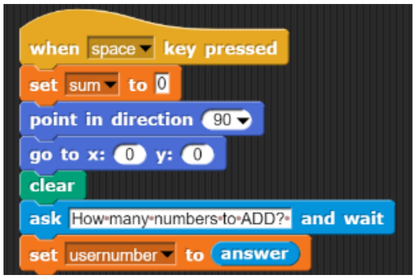
1. Create the following varibles: sum and usernumber
2. Place the follow code to setup the sprite and initialize values



```
when space key pressed
set sum to 0
point in direction 90
go to x: 0 y: 0
clear
```

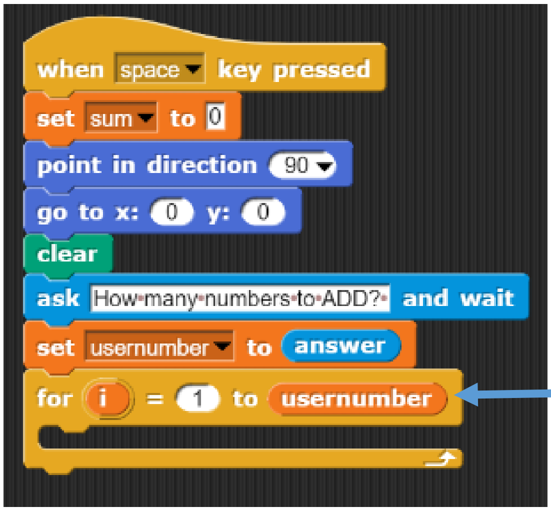
Resets Sum = 0 each time through the loop

3. Ask the User how many numbers they would like to add
   a. **Sensing Menu** > Drag and Drop **Ask** Command > Change text to “How Many numbers to ADD?”
   b. **Variables** Menu > Drag and Drop **Set** to Command > Set Variable to usernumber to answer variable from the **Sensing** Menu



```
when space key pressed
set sum to 0
point in direction 90
go to x: 0 y: 0
clear
ask How many numbers to ADD? and wait
set usernumber to answer
```

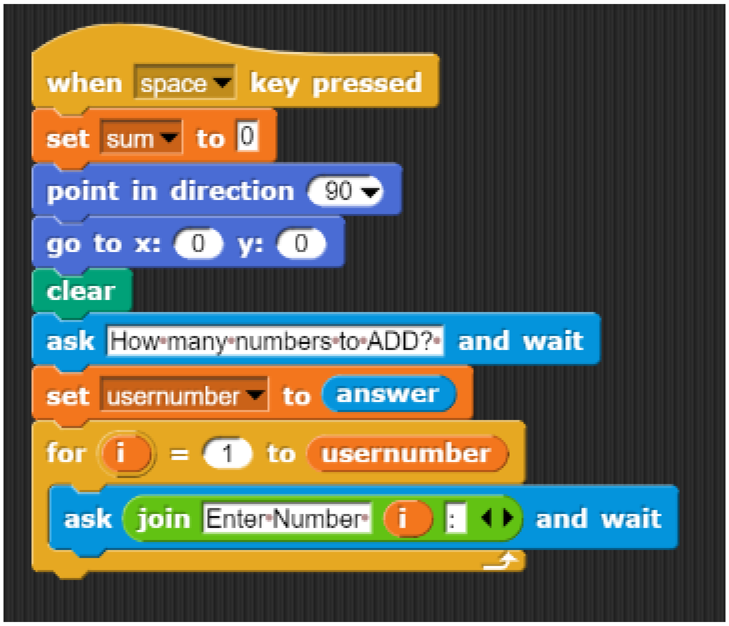
4. For Loop
   a. Select **Control** Menu > Drag and Drop **For Loop** Commond > Change end number to Variable usernumber Code should look as follows



```
when space key pressed
set sum to 0
point in direction 90
go to x: 0 y: 0
clear
ask How many numbers to ADD? and wait
set usernumber to answer
for i = 1 to usernumber
```

For Loop will from 1 to whatever the user enters counting up by 1 each time through the loop

b. Input: Select **Sensing** Menu > Drag **Ask** Command > Place inside the **For Loop** > Select **Operators** Menu > Drag and Drop **Join** Command into the **Ask** Command Body > Change the wording in the first box of the join to Enter Number > Change the wording in the second box to the variable I from the Foor (Hint Duplicate the For Loop Command > Drag the I from the For Loop to place into the Join Command) > Add a 3rd box to the join Command > Place a : in the space



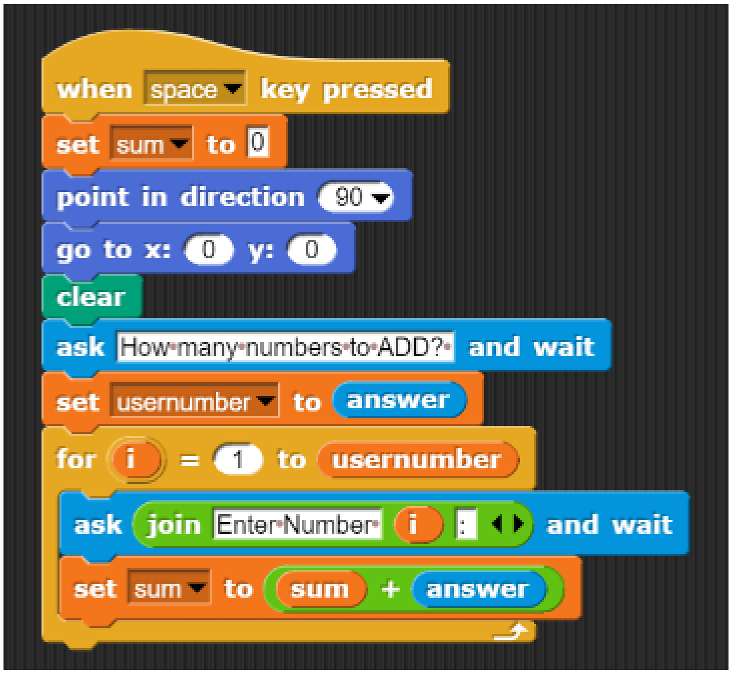
```
when space key pressed
set sum to 0
point in direction 90
go to x: 0 y: 0
clear
ask How·many·numbers·to·ADD?· and wait
set usernumber to answer
for i = 1 to usernumber
    ask join Enter·Number· i : and wait
```

c. Forumla for adding multiple numbers

sum = sum + answer

NOTE: we will use answer here because we will be overwriting it each time through the loop and do not have need to store the value the user inputs.

a. **Variable** Menu > Drag and Drop **Set** To Command into the **For Loop** after the **Ask** Commond > **Set Variable** to sum > Select **Operating** Menu > Drag and Drop **+** operator into the 2nd box of the **Set** Command > Place **Variable sum** in the 1st box of the **Add** Commond > Place answer variable in the 2nd box of the **Add** Command

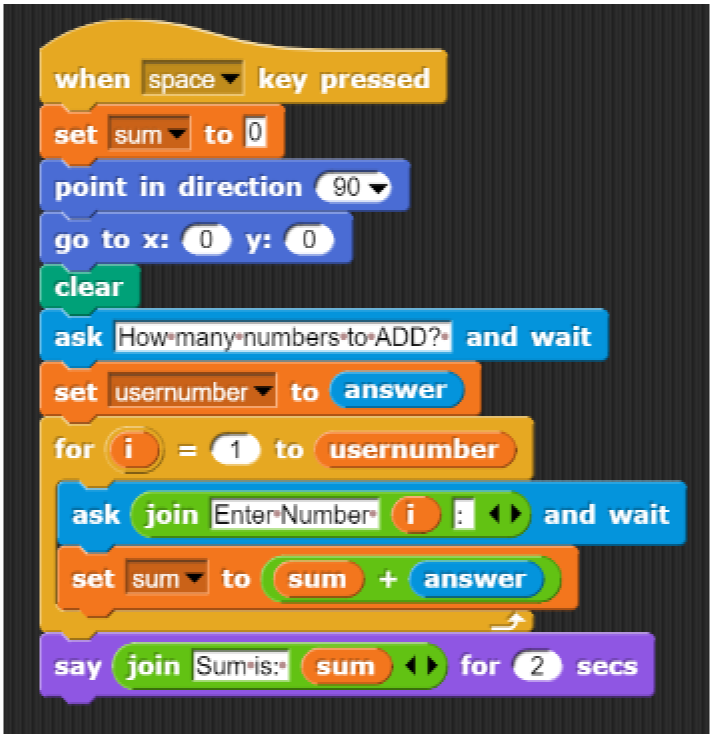


```
when space key pressed
set sum to 0
point in direction 90
go to x: 0 y: 0
clear
ask How·many·numbers·to·ADD?· and wait
set usernumber to answer
for i = 1 to usernumber
    ask join Enter·Number· i : and wait
    set sum to sum + answer
```

How the Loop Works: Overwriting sum with what sum equally previously in the loop to the user entry which is est to Answer. See Video Reference for explantion of the loop

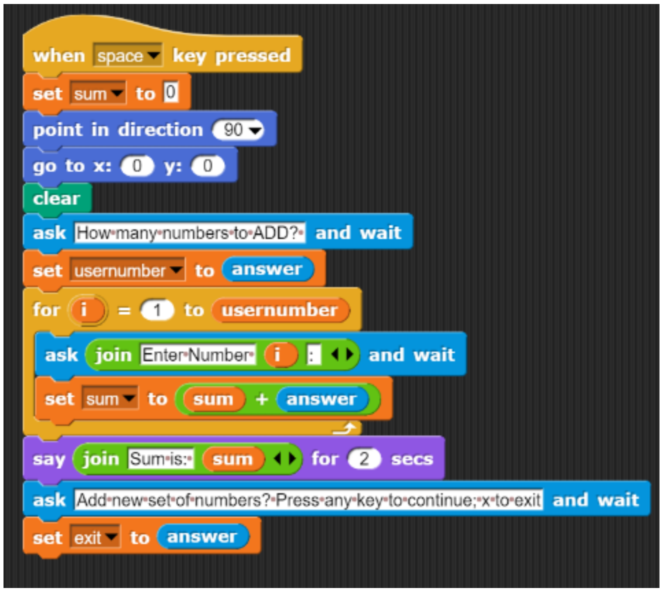
Times Thru the Loop User Enters to Add 4 Numbers	Variable Answer User Entery for number to add +	Sum = 0
1	2 =	2
2	5	7
3	3	10
4	7	17

5. Output Answer
   a. Select **Looks** Menu > Drag and Drop **Say** Command after the **For Loop** > Select **Operators** Menu > Drag and Drop **Join** Command > Change 1st Box of **Join** command to Sum is: >> Change 2nd Box of **Join** Command to the Variable sum



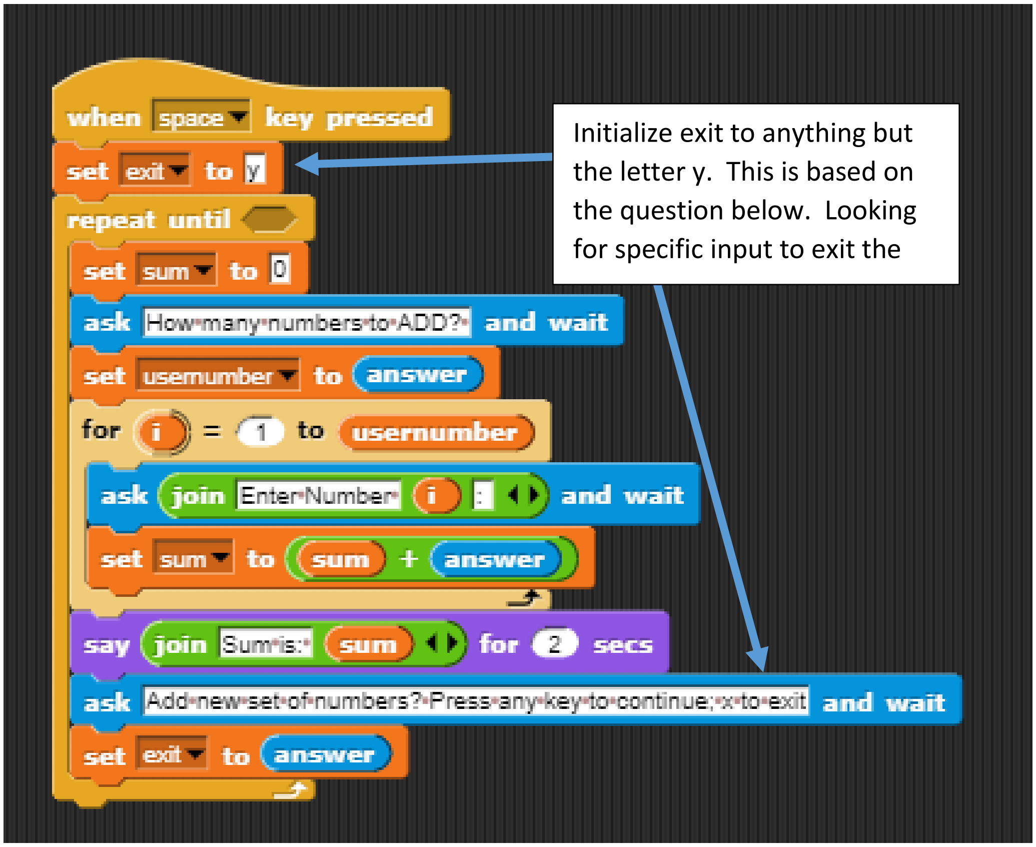
```
when space key pressed
set sum to 0
point in direction 90
go to x: 0 y: 0
clear
ask How·many·numbers·to·ADD?· and wait
set usernumber to answer
for i = 1 to usernumber
    ask join Enter·Number· i : and wait
    set sum to sum + answer
say join Sum·is:· sum for 2 secs
```

6. Asking the User if they would like to Add a new set of numbers
    a. Create a Variable called exit
    b. Select **Sensing** Menu > Drag and Drop **Ask** Command after the Say Sum is Output at the end of the program > Change the **Ask** command to Add More Numbers? Any Key to Continue; x to exit
    c. Select **Variables** Menu > Drag and Drop **Set** to Command after the **Ask** Command > Change variable to exit > Place answer variable in the to box of the **Set** Command



```
when space key pressed
set sum to 0
point in direction 90
go to x: 0 y: 0
clear
ask How many numbers to ADD? and wait
set usernumber to answer
for i = 1 to usernumber
    ask join Enter Number i : and wait
    set sum to sum + answer
say join Sum is: sum for 2 secs
ask Add new set of numbers? Press any key to continue; x to exit and wait
set exit to answer
```

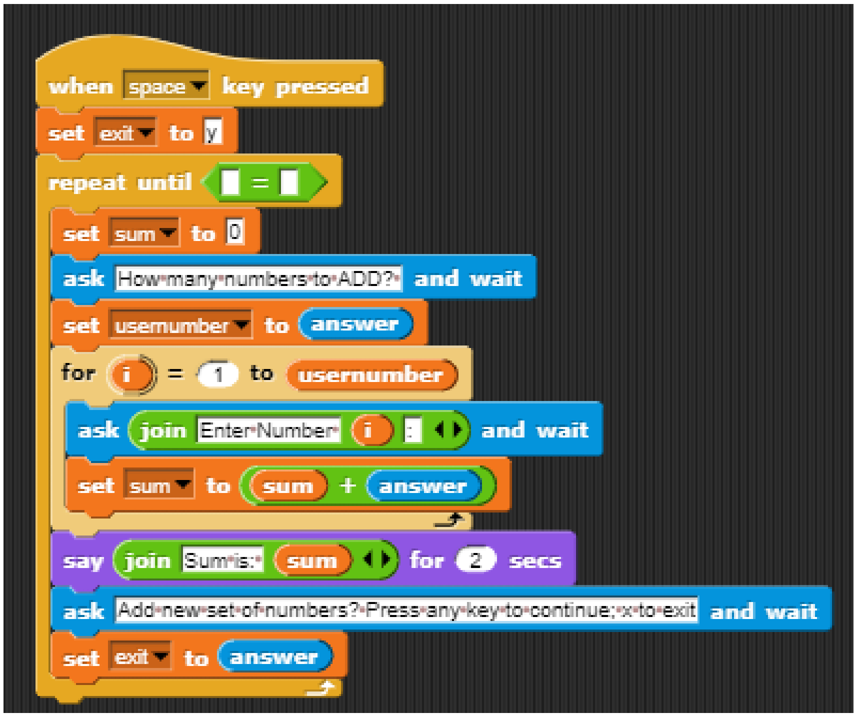
    d. Select **Control** Menu > Drag and Drop **Repeat Until** > Place all of the code into the **Repeat Until** EXCEPT for the When Key is Pressed start command > Place a **Set** to Command between **When Space Key pressed** and **repeat unitl** loop > **Set** Variable to exit and to y. This loop will work until the user enters x which will make the condition false then dropping the user out of the loop and initialize exit to anything but the letter x. Otherwise the software will retain the value of x from the previous time through the program, so the program will only run 1 time or until the variable exit gets changed to anything but the letter x



```
when space key pressed
set exit to y
repeat until
    set sum to 0
    ask How many numbers to ADD? and wait
    set usernumber to answer
    for i = 1 to usernumber
        ask join Enter Number i : and wait
        set sum to sum + answer
    say join Sum is: sum for 2 secs
    ask Add new set of numbers? Press any key to continue; x to exit and wait
    set exit to answer
```

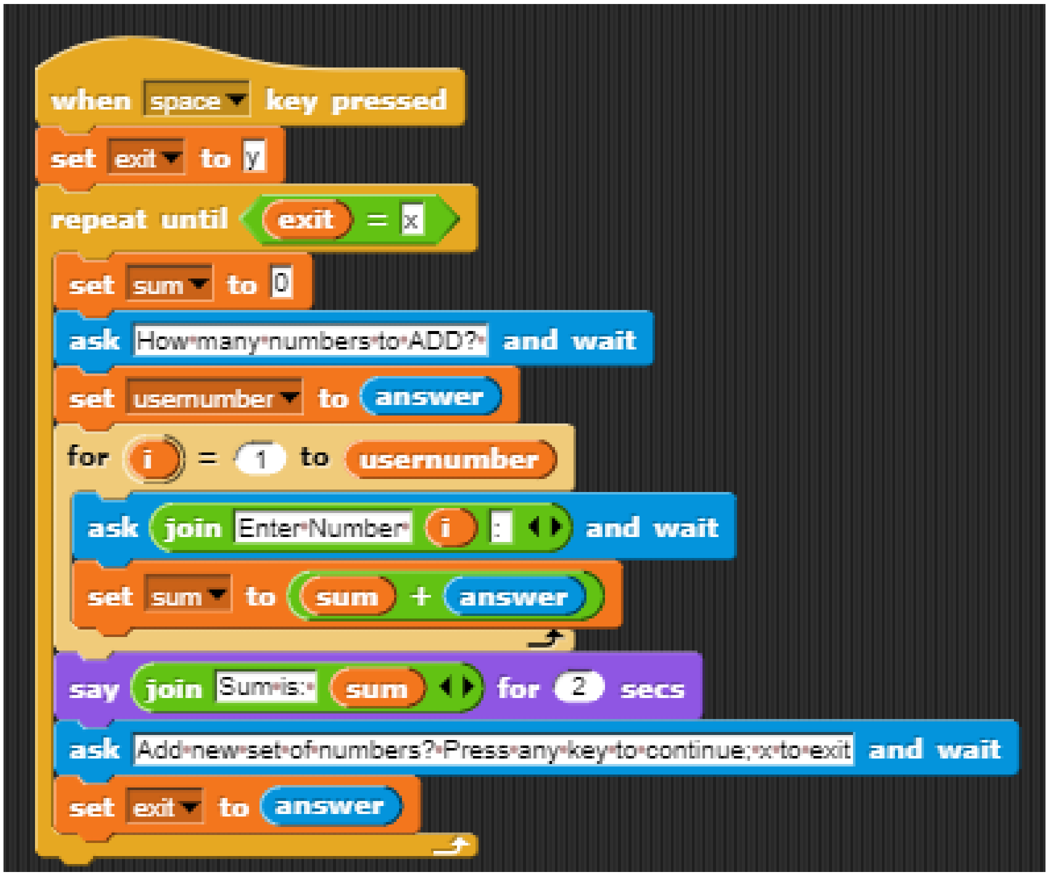
Initialize exit to anything but the letter y. This is based on the question below. Looking for specific input to exit the

e. Select **Operators** Menu > Drag and Drop **=** Comparison into the space provided in the **Repeat Until Loop**



```
when [space] key pressed
set [exit] to [y]
repeat until <[ ] = [ ]>
    set [sum] to [0]
    ask [How many numbers to ADD? ] and wait
    set [usernumber] to (answer)
    for (i) = (1) to (usernumber)
        ask (join [Enter Number ] (i) [:]) and wait
        set [sum] to ((sum) + (answer))
    say (join [Sum is: ] (sum)) for (2) secs
    ask [Add new set of numbers? Press any key to continue; x to exit] and wait
    set [exit] to (answer)
```

f. Add variable exit to the 1st box > Add letter x in the 2nd box



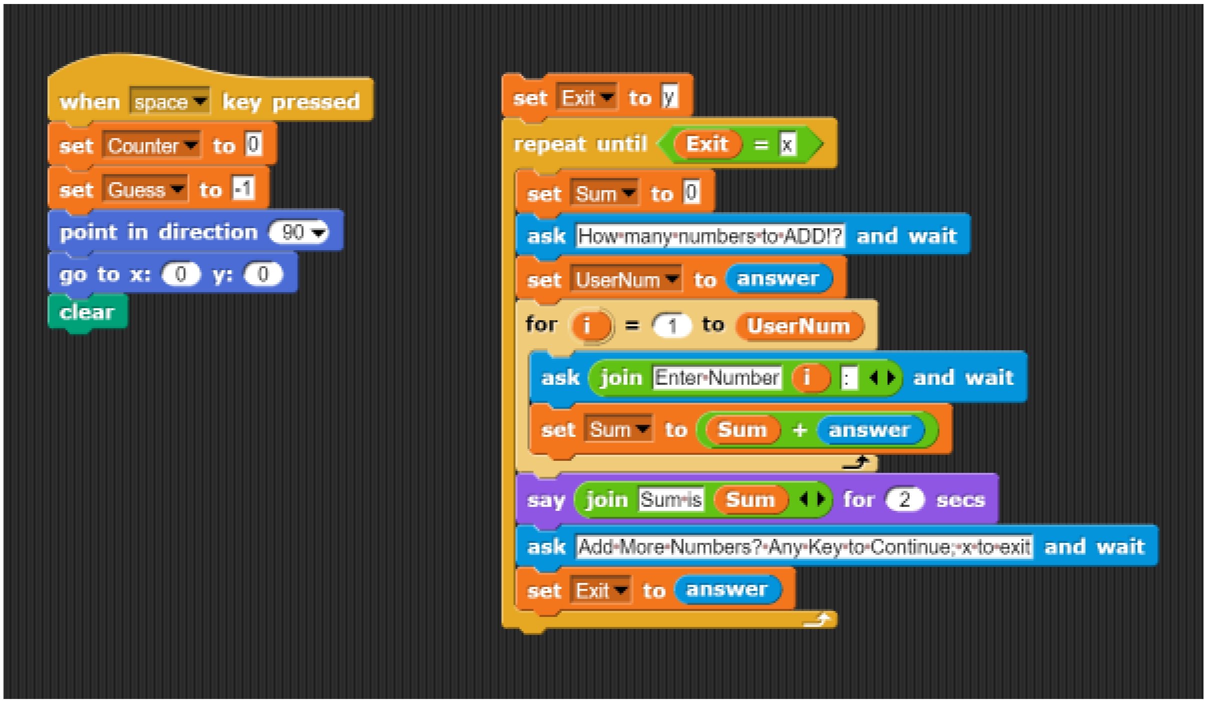
```
when [space] key pressed
set [exit] to [y]
repeat until <(exit) = [x]>
    set [sum] to [0]
    ask [How many numbers to ADD? ] and wait
    set [usernumber] to (answer)
    for (i) = (1) to (usernumber)
        ask (join [Enter Number ] (i) [:]) and wait
        set [sum] to ((sum) + (answer))
    say (join [Sum is: ] (sum)) for (2) secs
    ask [Add new set of numbers? Press any key to continue; x to exit] and wait
    set [exit] to (answer)
```

7. Test the Program
8. Remove the Set Exit to Y Line of code > Test the Code twice to see what happens.
   Place the line of code back into the code and Test again

Part 2: Repeat and Repeat Until Loops

Remove the For Loop, Repeat Until and output of Sum off to the right of the main code. Keep the Sprite Setup in place. DO NOT DELETE this code.

Program Should look as follows



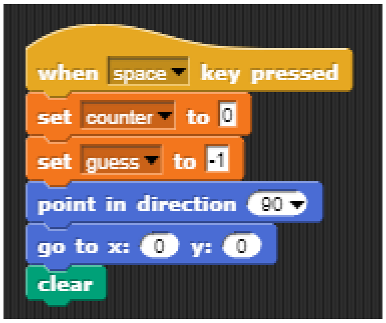
```
when space key pressed
set Counter to 0
set Guess to -1
point in direction 90
go to x: 0 y: 0
clear
```

```
set Exit to y
repeat until Exit = x
    set Sum to 0
    ask How many numbers to ADD!? and wait
    set UserNum to answer
    for i = 1 to UserNum
        ask join Enter Number i : and wait
        set Sum to Sum + answer
    say join Sum is Sum for 2 secs
    ask Add More Numbers? Any Key to Continue; x to exit and wait
    set Exit to answer
```

This program will choose a number at random and then have the user take no more than 3 guesses (good or bad) to try and guess the computer's random number 1 thru 10,

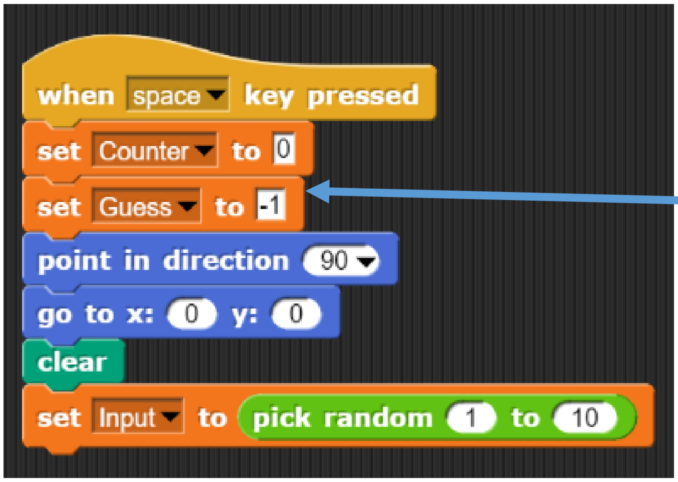
1. Open Birdbrain Snap! From the Desktop > use Chrome Browser
2. Go to Controls > drag and drop the **When "space" key is pressed**. Place this command next to previous program
3. Setting a counter. Since we are giving the user a limited number of guesses we need to setup a variable that will count the number of guesses.
    a. Tab Variable > **Make Variable** > call it **counter** > Ok
    b. Tab Variable > **Make Variable** > call it **guess**> Ok
    Note: The variables should appear on the list
    c. Drag **Set command** onto the screen > Change the information to the following
        i. Set counter to 0
        ii. Set guess to 1
4. We will also reposition the Sprite to the center of the screen. Place the first two commands from the first program below the Set Counter command, then place a Clear command below this to clear the screen of any drawings or text.

Program should look as follows



```
when space key pressed
set counter to 0
set guess to -1
point in direction 90
go to x: 0 y: 0
clear
```

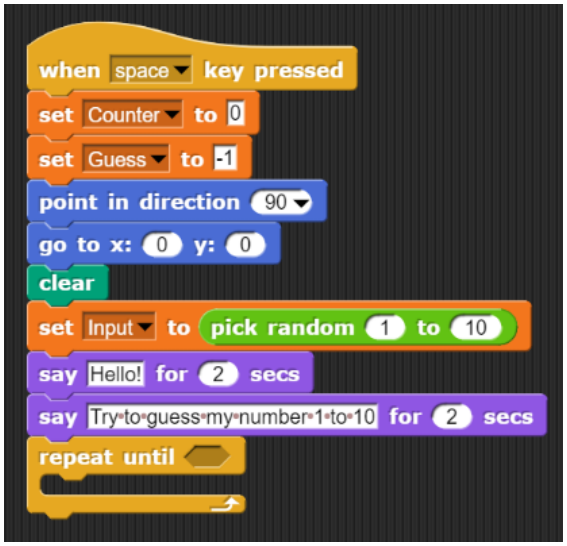
5. Next we will have the computer choose a random number and store it.
    a. Create a new variable called input
    b. Drag Set “variable” to 0 command below the clear command
    c. Click the down arrow and change the variable to input
    d. Goto Operators Menu > drag Pick Random 1 to 10 command and drop in the space of the Set ‘Input” to 0 in the 0 space. Should look as follows



```
when space key pressed
set Counter to 0
set Guess to -1
point in direction 90
go to x: 0 y: 0
clear
set Input to pick random 1 to 10
```

Guess is initialized to -1 because it is outside of the range of guesses. This will allow the user to enter the loops that will be created later

6. Next we will have the sprite talk to us using the Say and Ask commands
    a. Tab **Looks** > drag the **Say command** below the Set random # line
        i. Type “Hello” in the command box
    b. Place a 2nd **Say** command wait for 2 secs command
        i. Type “Try to guess my number 1 to 10”
7. Next we will place a loop to run until the number is either guessed or we reach 3 guesses
    a. Tab **Control** > drag and drop the **Repeat Until** command. This command will allow us to place a boolean (True or False) function to check if we reach the max number of guesses or we guessed the number. Boolean is a comparison option I.E =, <, >, or, and
    b. Tab **Operators** > drag the ___________ Or _____________ operator in the blank space in the **Repeat Until** command. This is used when we only need one of the two conditions to be true in order to carry out the lines of code following
    c. Tab Operators > drag _________ = ___________ into both blank spaces of the Or statement
    Program Looks as follows



```
when space key pressed
set Counter to 0
set Guess to -1
point in direction 90
go to x: 0 y: 0
clear
set Input to pick random 1 to 10
say Hello! for 2 secs
say Try to guess my number 1 to 10 for 2 secs
repeat until
```

d. Next we need to place the qualifying information for each of these

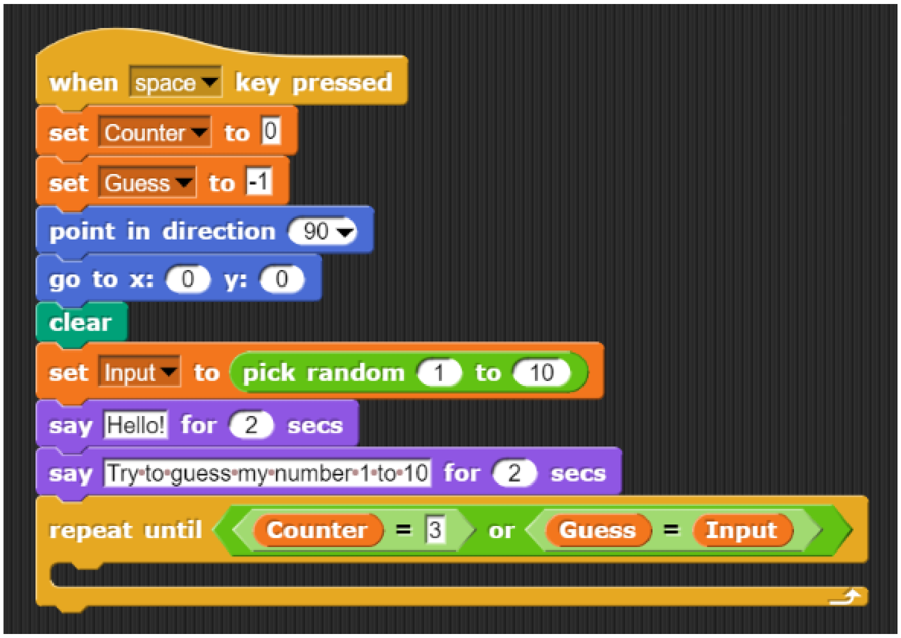
i. In the first box we want to check to see if the counter reaches the max 3 guesses

1. Tab **Variable** > drag **Counter** in one of the blank boxes and set the other blank box to 3

ii. In the second box we want to check to see if the inputted answer is equal to the guess

1. Tab **Variable** > drag variable **guess** and **input** (which will be the users guess)

Program Look as follows



```
when space key pressed
set Counter to 0
set Guess to -1
point in direction 90
go to x: 0 y: 0
clear
set Input to pick random 1 to 10
say Hello! for 2 secs
say Try to guess my number 1 to 10 for 2 secs
repeat until < <Counter = 3> or <Guess = Input> >
```

e. Next we will use a combination of **Repeat Until** loops and **If** statements to check to see if the user has guessed the correct number and/or has inputted bad entry.

If Statements: allow the user to make comparisons using boolean functions to check if an answer falls in a certain range or is equal to something.

Repeat Until Loop: allows the user to go through a loop a specfied number of times or until a compariosn (<, = or >) becomes FALSE.

i. Tab **Sensing** > drag and drop the **Ask** command in the While Statement > Type “Pick a Number 1 to 10” (The Ask command asks the user a question and stores the input in the variable answer that is located in the Sensing Menu)

ii. Tab **Variable** > Drag **Set** command below **ask** command > Change to **set guess to answer**

iii. Check to make sure the users input falls in the range of 1 to 10

1. Tab Control > drag **Repeat Until** statement place it below the **set guess to answer line**

2. Tab **Operators** > drag and drop the **AND** statement into the **If** statement blank space. An **AND** statement is used when we need both conditions to be true in order to carry out the following line(s) of code

3. In the first blank space on the Or statement place a ___ > _____ command

a. Tab Sensing > first box place variable **guess**

b. Tab Sensing second box place the value 0. The value needs to be 0 since 1 is an option for the user to enter and we are using the > option

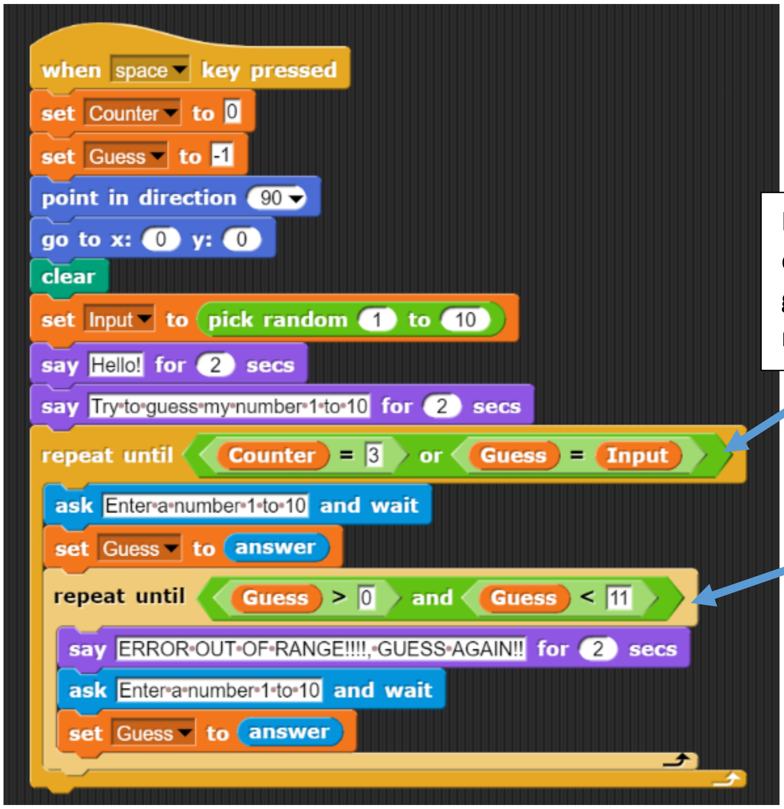
4. In the second blank space of the AND statement place the < command from the

a. In the first box place the variable **guess** command

b. In the second box type in the value 11

5. Within this **Repat Until** Loop add
   a. Statement that lets the user know there is an error
   b. Reasks them to input a number between 1-10
   c. Setting the new guess to answer

Program should look as follows



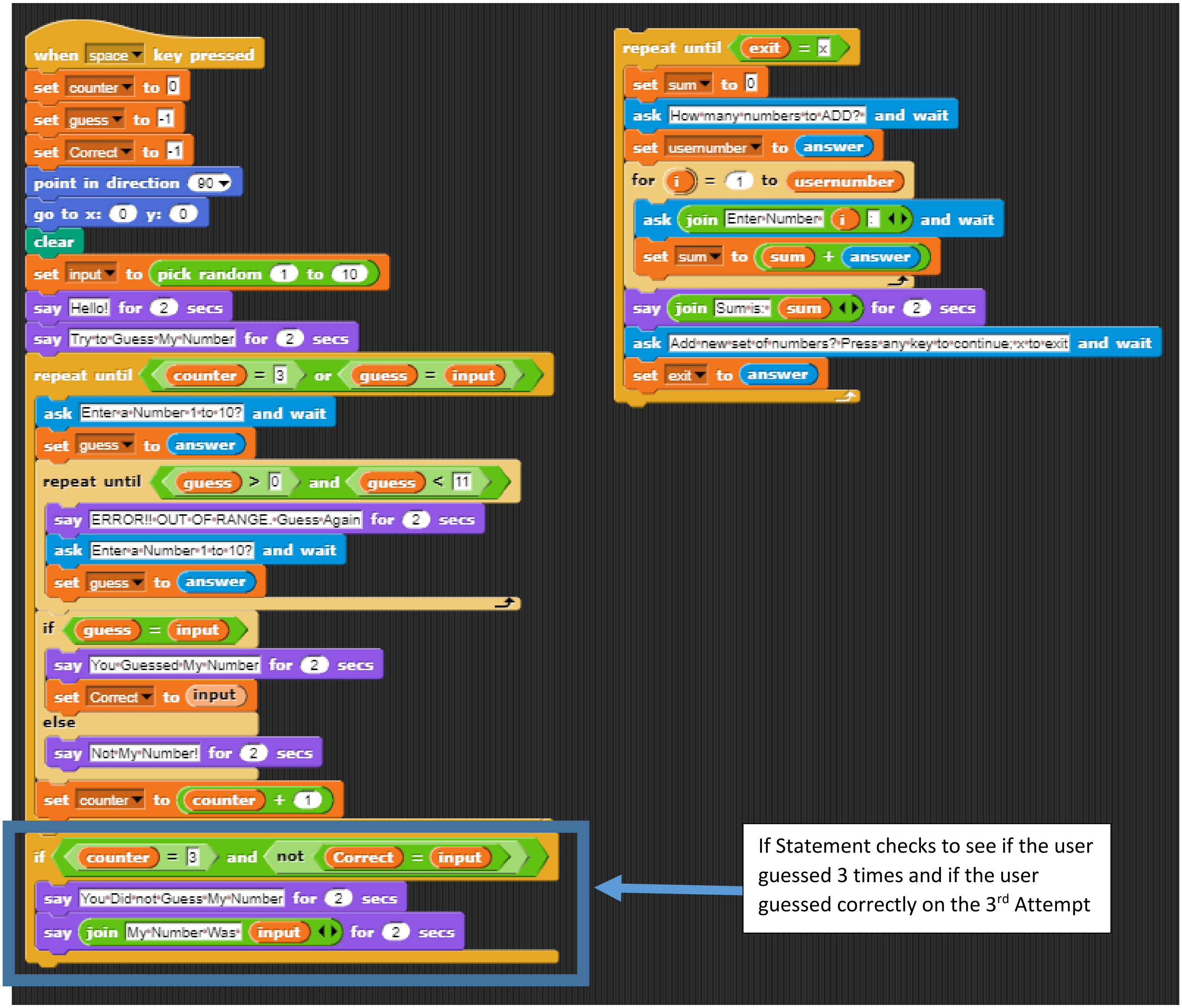
```
when space key pressed
set Counter to 0
set Guess to -1
point in direction 90
go to x: 0 y: 0
clear
set Input to pick random 1 to 10
say Hello! for 2 secs
say Try to guess my number 1 to 10 for 2 secs
repeat until Counter = 3 or Guess = Input
    ask Enter a number 1 to 10 and wait
    set Guess to answer
    repeat until Guess > 0 and Guess < 11
        say ERROR OUT OF RANGE!!!!, GUESS AGAIN!! for 2 secs
        ask Enter a number 1 to 10 and wait
        set Guess to answer
```

Loop to check to see the user does not exceed 3 incorrect guesses or the user guesses the random number

Loop to check to see if the user typed in the number between 1-10. If the number is not between 1-10 the user will be trapped in this loop until they enter a number 1-10.

f. Next we will test to see if the number is guessed which will allow us to exit the program or keep guessing. For this step we will do an If Else statement. This statement allows us to check for on thing and if it is false do something else
   i. Tab **Control** > drag and drop a **If Else** statement below (outside) the **Repeat Until** Loop that checks for good input, but inside the **Repeat Until** that checks the number of guesses and if the number was guessed correctly. Be careful not to place the **If Else** inside the **Repeat Until** loop that checks for bad input.
   ii. The **If** statement will see if the Input (random number) is equal to the users guess. If true Say " You guessed my Number!!!) and set a variable of Correct = Input so that way if the guess happens on the 3rd attempt it registers that the number was guessed. NOTE: Declare Correct = -1 at the beginning of the program
   iii. **Else** part of the statement, which would be a false answer (Input does not equal users guess) Say " Not my number try again.)
   iv. Since we only want to give the user a certain number of guesses we need to change our counter by 1 each time through the loop. At the end of the loop we will use the Change "counter" by 1 option. So when counter is equal to 3 we will fall out of the loop. If we do not change the counter we will have created an infinite loop with no possible way to exit other than crashing the program manually. (Snap! The Red Stop in the Sprite preview area)
   v. Add If Statement at the end to tell the user that the number was not guessed and what the number was.

Program Looks as follows



```
when space key pressed
set counter to 0
set guess to -1
set Correct to -1
point in direction 90
go to x: 0 y: 0
clear
set input to pick random 1 to 10
say Hello! for 2 secs
say Try to Guess My Number for 2 secs
repeat until < <(counter) = 3> or <(guess) = (input)> >
  ask Enter a Number 1 to 10? and wait
  set guess to answer
  repeat until < <(guess) > 0> and <(guess) < 11> >
    say ERROR!! OUT OF RANGE. Guess Again for 2 secs
    ask Enter a Number 1 to 10? and wait
    set guess to answer
  if <(guess) = (input)>
    say You Guessed My Number for 2 secs
    set Correct to input
  else
    say Not My Number! for 2 secs
  set counter to ((counter) + 1)
if < <(counter) = 3> and <not <(Correct) = (input)>> >
  say You Did not Guess My Number for 2 secs
  say join My Number Was (input) for 2 secs
```

```
repeat until <(exit) = x>
  set sum to 0
  ask How many numbers to ADD? and wait
  set usernumber to answer
  for (i) = 1 to (usernumber)
    ask join Enter Number (i) : and wait
    set sum to ((sum) + (answer))
  say join Sum is: (sum) for 2 secs
  ask Add new set of numbers? Press any key to continue; x to exit and wait
  set exit to answer
```

If Statement checks to see if the user guessed 3 times and if the user guessed correctly on the 3rd Attempt

Save the program

Click on this Icon > Save > Select Computer > Change Name as needed > the .xml code will download to the computer into the Downloads Folder > right mouse button on the file name or navigate to the Downloads Folder > Copy and Paste Code to a secure location (i.e. student directory, flash drive, etc.)

See Reference Video on Class Website or Google Classroom for video demonstration

Submission: Save Project > Change File name to Snap Tutorial 3 Loops "Student Last Name > Email .xml file to jourdem@brightonk12.com for grading