Data and Information Management

Data is raw material and facts that are unprocessed. Computer data is insignificant until it is processed and organised in a meaningful way.

Information is when data is processed, organised and structured to be used in a meaningful way.

Computer knowledge is the ability of a person to convert data into information.

Computer data storage refers to computer components that are used to retain or hold onto digital data or information. Data storage can be divided into primary storage and secondary storage.

Primary storage is the main area of the computer where data can be accessed quickly by the processor. Two examples of primary memory are random access memory (RAM) and read only memory (ROM). ROM is non – volatile permanent memory, where information is not lost when the computer is switched off. The computer manufacturer provides the programs in ROM and these programs cannot be changed by the user. The user does not have access to ROM, consequently programs cannot be modified, deleted or added on. RAM is volatile memory temporary memory where information is lost if the computer is switched off. RAM is also known as working memory where the user can delete and rewrite information when required.

Secondary storage is also known as auxiliary storage where data and information is stored on an external medium for future use. Examples of external or secondary storage devices are the hard drives, DVDs, CDs and memory sticks.

Bits and Bytes

The terms **bit** and **byte** are commonly used in the computer industry. Both terms refer to digital data transmitted over a connection. A bit is a single numeric value, either '1' or '0', that encodes a single unit of digital information. A byte is a sequence of bits; usually eight bits equals one byte.

Hierarchy of Data Storage:

Smallest entity of data	Bit
8 Bits	1 Byte
1024 Bytes	1 Kilobyte
1024 Kilobytes	1 Megabyte
1024 Megabytes	1 Gigabyte
1024 Gigabytes	1 Terabyte
1024 Terabytes	1 Petabyte
1024 Petabytes	1 Exabyte

Data Representation

This means the medium used to represent numbers in a computer. Every computer works essentially with a combination of two numbers that is 0 and 1, the numbers 0 and 1 are known to be binary digits. This number system is called the Binary number system where the term bi means 2.

Computer programmers need to translate the binary number system to other number systems to make translation of numbers and data easier for the user to understand.

The other number systems are the following, Decimal, Octal and Hexadecimal.

Altogether we will be studying 4 number systems that have the following bases.

Binary, **base 2**

Octal, **base 8**

Decimal, **base 10**

Hexadecimal, **base 16**

Number System	Base	Values
Binary	2	0 and 1
Octal	8	0, 1, 2, 3, 4, 5, 6, 7
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Representation of Numbers

We will start with decimal numbers since we are familiar working with these numbers in Maths and in everyday life.

Example 1: Representation of a decimal number 3725.

$= (3 \times 10^{3)} + (7 \times 10^{2}) + (2 \times 10^{1}) + (5 \times 10^{0})$

$= 3000 + 700 + 20 + 5$

$= 3725$

Example 2: Representation of a decimal number 24986.

$= (2 \times 10^{4)} + (4x\ 10^{3)} + (9 \times 10^{2}) + (8 \times 10^{1}) + (6 \times 10^{0})$

$= 20000 + 4000 + 900 + 80 + 6$

$= 24986$

Examples of numbers in other number systems.

Binary numbers: Notice the digits used can only be 0's and 1's.

$101_{2,}$ $11110_{2,}$ $10101100_{2,}$ 10_2

Octal numbers: Notice the digits used range from 0 to 7.

6431_8, $31226_{8,}$ $736755_{8,}$ 651_8

Hexadecimal Numbers: Notice the digits used range from 0 – 9 and letters from A to F.

$85BD_{16}$, $1CCF45_{16}$, $23ED_{16}$, $98ABC_{16}$

Conversions:

2^0	*1*
2^1	*2*
2^2	*4*
2^3	*8*
2^4	*16*
2^5	*32*
2^6	*64*
2^7	*128*
2^8	*256*

Binary to Decimal

Example 1: Convert 110_2 to a decimal number.

Step 1: Create a power for each digit, that is

2	1	0
1	1	0

Step 2: Multiply each digit by 2, raised to the given power, that is

$= (1 \times 2^{2)} + (1 \times 2^1) + (0 \times 2^0)$

$= (1 \times 4) + (1 \times 2) + (0 \times 1)$

$= 4 + 2 + 0$

$= 6$

Therefore $110_2 = 6$.

Example 2: Convert 101011_2 to a decimal number.

Step 1: Create a power for each digit, that is

5	4	3	2	1	0
1	0	1	0	1	1

Step 2: Multiply each digit by 2, raised to the given power, that is

$= (1 \times 2^{5)} + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

$= 32 + 0 + 8 + 0 + 2 + 1$

$= 43$

Therefore $101011_2 = 43$.

Activity 1.1: Convert from binary to decimal.

1. 111_2 2. 110011_2 3. 100101_2
4. 1010101_2 5. 11111_2 6. 10111011_2

Octal to Decimal

Example 1: Convert 413_8 to a decimal number.

Step 1: Create a power for each digit, that is

2	1	0
4	1	3

Step 2: Multiply each digit by 8, raised to the given power, that is

$= (4 \times 8^{2)} + (1 \times 8^1) + (3 \times 8^0)$

$= (4 \times 64) + (1 \times 8) + (3 \times 1)$

$= 256 + 8 + 3$

$= 267$

Therefore $413_8 = 267$.

8^0	*1*
8^1	*8*
8^2	*64*
8^3	*512*
8^4	*4096*
8^5	*32768*
8^6	*262144*

Example 2: Convert 2571_8 to a decimal number.

Step 1: Create a power for each digit, that is

3	2	1	0
2	5	7	1

Step 2: Multiply each digit by 8, raised to the given power, that is

$= (2 \times 8^3) + (5 \times 8^2) + (7 \times 8^1) + (1 \times 8^0)$

$= (2 \times 512) + (5 \times 64) + (7 \times 8) + (1 \times 1)$

$= 1024 + 320 + 56 + 1$

$= 1401$

Therefore $2571_8 = 1401$.

Activity 1.2: Convert from octal to decimal.

1. 252_8 2. 1364_8 3. 6332_8
4. 12771_8 5. 55612_8 6. 1777_8

Hexadecimal to Decimal

Example 1: Convert 248_{16} to a decimal number

Step 1: Create a power for each digit, that is

2	1	0
2	4	8

16^0	*1*
16^1	*16*
16^2	*256*
16^3	*4096*
16^4	*65536*

Step 2: Multiply each digit by 16, raised to the given power, that is

$= (2 \times 16^2) + (4 \times 16^1) + (8 \times 16^0)$

$= (2 \times 256) + (4 \times 16) + (8 \times 1)$

$= 512 + 64 + 8$

$= 584$

Therefore $248_{16} = 584$

Note:
When converting from any number system to Decimal, we write the exponents over the number and we use the expansion format

Example 2: Convert $8D17_{16}$ to a decimal number

Step 1: Create a power for each digit, that is

3	2	1	0
8	D	1	7

Step 2: Multiply each digit by 16, raised to the given power, that is

$= (8 \times 16^3) + (13 \times 16^2) + (1 \times 16^1) + (7 \times 16^0)$

$= (8 \times 4096) + (13 \times 256) + (1 \times 16) + (7 \times 1)$

$= 32768 + 3328 + 16 + 7$

$= 36119$

Therefore $8D17_{16} = 36119$

Activity 1.3: Convert from Hexadecimal to decimal.

1. 896_{16}
2. $A87_{16}$
3. $3F2_{16}$
4. 96512_{16}
5. $15B2_{16}$
6. $2CE5_{16}$

Decimal to Binary

Example 1: Convert 238 to a binary number.

Divide the number 238 repeatedly by base 2 until the value 0 is reached.

Base	238	Remainder
2	119	0
2	59	1
2	29	1
2	14	1
2	7	0
2	3	1
2	1	1
2	0	1

Read the remainder column from the bottom to the top.

$238 = 11101110_2$

Example 2: Convert 197 to a binary number.

Divide the number 197 repeatedly by base 2 until the value 0 is reached

Base	197	Remainder
2	98	1
2	49	0
2	24	1
2	12	0
2	6	0
2	3	0
2	1	1
2	0	1

Read the remainder column from the bottom to the top.

$197 = 11000101_2$

Activity 1.4: Convert from decimal to binary.

1. 126
2. 89
3. 1127
4. 679
5. 453
6. 756

Decimal to Octal

Example 1: Convert 578 to an octal number.

Divide the number 578 repeatedly by base 8 until the value 0 is reached.

Base	578	Remainder
8	72	2
8	9	0
8	1	1
8	0	1

Read the remainder column from the bottom to the top.

$578 = 1102_8$

Example 2: Convert 6337 to an octal number.

Divide the number 197 repeatedly by base 2 until the value 0 is reached

Base	6337	Remainder
8	792	1
8	99	0
8	12	3
8	1	4
8	0	1

Read the remainder column from the bottom to the top.

$6447 = 14301_8$

Activity 1.5: Convert from decimal to octal.

1. 6267
2. 5116
3. 11274
4. 1767
5. 2334
6. 3456

The Hexadecimal Number System

Decimal Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Decimal to Hexadecimal

Example 1: Convert 532 to a Hexadecimal number.

Divide the number 532 repeatedly by base 16 until the value 0 is reached.

Base	532	Remainder
16	33	4
16	2	1
16	0	2

Read the answer from the remainder column bottom to the top.

$532 = 214_{16}$

Example 2: Convert 983 to a Hexadecimal number.

Divide the number 983 repeatedly by base 16 until the value 0 is reached.

Base	983	Remainder
16	61	7
16	3	D
16	0	3

Read the answer from the remainder column bottom to the top.

$983 = 3D7_{16}$

Activity 1.6: Convert from decimal to hexadecimal.

1. 543
2. 49
3. 4597
4. 7692
5. 4553
6. 75644

Activity 1.7: Determine the value of X in each case.
The decimal number system can be represented without the base value.

1. $542_8 = X$
2. $X_2 = 7928$
3. $9317 = X_{16}$
4. $5AC3_{16} = X$
5. $10111011_2 = X$
6. $X = ECF3_{16}$

Digital Character Representation

This means the specifications for converting data into computer-usable form.

Examples are: Unicode, ASCII / UTF – 8

Unicode provides a unique number for every character, irrespective of the program or language that it is written in. This helps to make programs, operating systems and websites multilingual.

ASCII – American Standard Code for Information Interchange.

ASCII is a character encoding system based on the sequence of the English Alphabet.

UTF-8 is a multi – byte character encoding in unicode. This set is used to represent each character in the unicode chart using 8 bits.

ASCII chart (See Appendix at the end of the book)

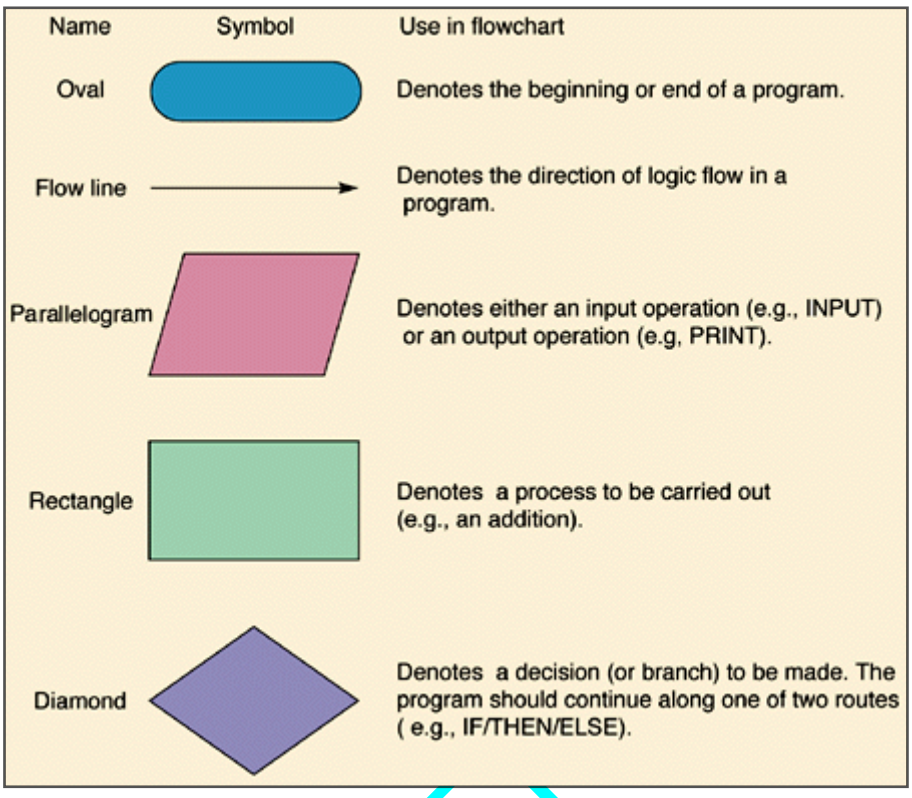
Data Storage Structures:

Data can be stored in various structures such as text files and databases. These structures store data permanently and can be manipulated at a later stage.

Algorithms

Before writing programs or solutions into an environment, one has to first write the steps involved in solving that problem. The step-by-step process to solving a problem is called an ***algorithm***. The reason for doing algorithms is to make a task or program easier to write and to understand. An algorithm can take one of two forms namely ***pseudocode*** or ***flowcharts***. Pseudocode is the process of writing the steps of a program using structured statements, without the use of any particular programming language while a flowchart is a diagrammatical representation of a program. Instructions and statements are placed in boxes that are linked by arrows depending on the flow direction of the flowchart. An algorithm can be checked using a trace table. A trace table is a logical representation of the steps of the algorithm using a table. The results of the table will indicate if the output of the algorithm is correct or not.

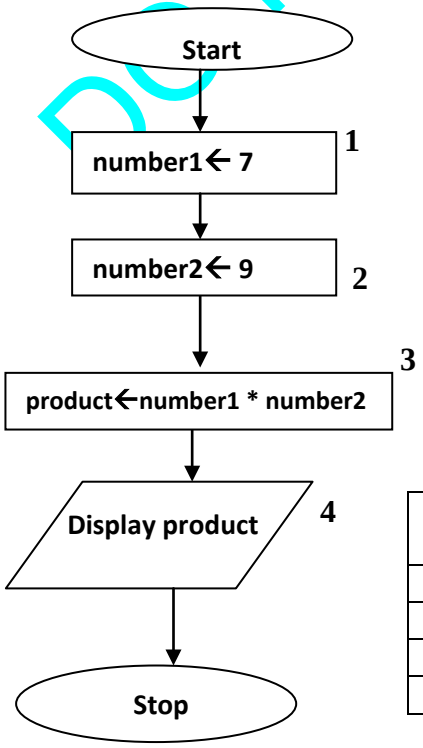
Flowchart representation



Example of a flowchart algorithm

Question 1: Draw a flowchart to assign a variable called number1 to 7 and a variable called number2 to 9. The flowchart must determine and display the product of the numbers

Answer:



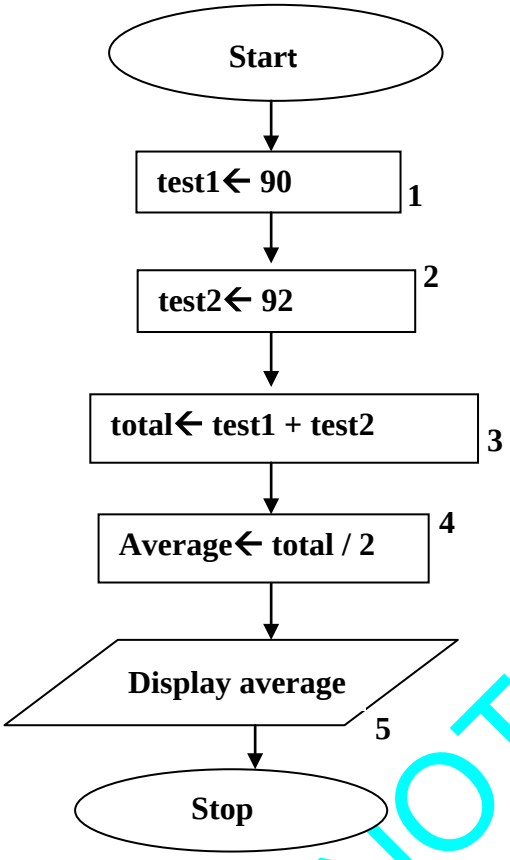
Trace Table

Box	number1	number2	product	Screen output
1	7			
2		9		
3			63	
4				63

Question 2:

You have written two tests in English. You achieved 90 and 92 for the tests. Draw a flowchart to find and display the average mark. Also draw a trace table to display the results.

Answer:



Trace Table

Box	test1	test2	total	average	Screen output
1	90				
2		92			
3			182		
4				91	
5					91

Activity 2.1

Draw flowcharts for the following questions.

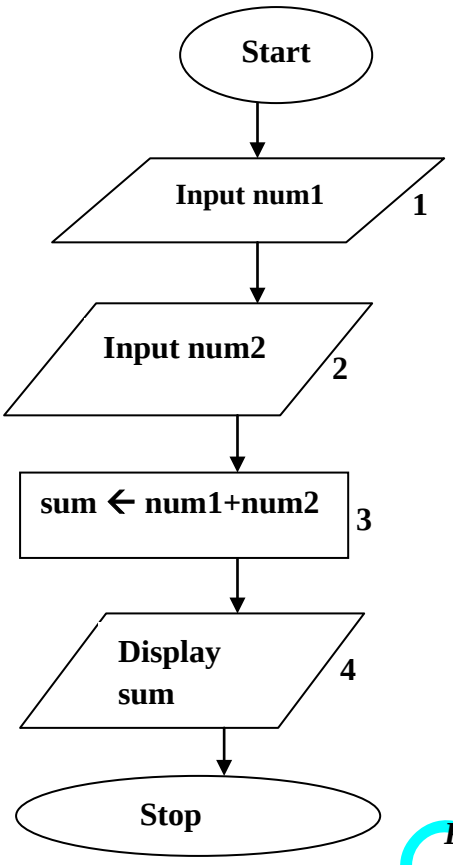
1. The length of a rectangle is 12m and the breadth is 8m. Determine and display the area of the rectangle.
2. A salesman works for 9 hours a day at a rate of R50.00 per hour. Calculate and display his earnings per day and for the week if he works only on weekdays.
3. A student gets an allowance of R500.00 per month. He buys the following:
   A book for R120.00, a hot stone massage for R100.00 and an adventure ride for R85.00.
   Calculate and display the amount spent and the amount left over.

Instead of assigning values as in the flow diagrams above, we can allow the user to input values as these values will differ each time.

Question 3:

Draw a flowchart to input two numbers. Process and display the sum of the numbers.

Answer:



Notice a parallelogram shape is used for input and for output.

Trace Table: ***Use values: num1 = 8, num2 = 3.***

Box	num1	num2	sum	Screen output
1	8			
2		3		
3			11	
4				11

Redraw the trace table to use num1 = 12, num2=9.

Activity 2.2: Draw flowcharts for each of the following:

1. Assign 3 numbers num1 = 7, num2 = 4 and num3 = 5. Find and display the product. Draw a trace table to illustrate the results.

2. The price of a chocolate is 9.49 and the price of a cool drink is 8.99. Draw a flowchart to find and display the total price. Draw a trace table to illustrate the results.

3. The computer you wish to buy is an amount of R5000. You have saved R2550. Determine and display the outstanding amount required to buy the computer.

4. Input the length and breadth of a rectangle. Calculate and display the perimetre of the rectangle. Draw a trace table for length = 25 and breadth = 14.

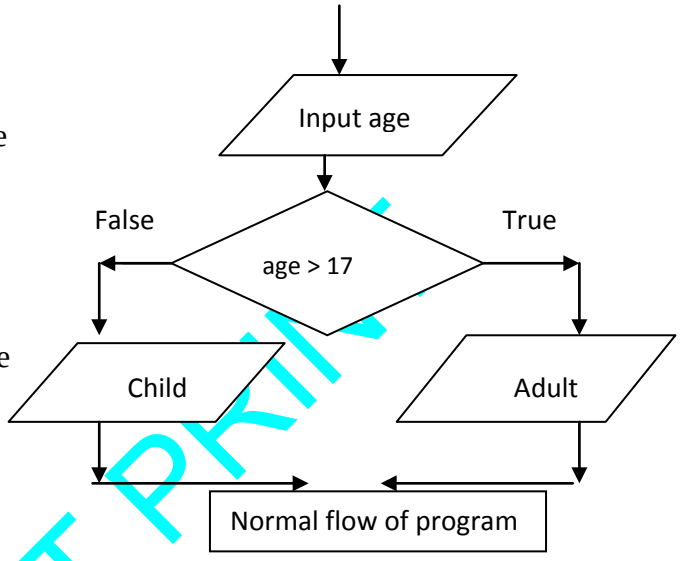
5. Input a temperature in °Celcius. Convert and display the same temperature in Fahrenheit. Formula: Fahrenheit = Celcius x 9 / 5 + 32
   Draw a trace table to indicate the result of 41°C.
6. Calculate and display the volume of a rectangle that has a length of 12.5m, breadth of 8.2m and a height of 7.9m. Draw a trace table to illustrate the results.

7. Input the prices of three items purchased. Calculate and display the total price. Calculate and display 14% VAT on the total price. If the customer tendered an amount of R300, calculate and display the change. Draw a trace table for the following values of the items, price1 = R54.99, price2 = R103.98 and price3 = R17.89.

Decision Boxes in Flow Diagrams

```
Get condition
if  (condition = true)
{
  statements  executed if condition is true
 }
else
 {
  statements executed if condition is false
  }
```

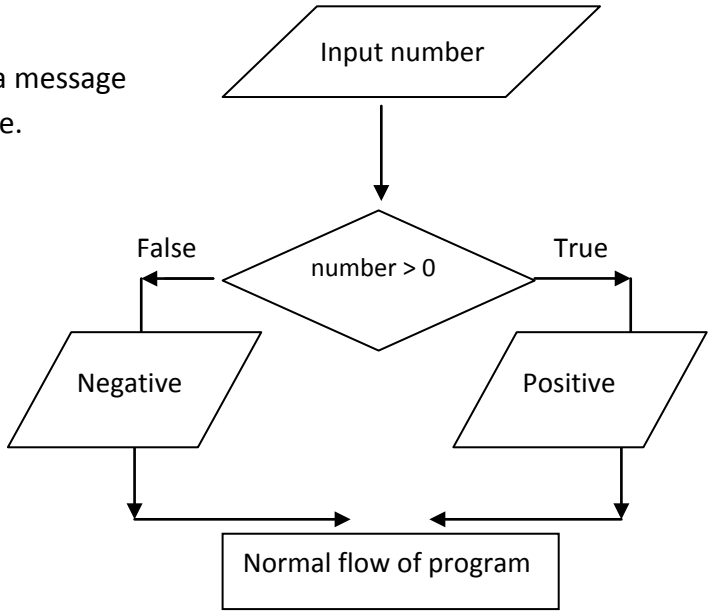
Flowchart representation



The decision box is a diamond shaped box controlled by the condition placed inside it.
The result of the decision box can either be true or false.
The result of the decision box will determine the flow control of the flowchart; that is if the result of the decision is true, you must follow the True arrow otherwise follow the False arrow.

Example 1: Enter a number. Display a message indicating Positive or Negative.

```
Input number
if  (number > 0)
{
  Display Positive
}
else
{
Display Negative
 }
```

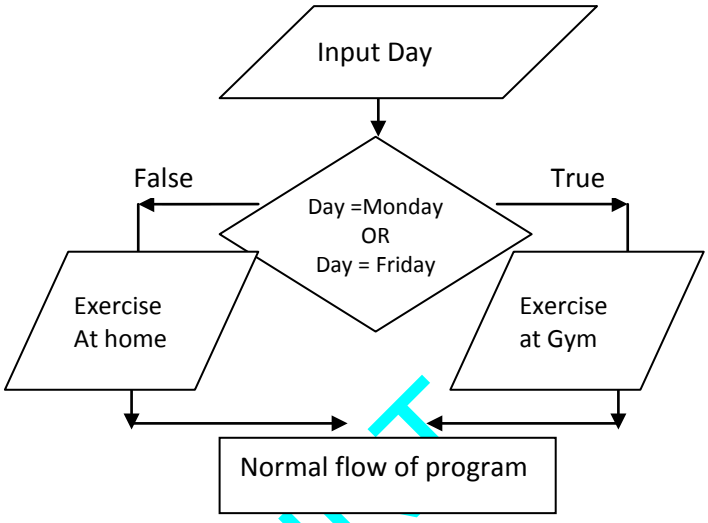


Example 2

Enter the day of the week. If day = Monday or Friday, display a message “Exercise at Gym”.

For all the other days, display a message “Exercise at home”.

```
Input Day
if  (Day = Monday OR Day = Friday)
{
 Display “Exercise at Gym”
 }
 else
{
   Display “Exercise at home”
}
```



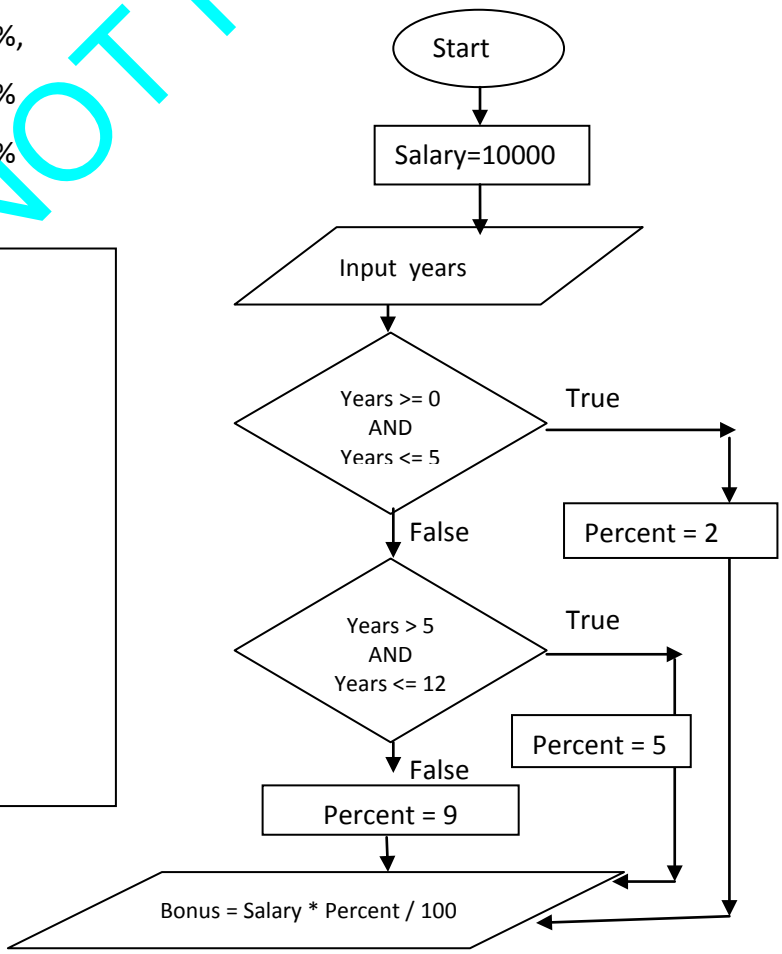
Example 3

An employee earns a basic salary of R10000.00. Calculate the bonus as follows.

The employee qualifies for a bonus based on the number of year’s service at the company.

0 years – 5 years	: 2%,
>5 years – 12years	: 5%
>12 years	: 9%

```
Salary = 10000
Input years
If years >= 0 And Years <=5
    Percent = 2
Else
If years > 5 And Years <=12
      Percent = 5
Else
     Percent = 9
Bonus = Salary * Percent / 100
```



Activity 2.3

Draw flowcharts for each of the following.

1. Input two numbers. Determine and display the smaller number.
2. Input the price of an item purchased at a store. A discount of 5% is awarded if the price is >=1000. Determine and display the discount and the final price.
3. Enter the value of sales made by an agent. The following percentages are awarded in commission based on the amount of sales made.

   0 – 5000 : 0%

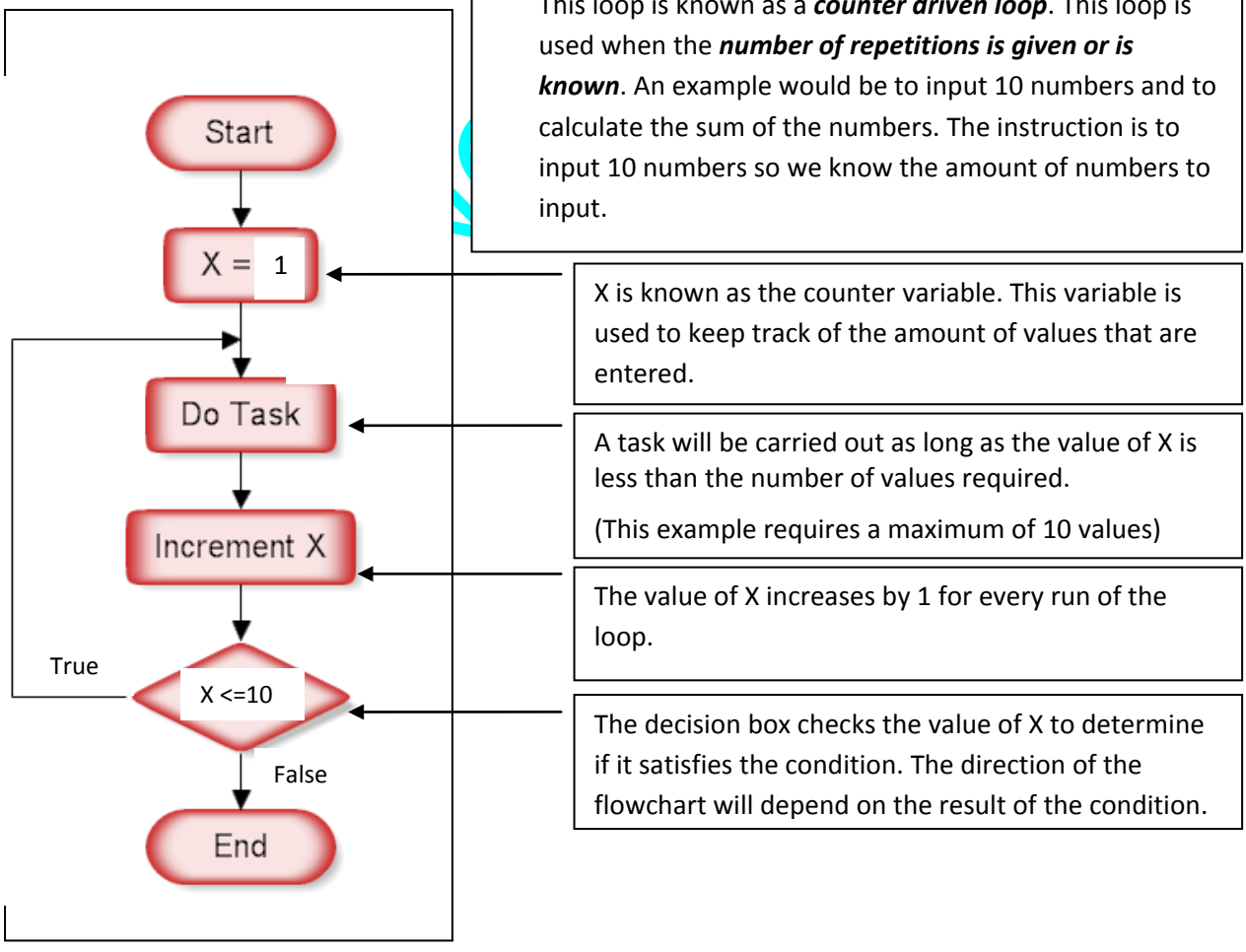
   >5000 – 10000 : 6%

   >10000 : 8%

   Display the value of commission based on the sales.

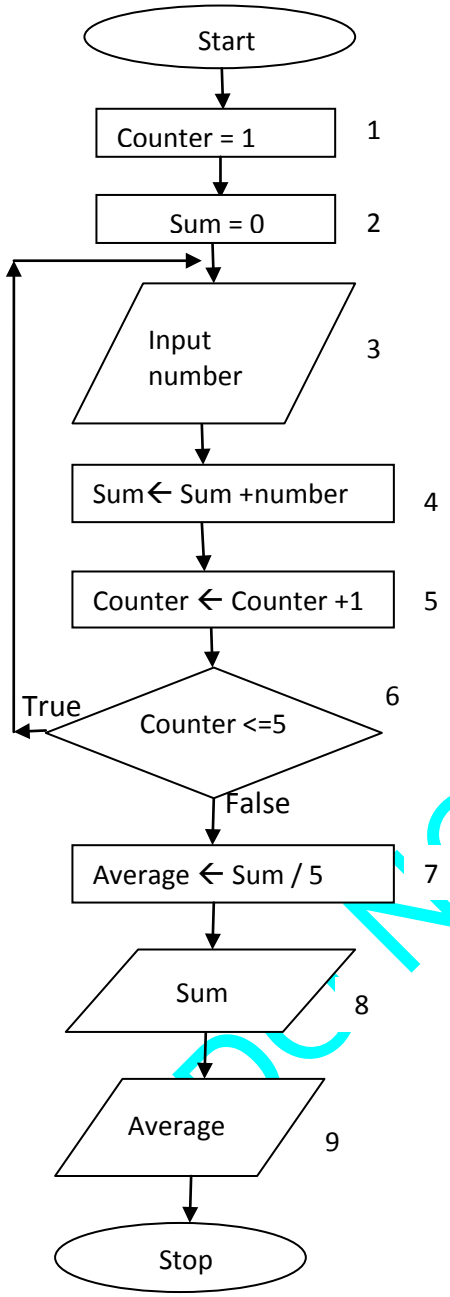
Repetition in Flowcharts: *When a task needs to be carried out more than once*

Figure A: **Looping using a counter**



Example 1:

Draw a flowchart to input 5 numbers. Calculate and display the sum and average of the numbers. Draw a trace table with the following data: 7; 9; 4; 12; 3



Trace Table

Step	Counter	Sum	number	Counter < =5	Average	Out put
1	1					
2		0				
3			7			
4		7				
5	2					
6				True		
3			9			
4		16				
5	3					
6				True		
3			4			
4		20				
5	4					
6				True		
3			12			
4		32				
5	5					
6				True		
3			3			
4		35				
5	6					
6				False		
7					7	
8						35
9						7

Activity 2.4

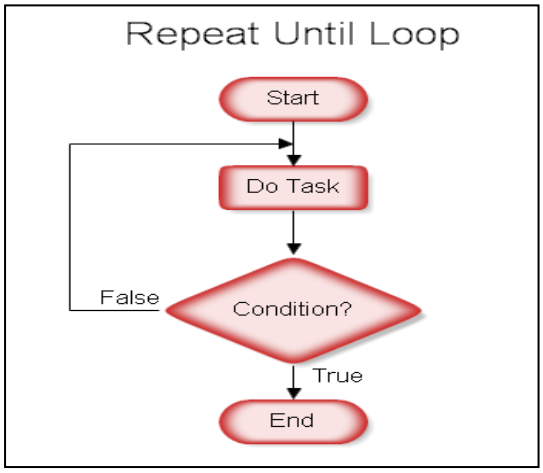
Draw flowcharts and trace tables for the following:

1. Input 4 numbers, count and display the amount of positive and the amount of negative numbers. Sample data: 13; -8; 9; 21
2. Input 5 numbers, determine and output the largest number. Sample data: 7; 12; 20; 14; 9
3. Input a number. Determine and output the factorial of the number.
   Example: The factorial of 4 = 1 x 2 x 3 x 4 = 24.
4. Input a number. Display the square of every alternate number from 1 to the number entered. Example if 10 is entered, the output would be 1 9 25 49 81.

Looping using a condition

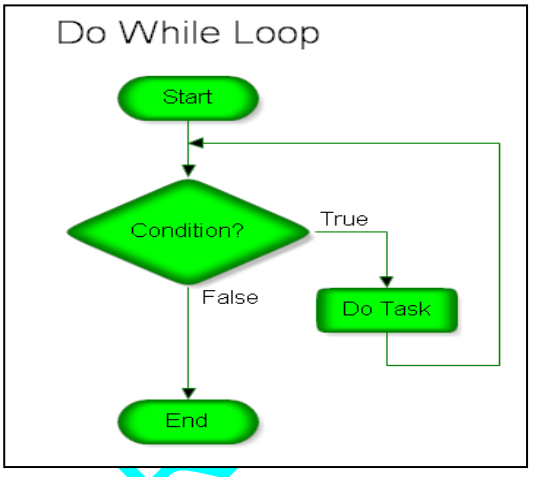
A ***conditional loop*** is used when the number of repetitions is not given or is ***unknown***. An example would be to enter values until the value 100 is entered. The number of repetitions is unknown as we could enter 5 values and then 100 or another might enter 8 values and then 100.

Figure B:



The Repeat…Until Loop carries out the task until the condition is true. The task is done before the condition is tested.

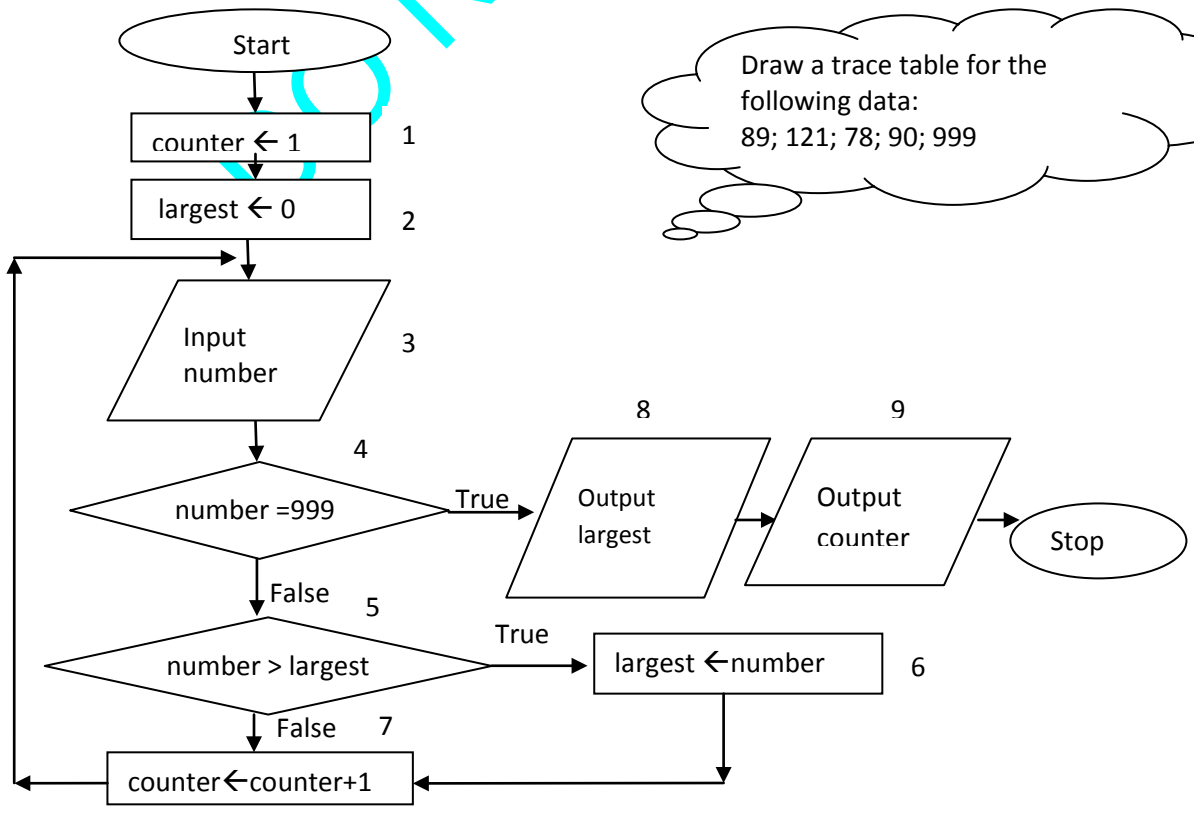
Figure C:



The While loop tests the condition first. The task will only be executed if the result of the condition is true.

Example 1:

Input numbers until the value 999 is entered. Display the largest number entered. Count and display the amount of numbers entered.



Activity 2.5

Draw flowcharts and trace tables for the following:

1. Input numbers until the number 555 is entered. Determine and output the sum of numbers
   Sample data: 10; 56; 69; 555

2. Input names of countries until the country South Africa is entered. Count and display the number of countries entered. Do not include South Africa in the count.
   Sample data: India; England; America; China; South Africa

3. Input marks of learners until the mark 100 is entered. Display the average mark.
   Sample data: 89; 61; 88; 56; 47; 100

4. Input a number until a number in the range 1 to 10 is entered. Display the cube of the number. Sample data: 12; 56; 13; 5

Pseudocode: Rules must be adhered to when writing pseudocode

Pseudocode is a combination of the spoken language eg English and a set of rules to allow the algorithm to be familiar to any programmer, irrespective of the programming language they are working with.

Let us learn the rules related to algorithms

Divide the solution into 4 parts: Sometimes, the solution might satisfy only 3 parts. Try as far as possible to keep your solution in this order.

Rule 1: Break up the algorithm into 4 parts (if you can).

- Part1: **Initialisation**, Are there any counter values or variables used for calculating totals?
- Part2: **Input**, Do you need to enter values?
- Part 3: **Processing**, Do you need to perform any calculations?
- Part 4: **Output**, Does the question require any display?

Rule 2: Use the words Input or Enter, Output or Display to input and output data.

Rule 3: Use the ← for assignment purposes, i.e. area = ½ *base * height will be area ← ½ * base * height.

Example of a pseudocode algorithm

Question 1: Write pseudocode to assign the value 16 to number1 and the value 19 to number2.

Calculate and display the sum of the numbers.

Answer

Step 1 number1 ← 16
Step 2 number2 ← 19
Step 3 sum ← number1 + number2
Step 4 Output sum

Trace Table

Step	number1	number2	sum	Screen Output
1	16			
2		19		
3			35	
4				35

Question 2: Write pseudocode to input the base and height of a triangle, Process and display the area. Draw a trace table for values base = 10, height = 12

Answer

Step 1 Input base

Step 2 Input height

Step 3 area ← ½ * base * height

Step 4 Output area

Trace Table

Step	base	height	area	Screen Output
1	10			
2		12		
3			60	
4				60

Activity 2.6: ***Provide a trace table for each.***

2.6.1

Step 1 Input num1
Step 2 Input num2
Step 3 product ← num1 * num2
Step 4 diff ← num1 – num2
Step 5 sum ← num1 + num2
Step 6 Output product
Step 7 Output diff
Step 8 Output sum

num1 = 8, num2 = 3

2.6.2

Step 1 Input length
Step 2 Input breadth
Step 3 area ←length * breadth
Step 4 perimetre ← 2 * (length + breadth)
Step 5 Output area
Step 6 Output perimeter

length = 10, breadth = 6

2.6.3

Step 1 Input name
Step 2 mark1← 87
Step 3 mark2 ← 93
Step 4 mark3 ← 67
Step 5 totalMark ← mark1+mark2 + mark3
Step 6 Output name
Step 7 Output totalMark

name = Jane

2.6.4

Step 1 Input client name
Step 2 Input price 1
Step 3 Input price 2
Step 4 Input price 3
Step 5 totalPrice ←price 1+price 2 +price 3
Step 6 vat ← totalPrice * 0.14
Step 7 finalPrice ← totalPrice + vat
Step 8 Output totalPrice
Step 9 Output vat
Step 10 Output finalPrice
Step 11 Input amt tendered
Step 12 change = amt tendered - finalPrice

Client name = Peter; price 1 = 5.69; price 2 = 4.99; price3 = 12.39
amt tendered = 50

Activity 2.7: Write pseudocode algorithms and draw tracetables for the following examples:

2.7.1 A slice of pizza at the tuck shop is R5.49. Enter the number of slices required and display the cost of the pizza.

2.7.2 Enter the name and sales made by an estate agent. Calculate 5% commission on the sales. Display the name, sales and commission.
Draw a trace table to illustrate the values.

2.7.3 Enter the employee number, the number of hours worked and the rate per hour. Calculate the gross salary, 14 % tax and the net salary. Display the net salary.
Draw a trace table to illustrate all values.

2.7.4 Assign the prices of pens, pencils and rulers to be R2.99, R 3.49 and R1.99 respectively. Calculate the income from each item if the quantity sold is:
pens: 50; pencils: 23; rulers: 40
Display the income for each item as well as the total income.
Draw a trace table to illustrate the values.

Selection Algorithms

Question Three:

Write pseudocode to input a mark. Display a message "Successful" if the mark >= 40 otherwise display a message "Unsuccessful". Draw a trace table for a mark of 75.

```
Step 1  Input mark
Step2   If mark >= 40
Step3   Output   "Successful"
        Else
Step4   Output "Unsuccessful"
```

Step	mark	If mark >= 40	Screen Output
1	75		
2		True	
3			Successful
4			

Note: Step 2 is known as the condition step. The result of the condition is either true or false; it cannot be both. If the condition is true, step 3 will be executed otherwise step 4 is executed.

Question Four:

Write pseudocode to input the sales value made by a property agent. Calculate and display the commission as follows: Draw a trace table for R60000.00 in sales.

R1.00 to R50000.00 : Commission = 5%
>=R50000.00 to R100000 : Commission = 8%
>=R100000.00 : Commission = 10%

```
Step 1  Input sales
Step 2  If sales >= 1 and sales <=50000
Step 3  Output   sales * 5 / 100
        Else
Step 4  If sales >50000 and sales<= 100000
Step 5  Output   sales * 8 / 100
        Else
Step 6  Output   sales * 10 / 100
```

Step	sales	If sales >= 1 and sales <=50000	If sales >50000 and sales <=100000	Screen Output
1	60000			
2		False		
3				
4			True	
5				4800
6				

Note : Step 2 and step 4 have conditions.
If the condition in step 2 is satisfied, then the result in step 3 will be executed.
If the condition in step 4 is satisfied, then the result in step 5 will be executed.
If the conditions in step 2 and step 4 are not satisfied, then step 6 will be executed.

Activity 2.8
Write pseudocode algorithms and draw trace tables for the following examples:

1. Input a number. Display a message indicating if the number is positive or negative.
   Sample data: 56
2. Input 2 numbers. Determine and display the larger number.
   Sample data: number1 = 8; number2 = 5
3. Input 3 numbers. Determine and display the largest number.
   Sample data: number1 = 25; number2 = 14; number3 = 91
4. Input a temperature reading. Display messages using the following range of values.
   -20 to 0 :Freezing
   >0 to 10 :Cold
   >10 to 30 :Normal
   >30 :Hot
   Sample data: 25

Looping Algorithms

Question One:
Write pseudocode to input 4 numbers. Display the sum and average of the numbers.
Sample data: 9; 12; 7; 8

```
Step 1  sum ← 0
Step2   Loop from 1 to 4
        Start loop
Step3           Input number
Step4   sum ← sum + number
        End loop
Step5   average← sum / 4
Step6   Display average
```

Note: Steps 2; 3; 4 are executed 4 times

Step	sum	Loop counter	number	average	Screen Output
1	0				
2		1			
3			9		
4	9				
2		2			
3			12		
4	21				
2		3			
3			7		
4	28				
2		4			
3			8		
4	36				
5				36/4 =9	
6					9

Activity 2.9
Write pseudocode algorithms and draw trace tables for the following examples:
1. Input 7 marks each out of 100. Display the total mark and the average mark.
   Sample data: 89; 56; 67; 83; 91; 90; 78
2. Input 5 numbers. Display the number and the square of the number.
   Sample data: 6; 4; 1; 9; 7
3. Input 5 numbers. Display the largest and the smallest number separately.
   Sample data: 89; 67; 93; 82; 77
4. Input 3 names and 4 marks for each name. Display the total mark for each name.
   Also display the name of the person with the highest total mark. Sample data:
   Carol 87; 56; 77; 42 Nealin 99; 92; 91; 89 Tamara 78; 67; 55; 62

SCRATCH Programming

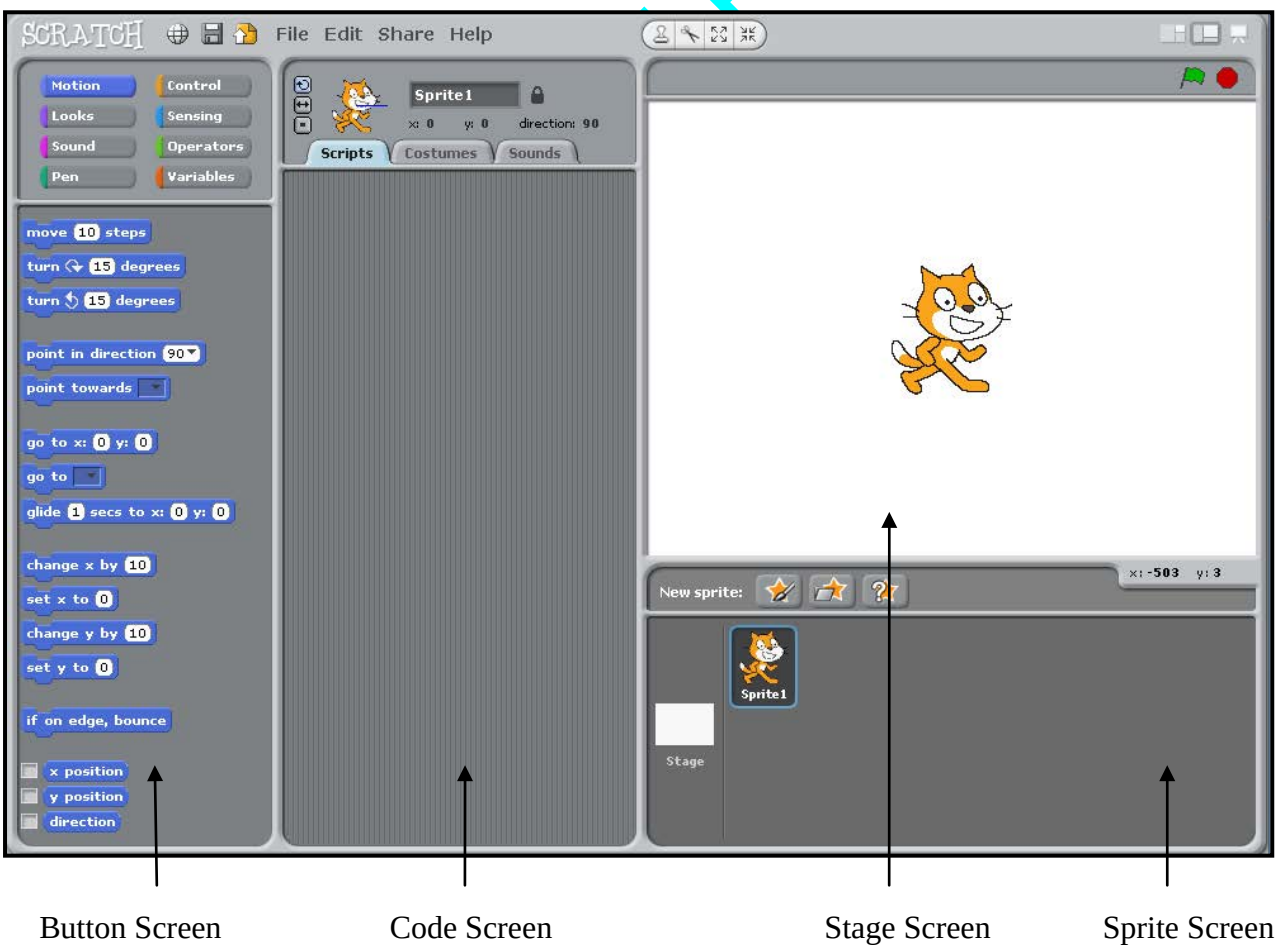
Scratch is a visual programming language designed to teach concepts and constructs using an interactive, creative, innovative and fun filled way. Scratch is developed by the Lifelong Kindergarten group at MIT by a team led by Mitchel Resnick and first appeared in 2006 as an educational language to teach programming using blocks of commands, sounds, animation and variables. Scratch is very popular as it provides a syntax free environment to create a project.

Reasons for using Scratch

- Provides a graphical user interface that teaches programming in a creative way.
- Allows for the design of tasks that require the use of sequence, characters and animation.
- Develops important concepts like conditional statements, looping, variables, arrays and lists without the use of syntax.

Scratch environment: This is generally controlled by a sprite. A sprite is a character or an object that is selected to carry out a sequence of instructions. The object which is a cat at the centre of the screen below is an example of a sprite.

The screen is divided into four parts namely the button screen, the code screen, the stage screen and the sprite screen.



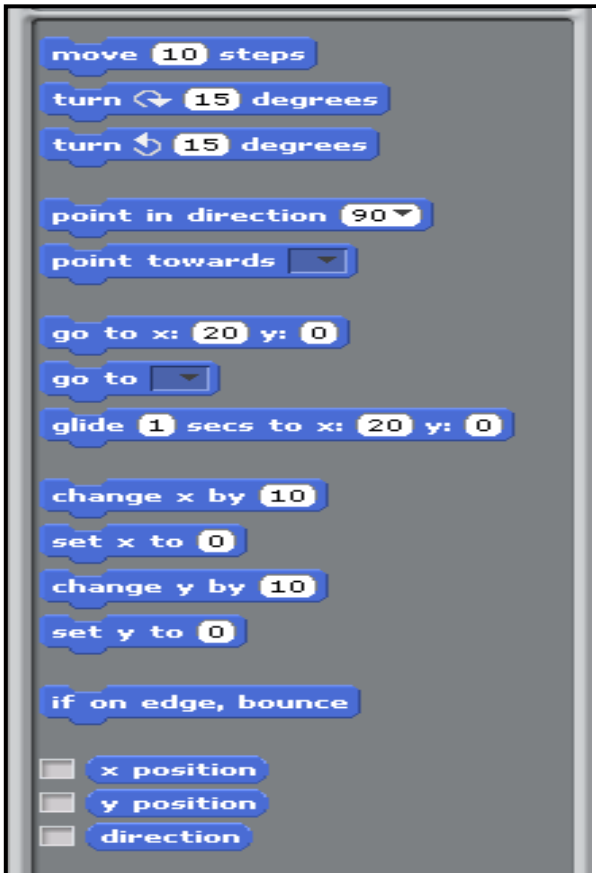
Button Screen Code Screen Stage Screen Sprite Screen

The Button Screen

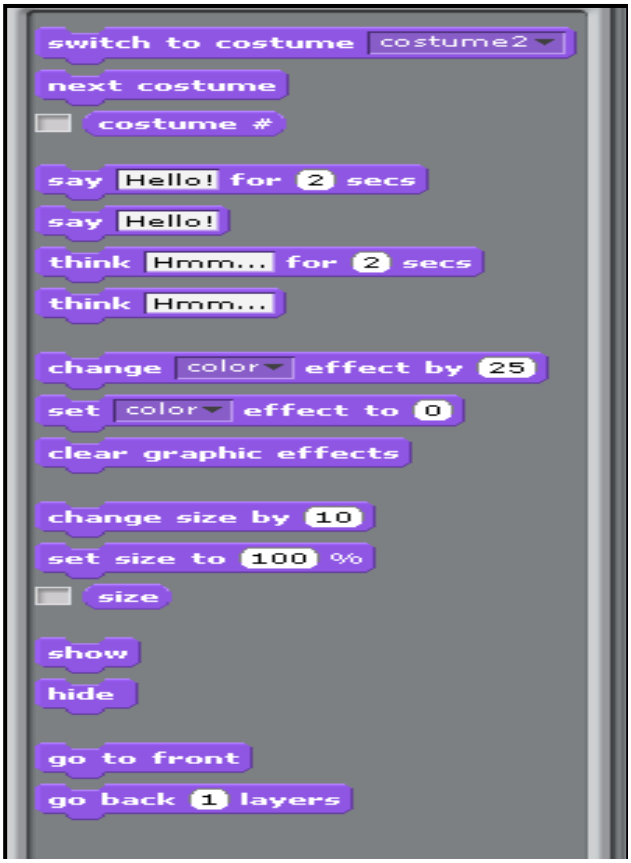
Here there are eight (8) buttons to select from and an area that displays the blocks for the selected button. The buttons are Motion, Control, Looks, Sensing, Sound, Operators, Pen and Variable.

Each button has its own set of blocks and a unique shade of colour.

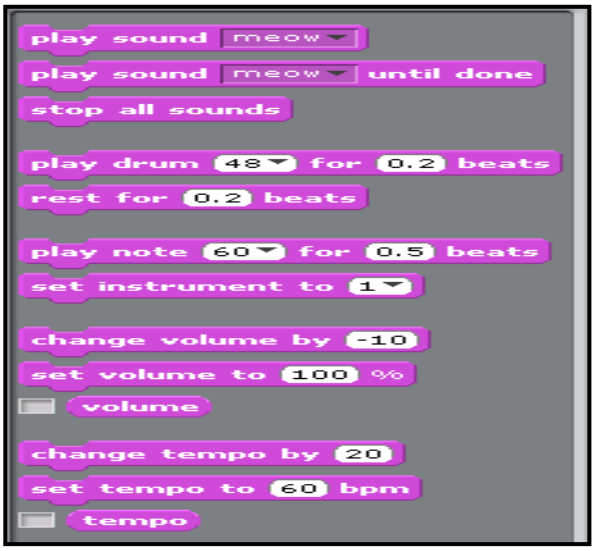
Motion



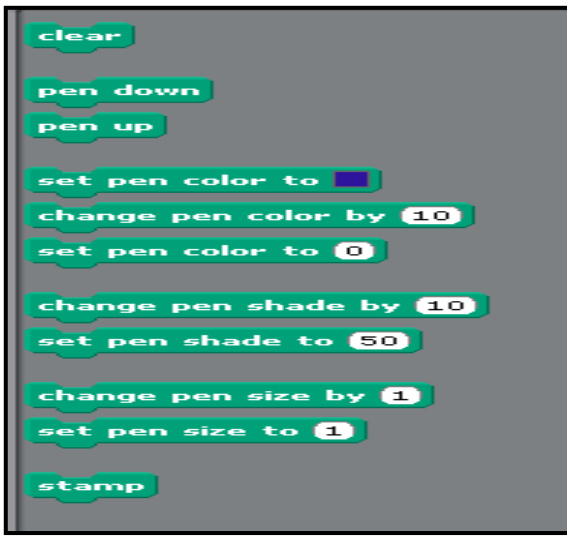
Looks

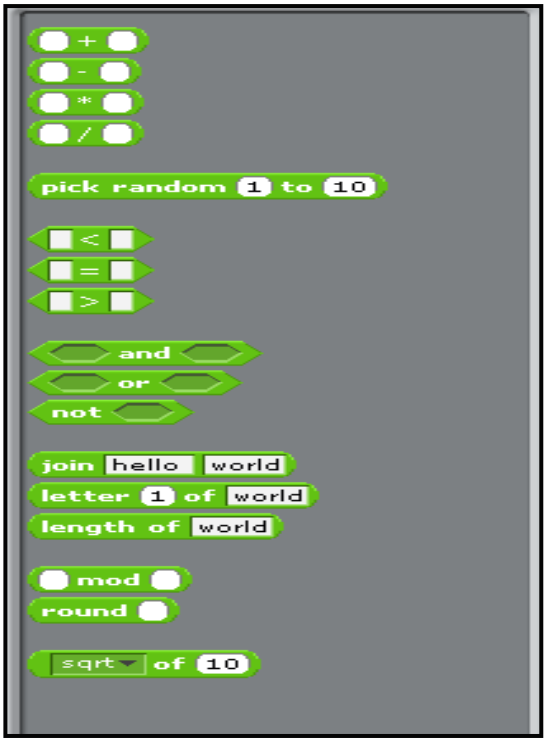


Sound



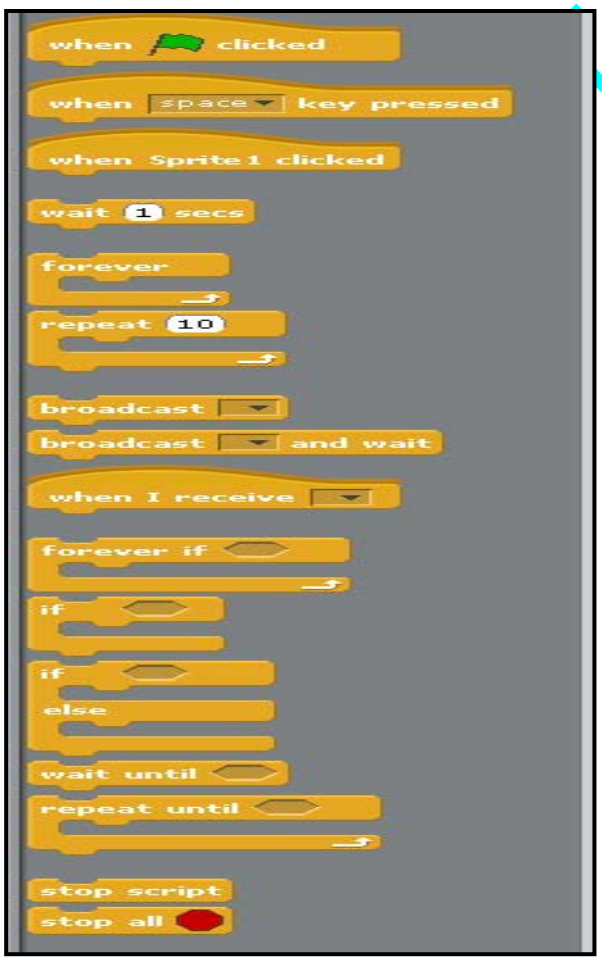
Pen



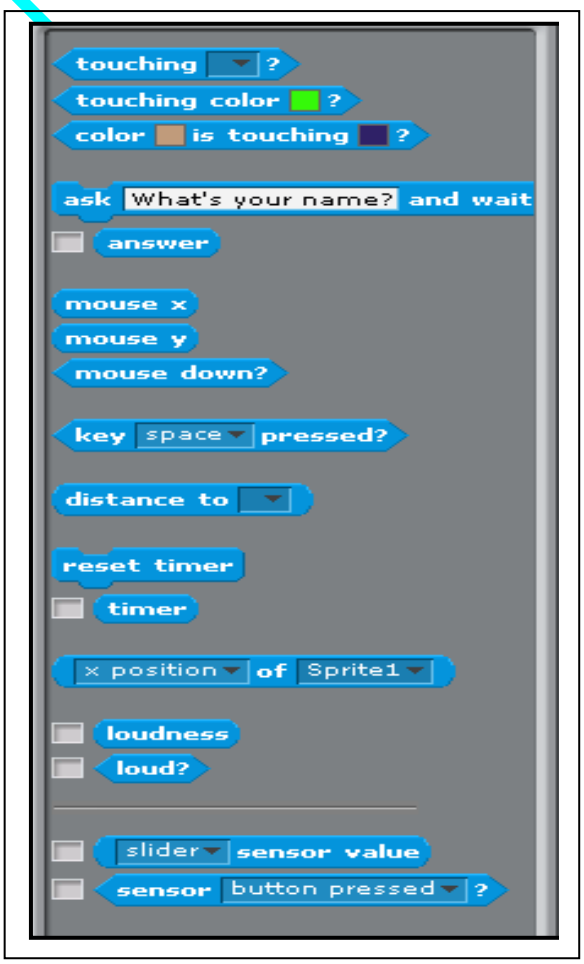


Operators

Control

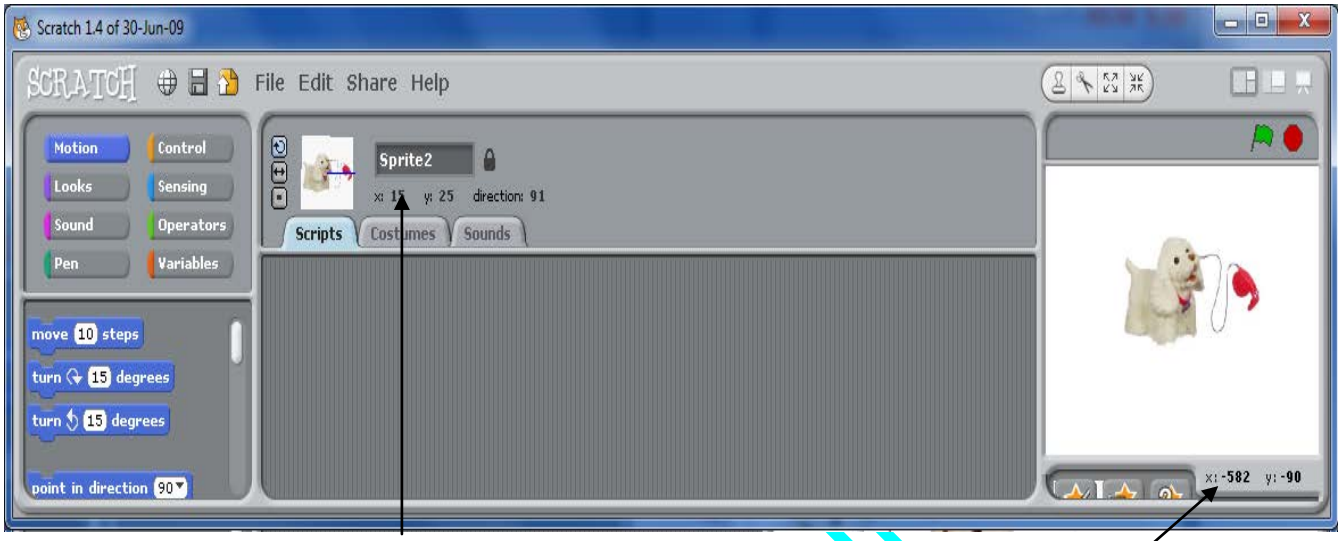


Sensing



The Code Screen

This area is used to place blocks together to perform a goal. The sequence of the blocks will control the selected sprite.



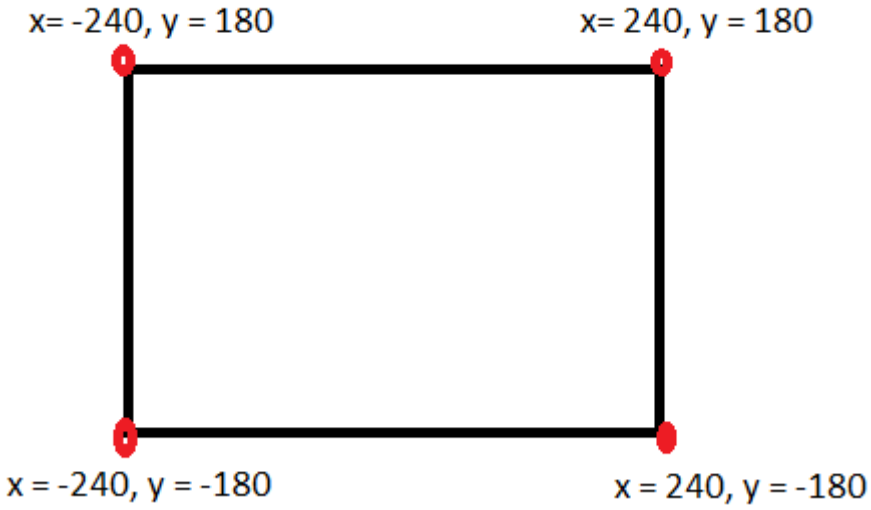
Represents the current co-ordinates and direction of the sprite

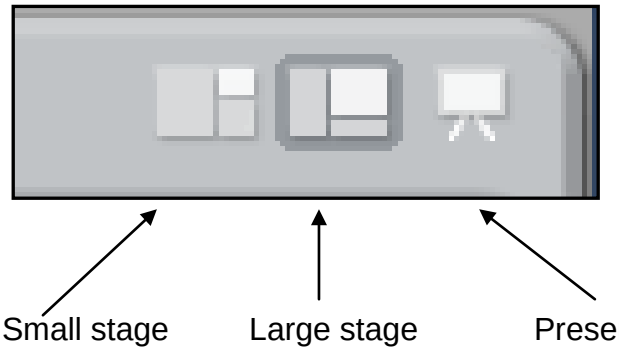
Represents the current co-ordinates of the mouse.

The Stage Screen

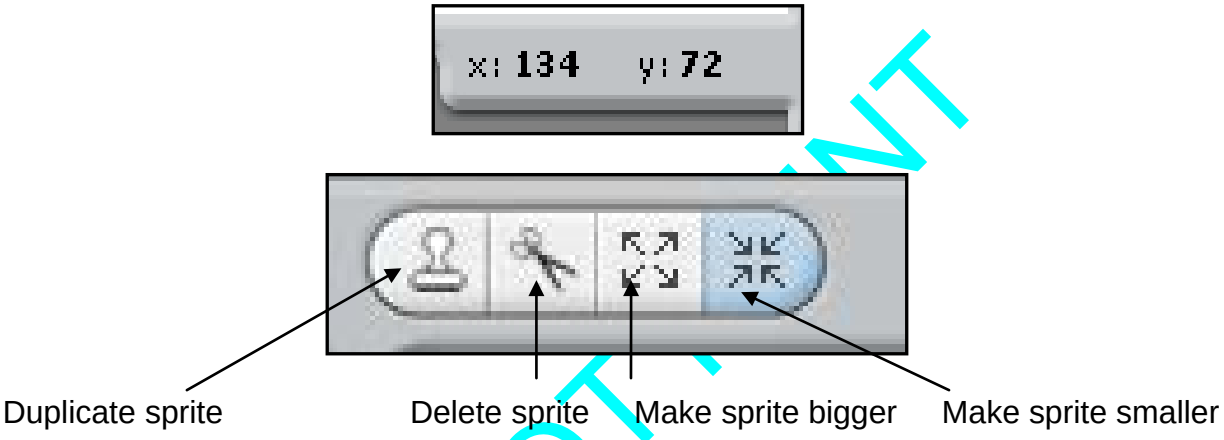
This area is where the blocks will come alive. The performance on the stage is controlled by the code screen. The stage is 480 units wide and 360 units high. It is divided into an x-y grid. The centre of the stage has an x-coordinate of 0 and a y-coordinate of 0.

To find out x-y positions on the stage, move the mouse (cursor) around and look at the **mouse x-y display** just below the Stage.

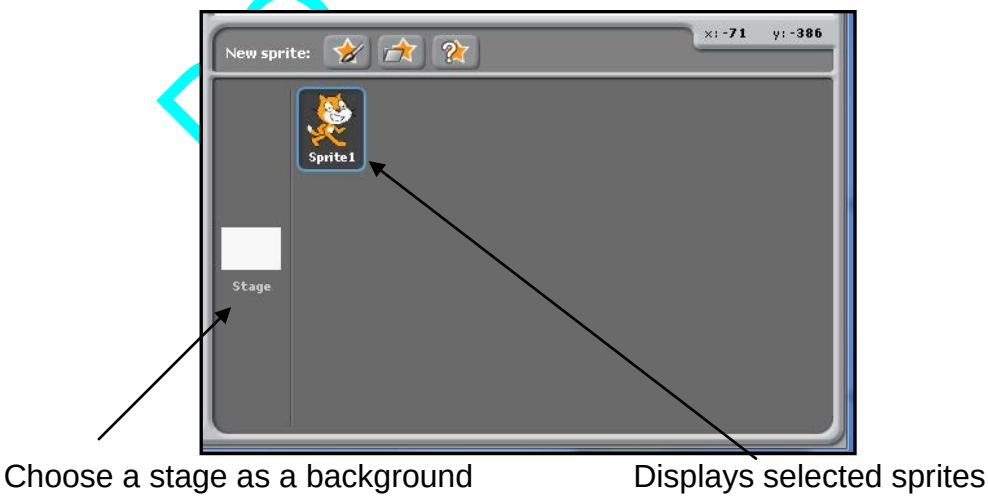




The x and y co-ordinates on the stage screen represent the position of the mouse at that point.



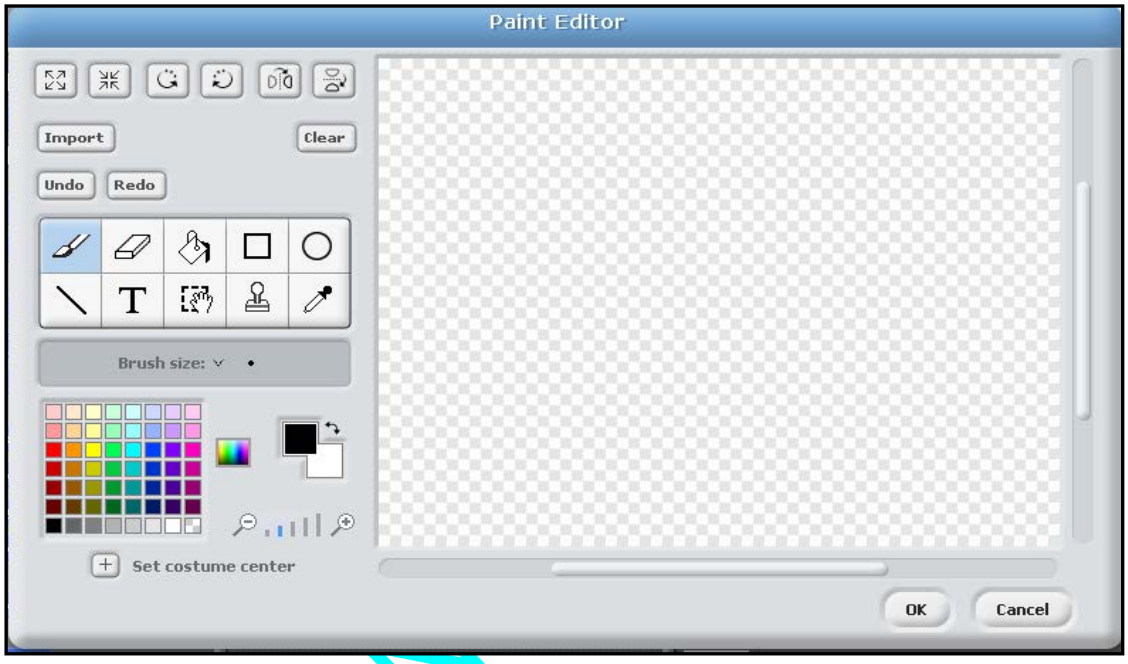
The Sprite Screen



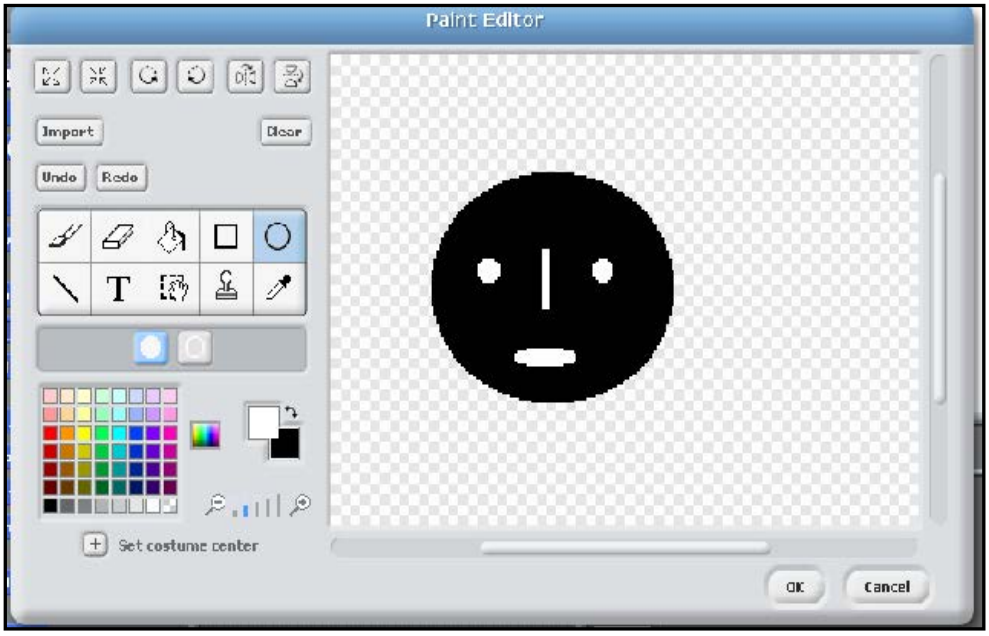
This area displays all the selected sprites for the program and creates / selects a stage. When a sprite is clicked on, the corresponding code to control the sprite will appear in the code screen.

How to select a sprite

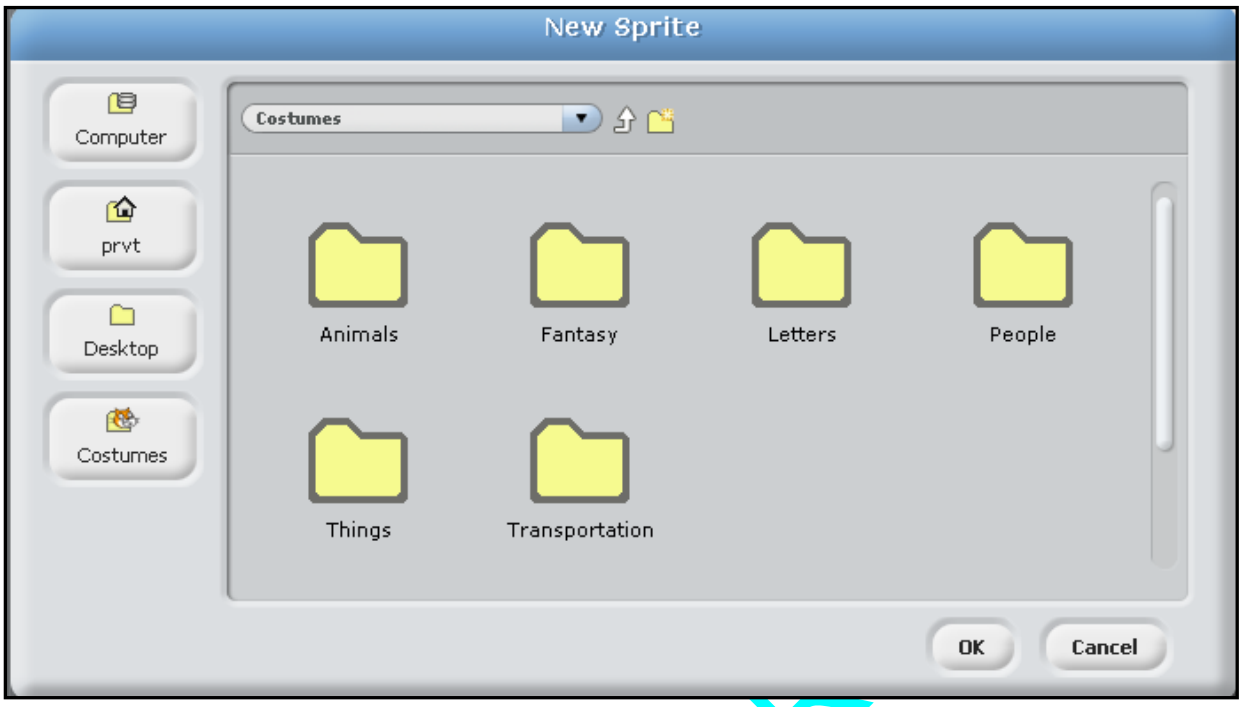
Paint a new sprite



Design a sprite as required and click on ok.
This is an example of a sprite that has been painted. Click on ok after the sprite is created.



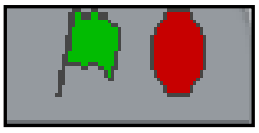
Choose a new sprite from file



You may choose a sprite from any file location, e.g. You may have saved a file on the desktop or in the Pictures folder etc.

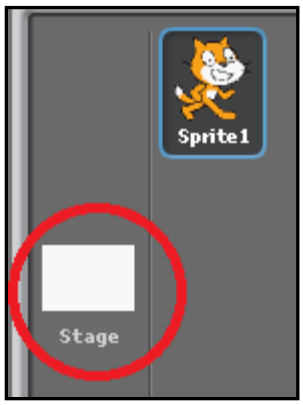
Get a surprise sprite

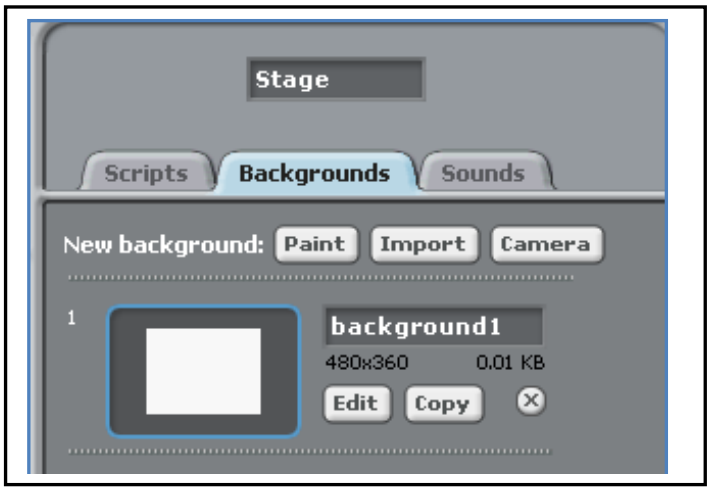
This option will choose a sprite randomly from all predefined sprite costumes.



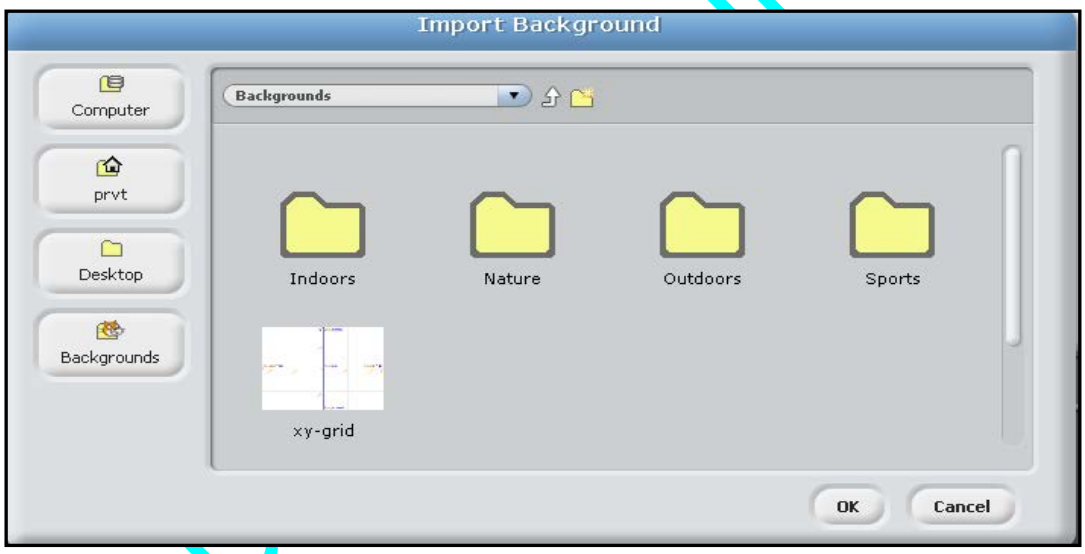
When you click on the green flag, the stackable blocks that appear in the code screen will be executed. The red dot will terminate the execution of the task.
The control button has a block called the “When Clicked” that performs the same task as the green flag.

How to select a stage





A stage is the background of the project.
You have a choice to paint, import or choose a stage from a camera.
The import button has predefined stage folders that look like this.



You can choose a stage from the different folders or from another file destination.

Scratch Block Types

There are essentially five types of blocks namely the trigger blocks, the command block, the function block, the control structure block and the decision block. Examples of these blocks are:

	Trigger block
	Command block

	Function block
	Control Structure block
	Decision block

Trigger blocks

These blocks are used to start the execution of a project or script. Examples are:

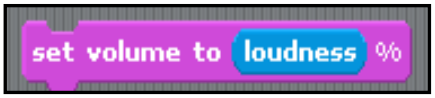
Statement or Command blocks

A statement is an instruction given by the programmer indicating what needs to be performed. A group of statements indicates the sequence in which the project must be executed. A statement has a stackable block that has a specific shape. Examples of statements are:

Function blocks

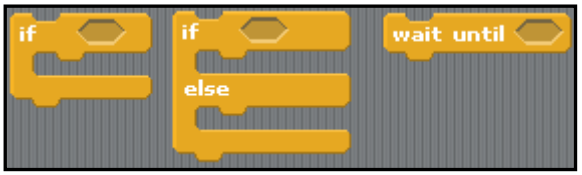
These blocks are used as part of other blocks to carry out a specific function. Examples are

An example of how it can be used.



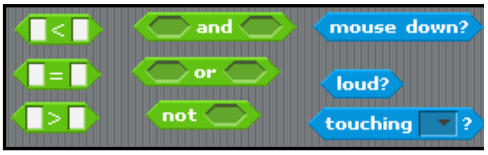
Control Structure Blocks

These blocks are used to control a decision or a condition. The blocks have a slot that fits a decision block. Examples are:



Decision blocks

These blocks are used as part of other blocks for decision / checking purposes. Examples are:



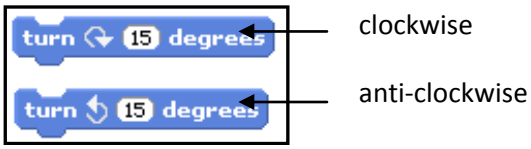
MOTION

When the flag is clicked, the sprite will move 15 spaces forward.

When the flag is clicked, the sprite will move 20 spaces backwards.

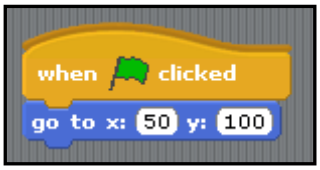
When the flag is clicked, the sprite will move 10 spaces forward and turn clockwise 15 degrees

To change direction



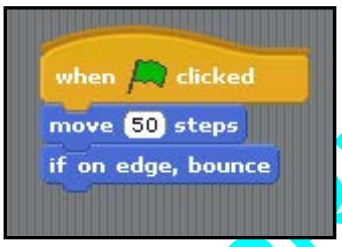
The sprite will move 25 spaces forward in the direction of the mouse.

The sprite will be positioned at the co-ordinates given below, that is at x = 50 and y = 100.

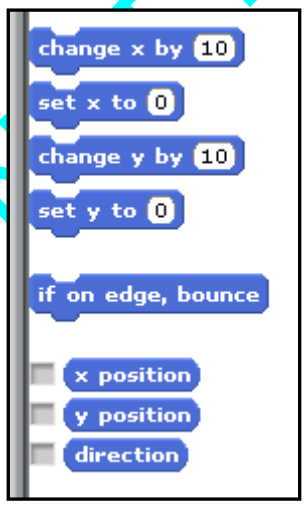


The sprite will glide to the specified co-ordinates 1 second at a time.

Use the “if on edge, bounce” block to keep the sprite on the screen.



To set x and y positions



SOUND

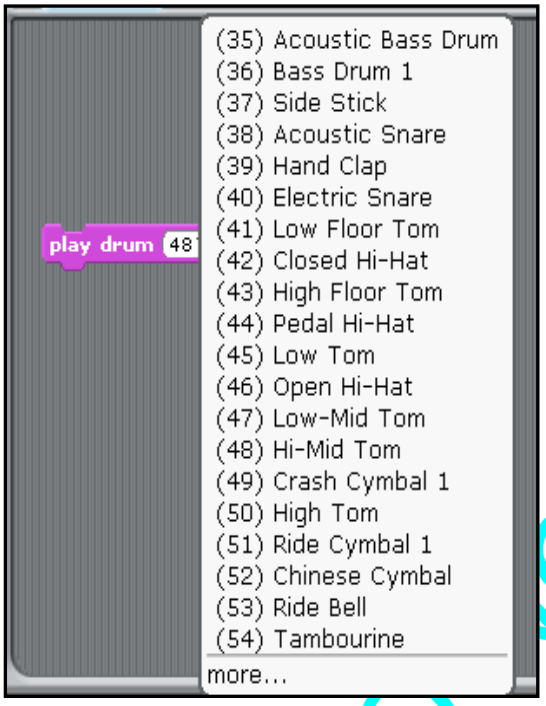
Adding sound to a project

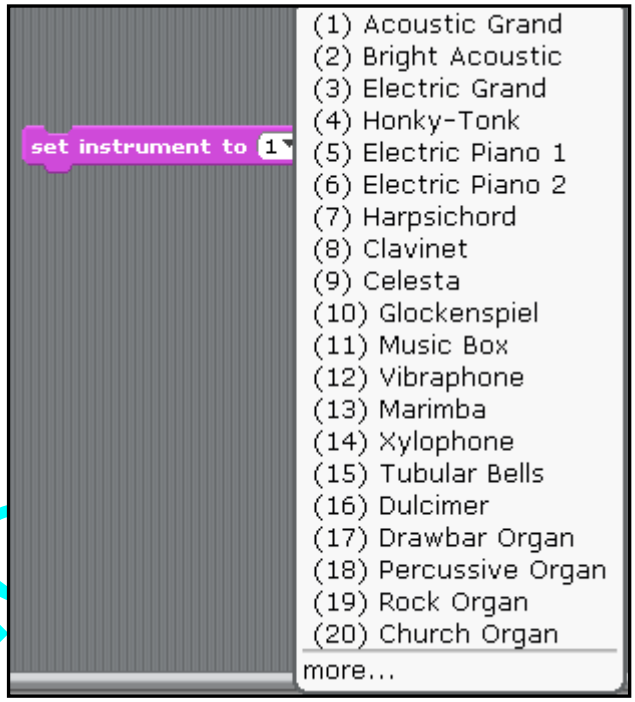
Select the sound button on the Buttons screen to choose the required blocks.You can import or record sound using the following buttons on the code screen.

Alternatively, you can record using the following block. 

The following block can be used to stop all sound. 

You can choose a drum sound or an instrument sound using the following blocks.



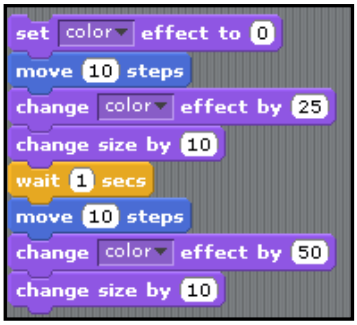


The numbers represent a tune of music. Select the required number for the desired music.

LOOKS

This button is mainly used to decide on the appearance of the sprite. It can change the colour, position, size and the costume of the sprite.

Build the following script to see the changes in colour of the sprite.



Example 1: Choose Sprite1 as boy4-walking-a. Add two costumes boy4-walking-c and boy4-walking-e. Use the sprite and the costumes to make the boy walk 3 steps. In order to create new costumes, click on Costumes and then import.

Change the number of seconds from 1 to 3 to see the boy walking slower. Change the number of seconds from 1 to 0.5 to see the boy walking faster.

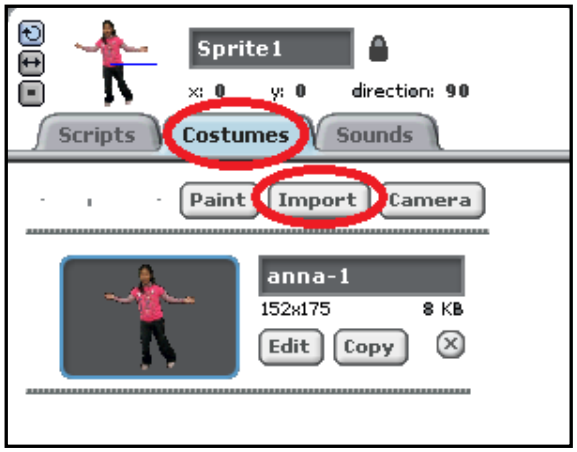
Example 2:

Sprites can change from one to another by simply changing the costume.

This project is to create an exercise routine for Anna.

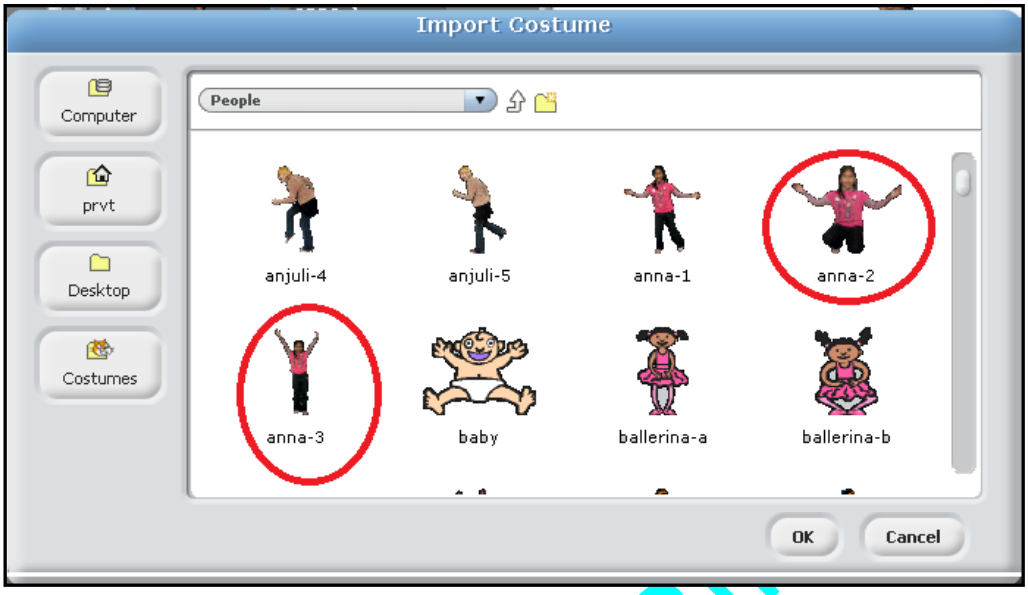
Choose “anna-1” from the People category as a new sprite.

We want to change her costume to appear as if she is doing an exercise routine. We would have to decide how many new appearances she needs to have. Let’s say we want Anna to do five different types of movements. That means we need five costumes. In order to create new costumes, click on Costumes and then import.

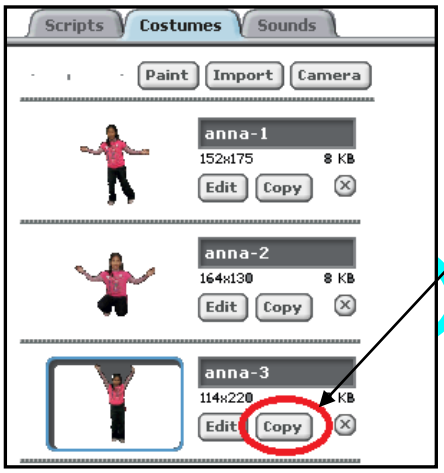


Note that you can choose only one costume at a time.

Step 1: Gather all the sprites and the costumes.



Select “anna-2” and click on the OK button. Click on import again, select “anna-3” and click on the OK button. By now you will have three costumes for Anna.



A duplicate copy of “anna-3” will appear. Now click on edit.

Click here to flip Anna horizontally.

Click on the OK button

You now have costume number 4.

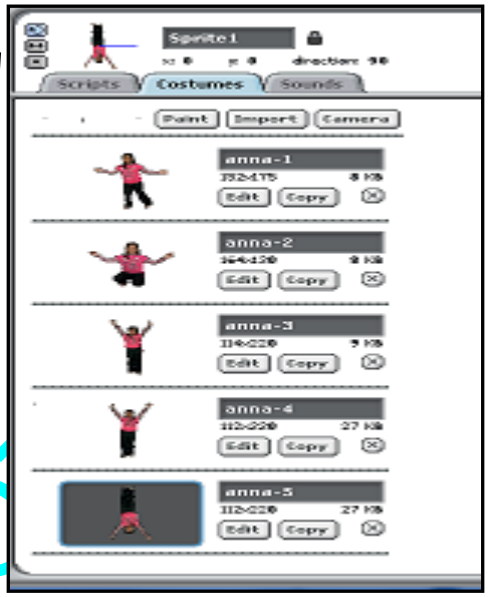
We now have 4 costumes. Click on copy again. Then click on edit.

Click here to flip Anna vertically.

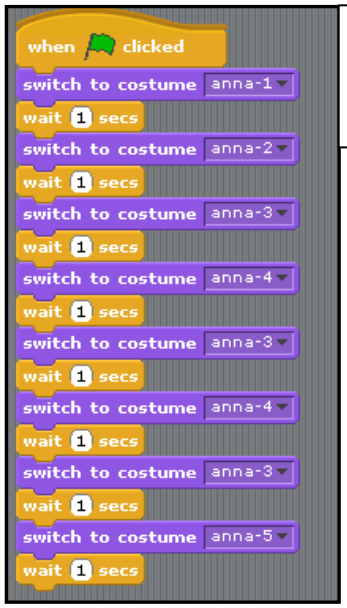
Click on the OK button

You now have costume number 5.

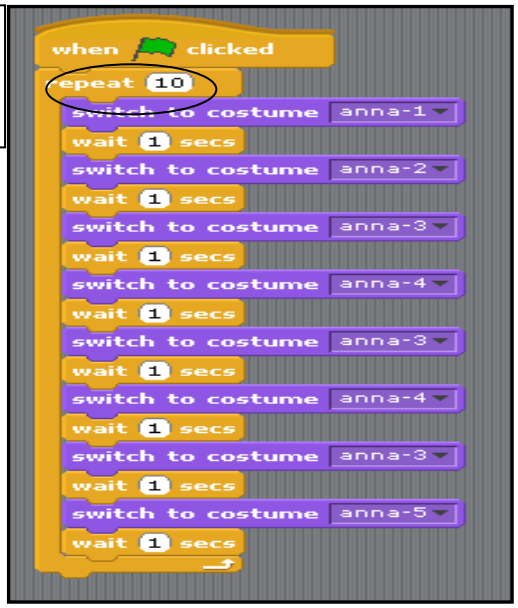
Here are your 5 costumes.



Step 2: Write a script for Anna so that she can do her exercise routine.

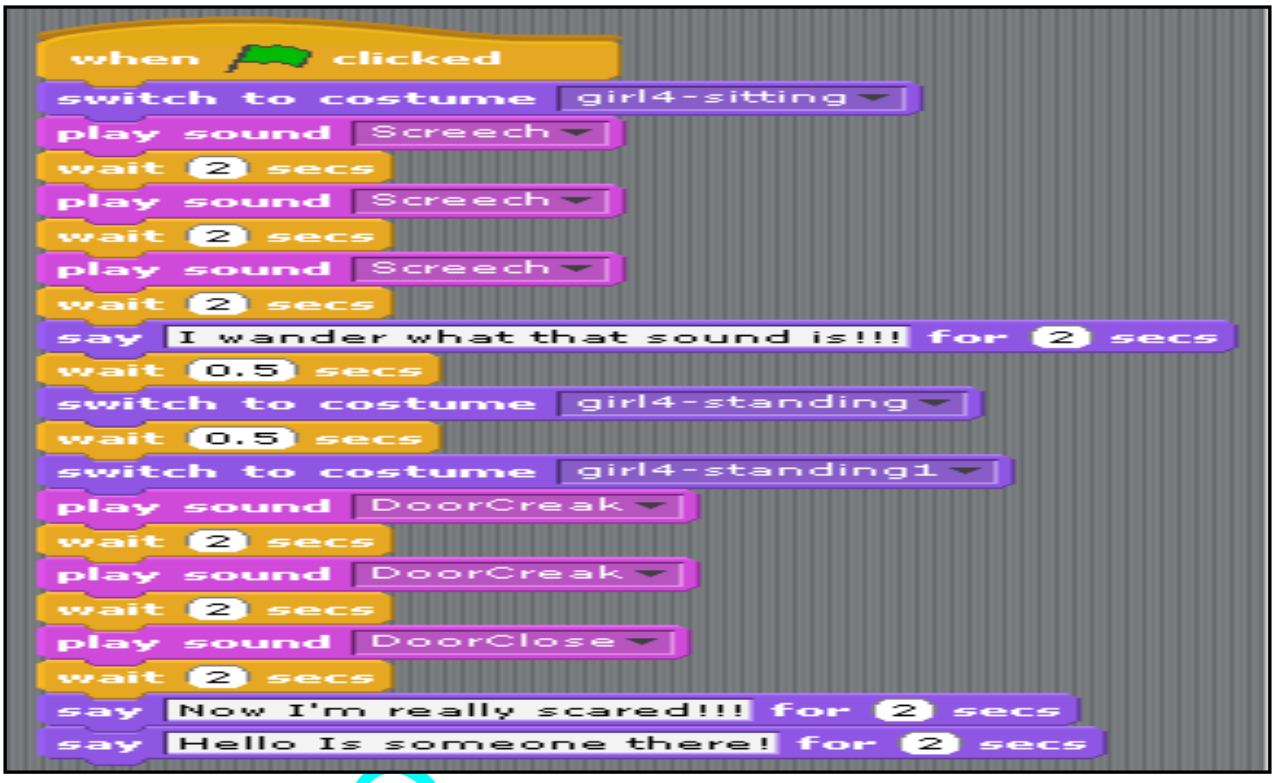


To repeat this routine 10 times, add a repeat block from the category.



Animation

Write a script that creates a scary atmosphere. I have used girl4-sitting  as my sprite. My costumes are girl4-standing  and girl4-standing1  where I have edited the direction of girl4-standing. Remember to import the sounds before you start the script. Try this script and add to it to make it scarier.



Variables

The main purpose of creating a variable is to store data for later use. When a variable is created, a small memory box / place holder appears with that variable name. Once the memory box is created, we can assign a value to the space in that memory box. These memory boxes can hold onto only one value at a time. The value of a variable can be changed whenever required.

Variable Naming conventions

A variable name must be meaningful and appropriate. The variable name must indicate the type of data the variable will hold. It is important to name the variables appropriately so that you can remember which variable to use when your projects get bigger and you need to use a lot of variables.

Examples of appropriate variable names would be:

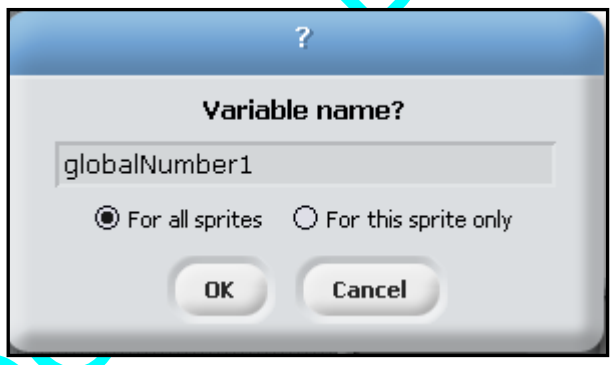
Variable Name	Description
name1	To store the first name
name2	To store another name
number1	To store the first number
number2	To store the second number
totalValue	To calculate and store the total of number1 and number2
age	To calculate and store the age of a person
average	To calculate and store the average value

Avoid using a, b, c etc as variable names as these can be confusing when the program gets larger and more complicated.

Global vs Local variables

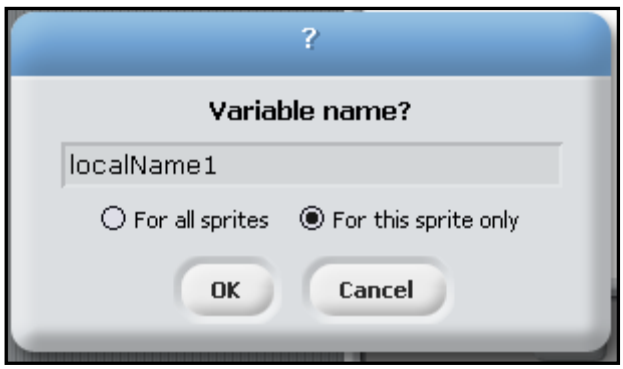
A global variable means that the value of the variable can be used and changed by ***any sprite*** on the stage. When a variable is created, it is a global variable by default.

Example: We want to create a variable called globalNumber1 to be used by all sprites.



A local variable is personally created for a ***specific sprite***. This variable can be used or changed only by this sprite. A local variable is also known as an object specific variable.

Example: We want to create a local variable called localName1 to be used by Sprite1.



Overview of Data Types:

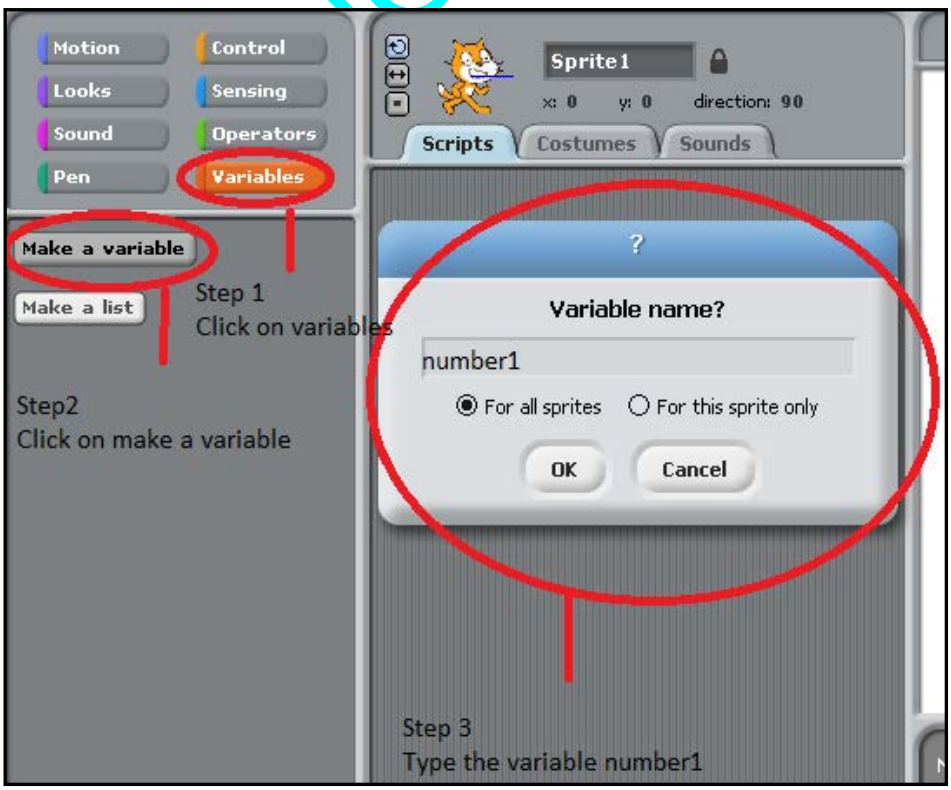
Data types specify the type of data that can be stored in a variable.

Values can either be numbers generally stored as integers or decimals numbers, or they can be text / string / alphanumeric that is it consists of numbers, letters and special characters. Floating-point representation supports a much wider range of values than integer. It also offers more precision and accuracy than integers. An example of a floating point data type is a real number better known as “double” in the programming environment.

Data Types	Description
Integer	Whole numbers
Float	Allows for the use of whole numbers and decimal numbers.
String	Allows for the use of alphanumeric data as well as special characters and spaces.
Boolean	The data can either be true or false

Scratch however does not allow us to choose the data type of a variable. When you create a variable in Scratch, it takes on all data types, however when we move to a higher programming language, a variable takes on a specific data type.

Making / Creating a variable



Once the variable is created, the variable is represented on the stage in one of three forms. A variable can be viewed in different forms called normal form, large form and slider form. The desired form can be selected by double clicking on the current form of the variable on the stage.

Form	Appearance
Normal form	
Large form	
Slider form	

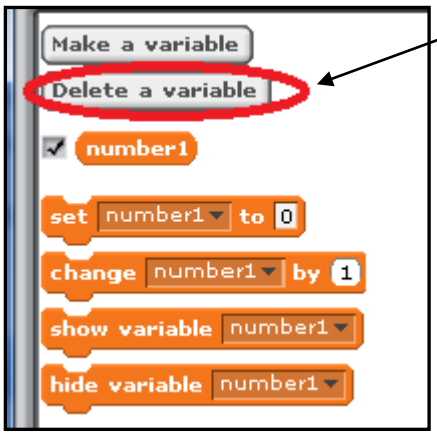
As soon as you create a variable, five blocks will appear under it in the blocks area.

	Reports the value for number1
	Stores a value for number1
	Changes the value for number1
	Shows the value for number1 on the stage
	Hides the value of number1 on the stage

To ***show*** a variable on the stage, the checkbox must be ticked. 

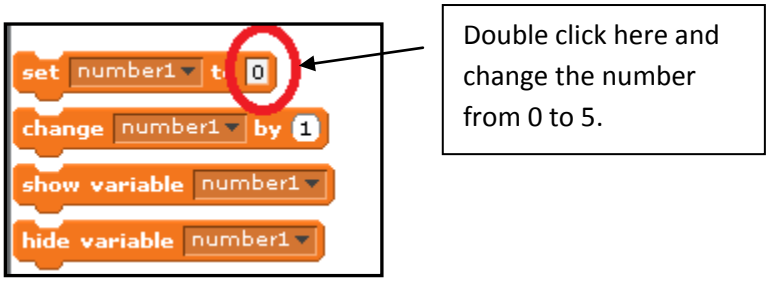
To ***hide*** the variable or remove the variable from the stage, the checkbox must be un-ticked.

Deleting a variable

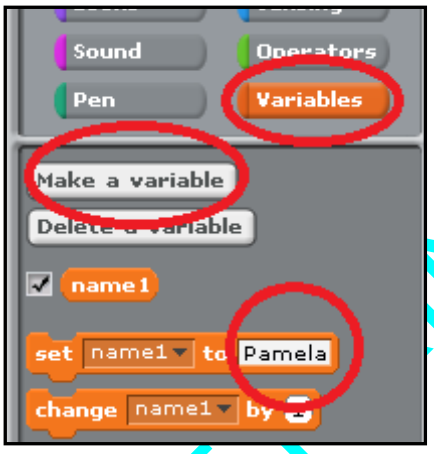


Click here and enter select the variable to delete

Assigning values to variables



Create a variable called “name1”. Assign this variable the name “Pamela”.



<u>Operators in Scratch</u>

Operators according to their order of priority, (highest to lowest).

Mathematical Operators

Operator	Symbol	Illustration
Brackets	( )	
Multiplication, Division, Modulus	* / mod	
Addition, Subtraction, String concatenation	+ - join	
Comparison	> < =	
Logical AND	and	
Logical OR	or	
Not	not	

Functional Operators

Round(),	Round to the nearest whole number	
Random()	pick random number from 1 to 10 inclusive	
Length()	length of "world" is 5	
Extracts a single letter() from a word	Letter 1 of world is "w"	
Sqrt()	Finds the square root of a number	

Modulus which is illustrated by mod is a remainder function.

When two numbers are divided, mod keeps the remainder after the division is done.

After the division is done, the answer is ignored and mod keeps the remainder.

Example 1: 10 mod 3 = 1 ..... the answer to the division is 3 and the remainder is 1
Example 2: 12 mod 9 = 3 ..... the answer to the division is 1 and the remainder is 3
Example 3: 100 mod 40 = 20.... .the answer to the division is 2 and the remainder is 20
Example 4: 18 mod 2 = 0 ..... the answer to the division is 9 and the remainder is 0
Example 5: 29 mod 6 = 5 ..... the answer to the division is 4 and the remainder is 5

Examples: Remember that we need to follow the order of priority.

1) 8 / 2 * 3 – 1 + 2
= 4 * 3 – 1 + 2
= 12 – 1 + 2
= 11 + 2
= 13

2) 10 - 8 mod 6 + 7 * 2
= 10 – 2 + 7 * 2
= 10 – 2 + 14
= 8 + 14
= 22

3) 24 - (3 + - 7) + 5
= 24 - (- 4) + 5
= 24 + 4 + 5
= 33

4) 20 * 2 / 10 + 19 mod 8
= 40 / 10 + 3
= 4 + 3
= 7

5) 100 / 50 * 10 mod 8
= 2 * 10 mod 8
= 20 mod 8
= 4

6) 30 – (27 mod 9 * 3 -1)
= 30 – (0 * 3 - 1)
= 30 – (0 – 1)
= 30 – (-1)
= 31

Exercise on Operators:

1. 67 – (4 mod 2) * 9.6
2. 80 mod 9 / 4 + 143 mod 12
3. 91 – 64 / 8 * 2 + 7
4. 99 / 11 mod 7 * 3
5. (72 / 9 + 3) mod 8 * 0
6. -8 * 7 / 4 + 6 * -7
7. 11 * (9 / 3) – 6 *2 +1
8. 16 mod 4 / 2 * 2
9. (8 / 8) * 7 mod 5 / 1
10. 100 – 50 / 25 * 20 +(90 mod 8 / 2)

Boolean Operators (True can also be represented by 1 and False by 0)

True	AND	True	True
True	AND	False	False
False	AND	True	False
False	AND	False	False
True	OR	True	True
True	OR	False	True
False	OR	True	True
False	OR	False	False
NOT (True)	False		
NOT(False)	True		

Examples on Boolean operators used for comparison.

1) 10 > 8 AND 2 = 1
= True AND False
= False

2) 100 < 99 OR 100 > 99 AND 100 = 99
= False OR True AND False
= False OR False
= False

3) 81 mod 9 = 1 AND 3 = 5 - 2
= False AND True
= False

4) 5 > 4 OR 6 = 5 OR 5 > 6
= True OR False OR False
= True

5) 5 > 4 AND 6 = 5 AND 5 > 6
= True AND False AND False
= False

6) 23 > (100 / 5) AND 50 mod 2=0 AND 45 = 55 – 10
= True AND True OR True
= True

7) Not (3=3)
= Not (True)
= False

8) 8 mod 5 = 1 AND Not (7 > 8)
= False AND Not(False)
= False AND true
= False

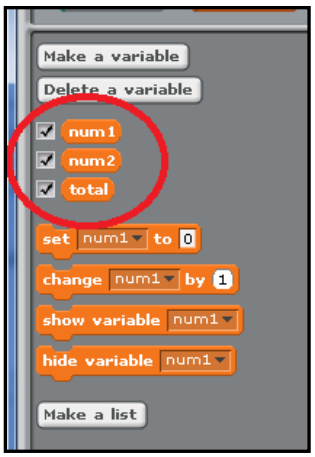
Exercise

1) 3 = 2 AND 4 = 3 AND 6 =6
2) 3 =2 OR 4 = 3 OR 6 =6
3) 24 mod 2 = 0 AND 18 mod 3 = 0 AND 8 > 7 mod 5
4) not (85 / 5 > 100 / 10)
5) 67 mod 8 < 5 AND not (80/10 > 12)
6) 16 <(33 mod 16) OR 7 <6

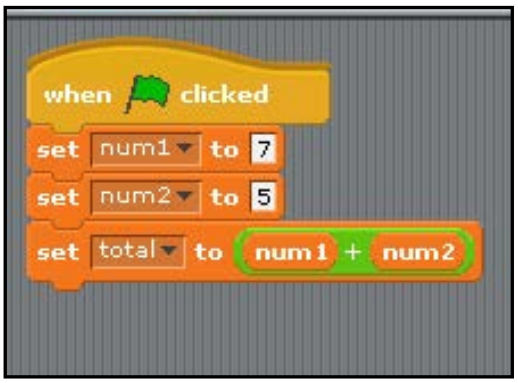
Example 1

Create a project that makes three variables num1, num2 and total. Assign the value 7 to num1 and 5 to num2. Calculate and display the total.

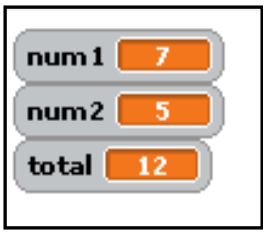
Step 1 (Make 3 variables)



Step 2 (Write a script)



Step 3 (Click on the green flag)



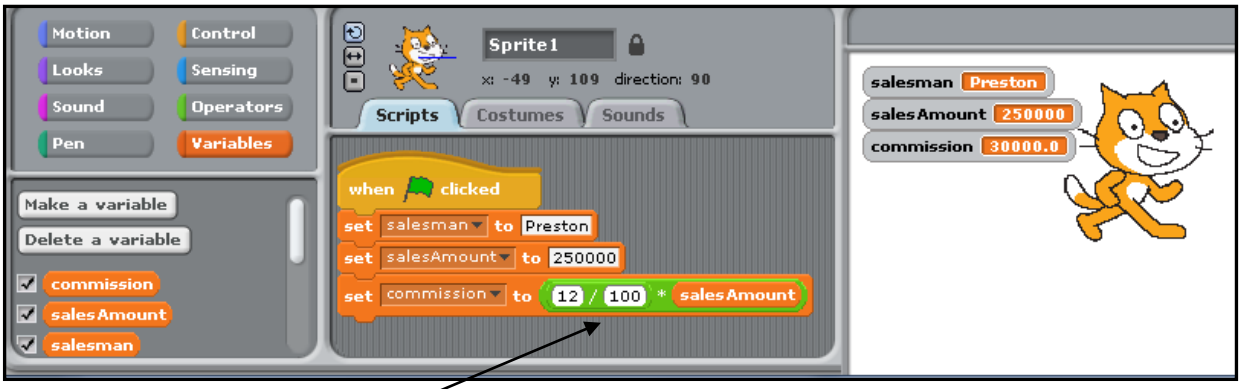
Example 2

Create a project that makes three variables length, breadth and area. Assign the value 10 to length and 5 to breadth. Calculate and display the area of the rectangle.

Example 3

A salesman at a furniture store earns a 12% commission based on the amount of sales achieved for the month.

Create a project that uses variables for the following:
salesman, salesAmount and commission.
Assign the name “Preston” to salesman and the value 250000 to salesAmount.
Calculate and display the commission.



Can be done in two steps

Drag 

, then add 

,

place this block 
here 

to look like this 

Exercise:

1. Create a project to use variables breadPrice, milkPrice, juicePrice and totalPrice.

   Set breadPrice to 9.90, milkPrice to 11.99 and juicePrice to 14.50.

   Calculate and display the total price.

2. Create a project to use variables length, breadth and height.

   Set length to 25, breadth to 18 and height to 7.

   Calculate and display the volume of the rectangle.

3. Create a project to use variables penPrice, pencilPrice, penQuantity, pencilQuantity, totalPens, totalPencils and totalPensAndPencils. Use the following values to set the variables:

Variable	Value
penPrice	3.99
pencilPrice	2.59
penQuantity	35
pencilQuantity	30
totalPens	penPrice * penQuantity
totalPencils	pencilPrice * pencilQuantity
totalPensAndPencils	totalPens + totalPencils

Calculate and display the values for the following: totalPens, totalPencils and totalPensAndPencils

4. Create a project to use variables employeeName, hoursWorked, ratePerHour, basicSalary, tax, medicalAid, bonus and finalSalary

Use the following values to set the variables

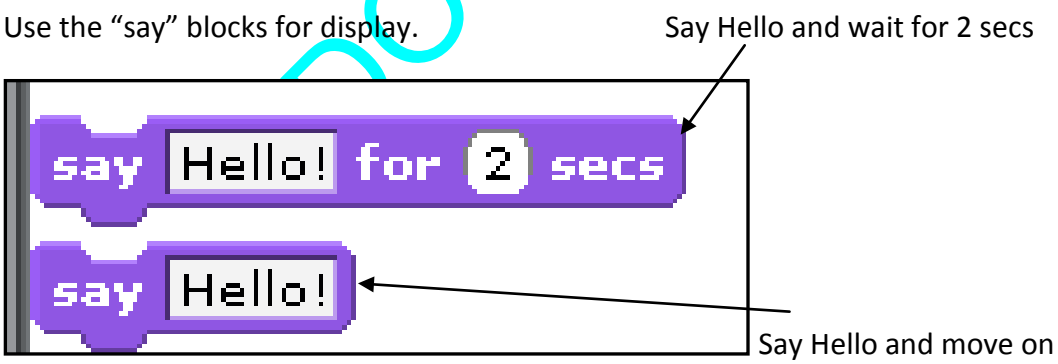
Variable	Value
employeeName	Peter Mathers
hoursWorked	52
ratePerHour	150
basicSalary	hoursWorked x ratePerHour
tax	14 / 100 * basicSalary
medicalAid	950
bonus	10 / 100 * basicSalary
finalSalary	basicSalary + bonus - tax - medicalAid

Calculate and display values for basicSalary, tax, medicalAid, bonus and the finalSalary.

5. The South African Youth Alliance is considering sending their members to the World Youth Alliance Conference in America. The cost of the ticket is R10000, the cost of food and accommodation is R6000 each and each member must carry an extra R5000 for spending. There are 9 grade 8 members, 6 grade 9 members and 14 grade 10 members. Each member is awarded a 30% bursary.
Calculate and display the total required to be paid in by the Grade 8's, Grade 9's and the Grade 10's separately. (Do not add in the bursary amount).
Also display the total amount required by all members before and after the bursary.

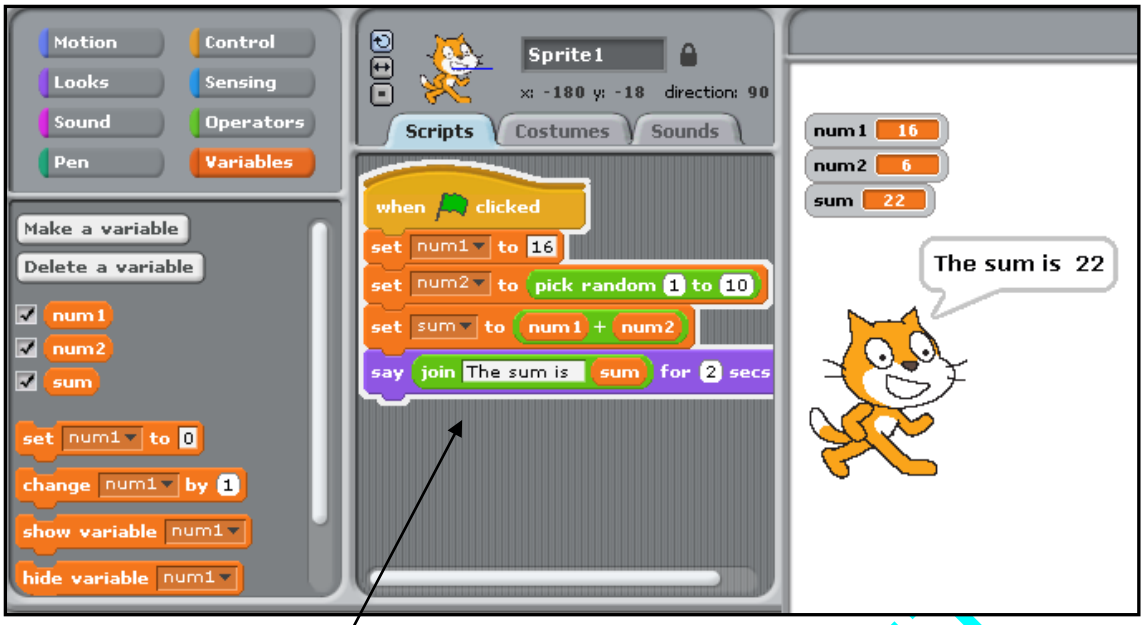
Displaying messages

Use the "say" blocks for display.



Example 1

Create a project to make three variables called num1, num2 and sum. Set num1 to the value 16. Randomly generate a number from 1 to 10 for num2. The sprite must display the sum of num1 and num2 using a suitable message.



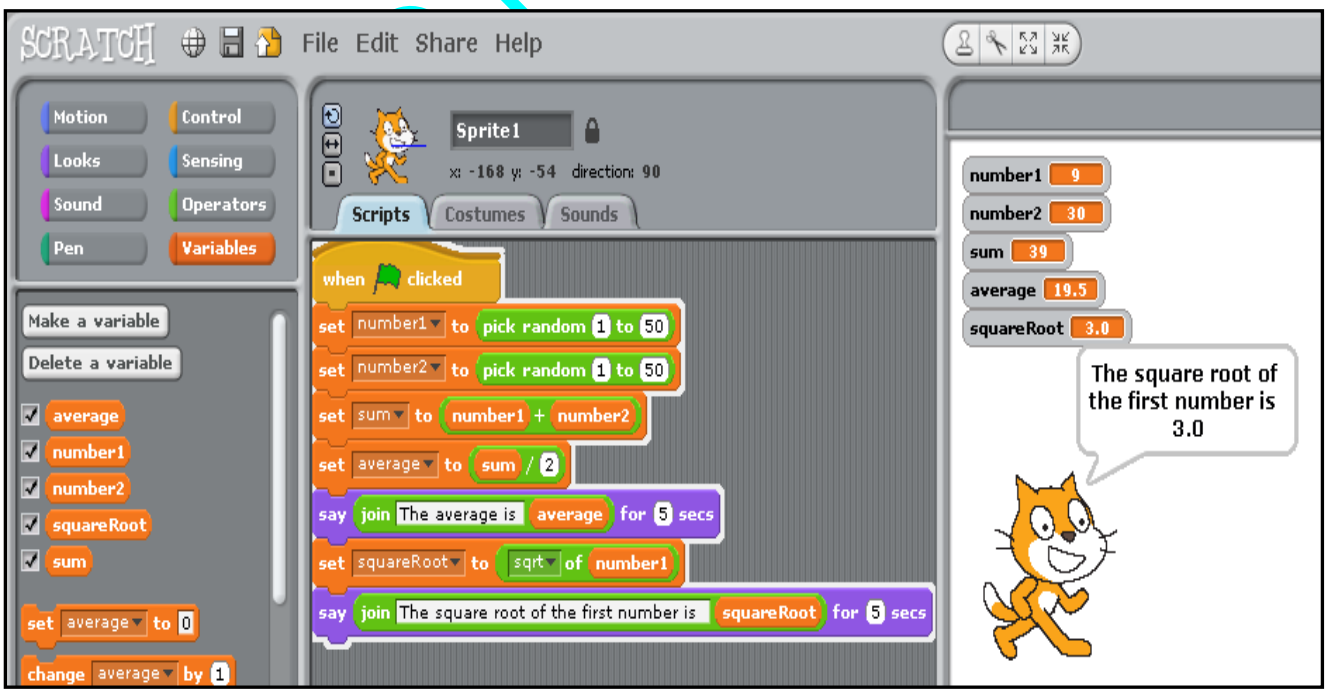
Example 2

Notice the join is used to display a message that requires two or more things joined.

Create a project to randomly generate 2 numbers in the range 10 – 50. The sprite must display the following after 5 seconds each.

2.1 Display the average of the two numbers.

2.2 Display the square root of the first number



Note that the first screen output would have displayed the average. This is the second screen output.

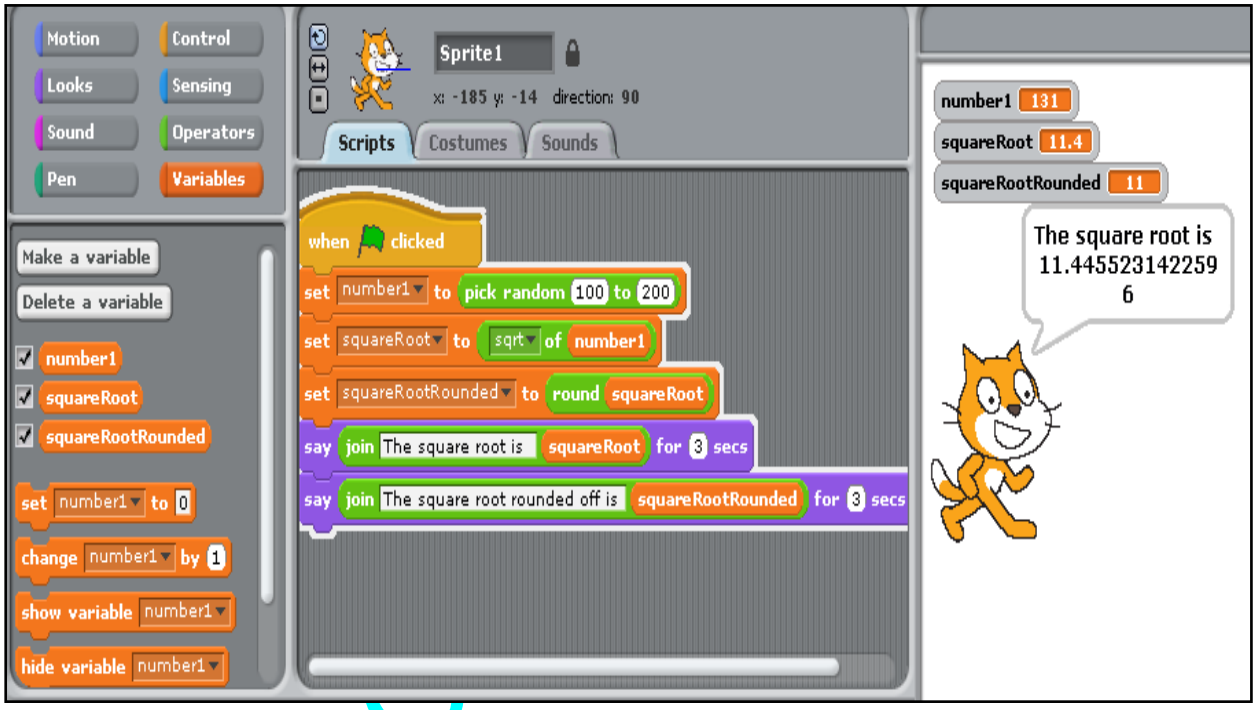
Example 3

Create a project to randomly generate a number in the range 100 – 200. Display the following:

3.1 The square root of the number.

3.2 The square root rounded off to the nearest whole number.

Display both answers using suitable messages for 3 seconds.



The square root rounded off should appear on the second screen.

Exercise

1. Create a project to work out the cost of carpeting of a rectangular room.

   The dimensions are 8m by 5m. The cost per square metre is R120.00.

   Display the area of the room and the cost of carpeting using suitable messages.

   Each message must be displayed for 2 seconds.

2. You have invited your friends over for a get together. The number of friends that attended is 28. You will be serving pizza and would need to decide the number of pizzas to buy if each friend eats an average of three pieces. Each pizza has eight pieces.

Create a project that will display a suitable message indicating the number of pizzas to purchase if each person eats at least three pieces.

3. It is movie day at Gateway Shopping mall and dad has decided that he will take the family to watch the movie titled “Twilight”. The cost of the adult ticket is R52.00 and children under the age of 12 pay half the price. My mum, dad, my brother (10 years old) and myself (15 years old) will be attending the movie. Calculate and display using a suitable message, the total amount payable by my family.

4. Design a program to be used as a cash register slip. The following items were purchased.

Four toothbrushes @ R49.99 each

Two tubes of toothpaste @ R15.99 a tube

One bottle of gargling fluid @ R59.99 and

One spearmint floss @ R29.99

Calculate and display the total cost, vat at 14 %, and final total.

Calculate and display the change if R400 was tendered.

5. Create a project to enter two numbers. Display the square of the first number and the square - root of the second number.

Keyboard input

The following blocks are used to allow the user to input data via the keyboard or mouse.

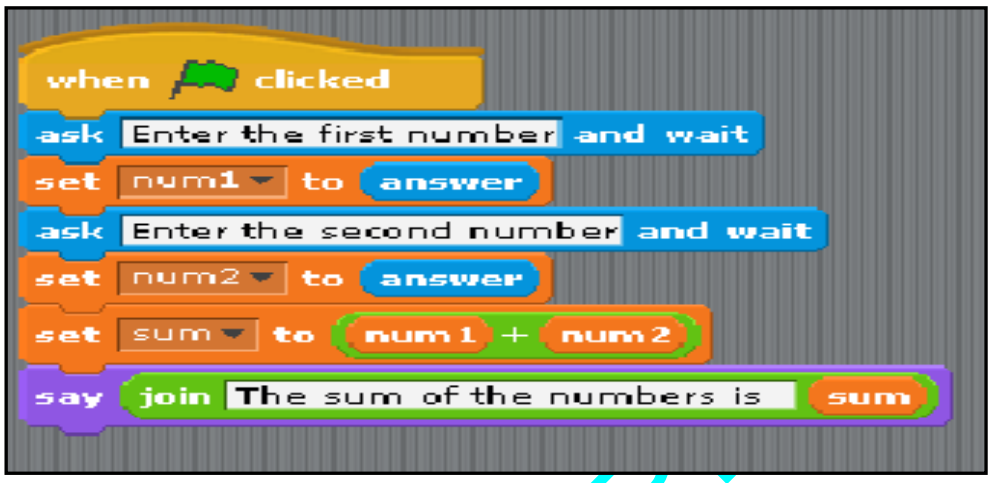
Block	Button Category	Description
	Sensing	Asks a question and waits for the user to type in the response
	Sensing	Stores the users answer in a memory box called “answer”. Can store only one value at a time. If a second question is asked, the answer to the first question will be overwritten.
	Control	The answer can be saved in another variable if you need to use the variable “answer” again.

Example 1

Create a project to input two numbers and display the sum of the numbers.

Create three variables num1, num2 and sum.

The script should appear as follows.

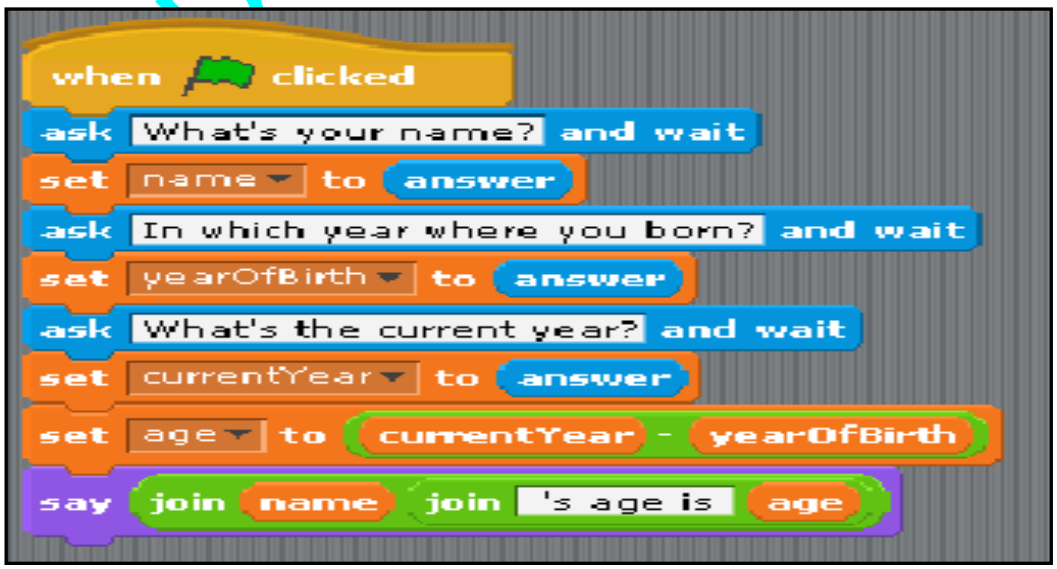


Example 2

Create a project that makes the following variables.

name	Stores the name of the user
yearOfBirth	Stores the year the user was born
currentYear	Stores the current year
Age	Calculates the age of the person

The script should appear as follows.



Example 3

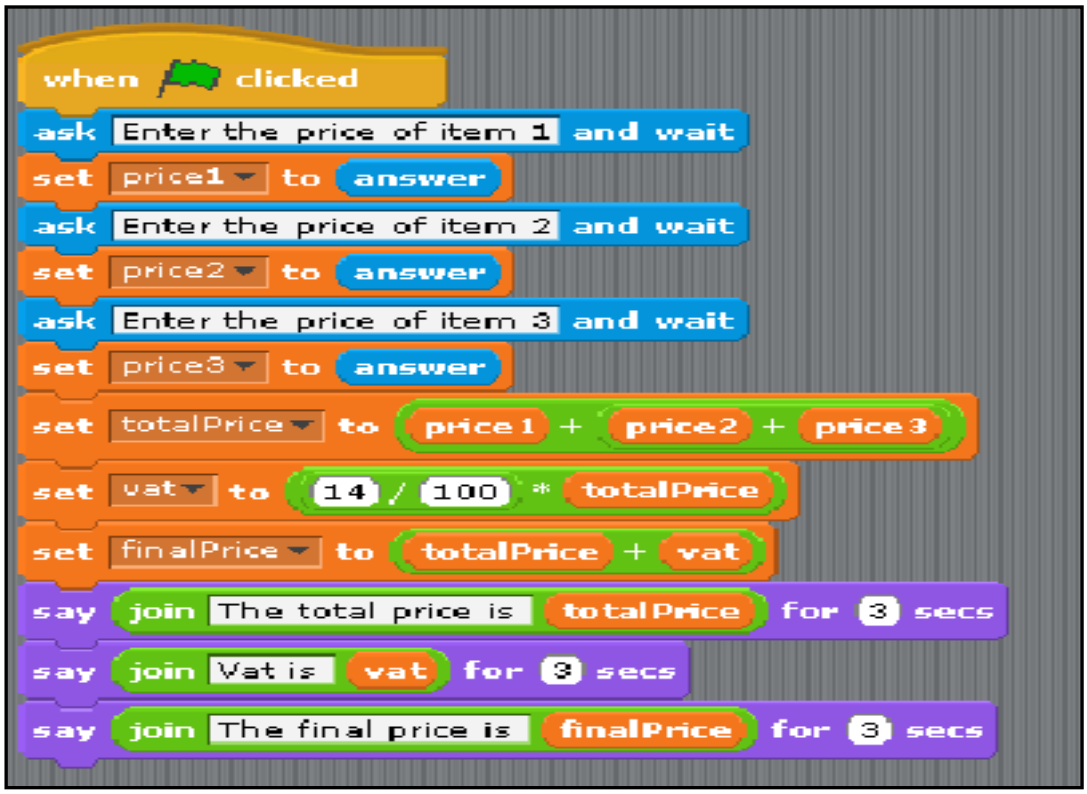
Create a project to simulate a cash register slip. Allow the user to enter the price of three items. Calculate and display the total price. Calculate and display 14% vat. Calculate and display the final price consisting of the total price + vat.

I will create the following variables to store my values namely:

price1, price2, price3, totalPrice, vat and finalPrice

Allow for 3 seconds between each display message.

The script should appear as follows:



Exercise

1. Create a project to enter your English marks for the four terms. Calculate and display the average mark for English.

2. Create a project to enter the dimensions of a rectangular ground at your school. Calculate and display the perimeter of the ground. Calculate and display the cost of fencing the ground if the price of fencing is R198 per metre. The display statements must have suitable messages and each message must be displayed for 3 seconds.

3. Create a project to enter the name and the sales made by an estate agent for January. Calculate and display using a suitable message the agents' commission based on 20% of the sales.

4. Create a project to enter the name of your school. Randomly generate a value in the range 800 – 1000 to indicate the number of learners at your school. Allow the user to enter the rate of school fees per learner per year. If every learner pays the full school fees, calculate and display the amount of school fees collected by the school for the year.

5. Create a project to enter a number. Calculate and display the following using suitable messages. Each message must be displayed for 2 seconds.
   5.1 the square of the number
   5.2 the square root of the number
   5.3 The square root of the number rounded off to the nearest whole number
   5.4 the number that precedes this number (precedes means before)
   5.5 the number that follows this number

6. Create a project to enter two names and a mark for each name. Let us assume that the marks were captured incorrectly and you need to swap the marks.
   Example: You entered name1 to be Leeanne and Leeanne's mark is 67.
   You entered name2 to be Letisha and Letisha's mark is 91.

   Your program must swap the marks to display the following.
   Leeanne's mark is 91
   Letisha's mark is 67

7. Create a project to enter the length, width and height of two cylinders separately. Calculate and display the volume of each cylinder using suitable headings.

8. Statistics have shown that the number of voters in SA range in the following age categories:
   18 – 25 : 20%
   >25 – 40 : 50%
   >40 and over : 30%

   Calculate and display the number of voters in each age group if there are 27 000 voters.

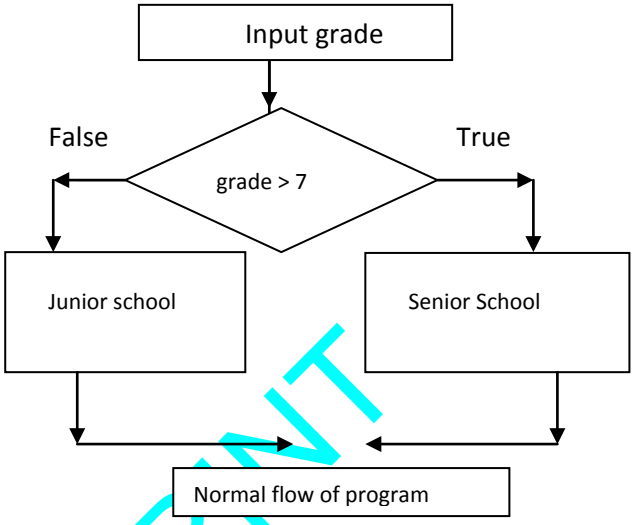
9. There are 4 telephone networks used in SA namely P, Q, R and S.
   Network R has 12% more users than network P.
   Network P has 7% fewer users than network S.
   Network S has 3 000 users.
   Calculate and display the number of users for each network if there are 40 000 users.

Conditional Statements / Structures

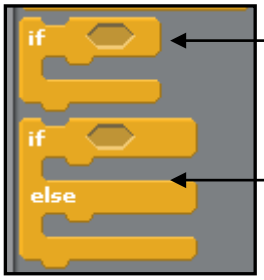
There will be projects that requires for you to make a decision between this or that, meaning that code will be executed according to the result of a decision

```
Input grade

If  grade >  7

   Output Senior School

Else

  Output Junior School
```

Flowchart representation



The following blocks from the Control category can be used to carry out the decision.

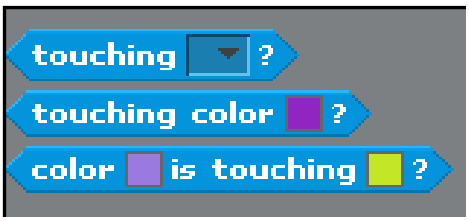


Used to check for a single outcome. Example: If the number is more than 0, display “Positive”

Used to check for more than one outcome. Example: If the number is more than 0, display “Positive” otherwise display “Negative”

NB. Notice the shape of this area 

, it will only allow for blocks of the same shape to fit. These are examples of some of the blocks that will fit.



Later on we will learn about the

Example 1

Create a project to ask the user to enter if they would like to play soccer or tennis. If the choice is Soccer, display a soccer ball. If the choice is tennis, display a tennis ball.

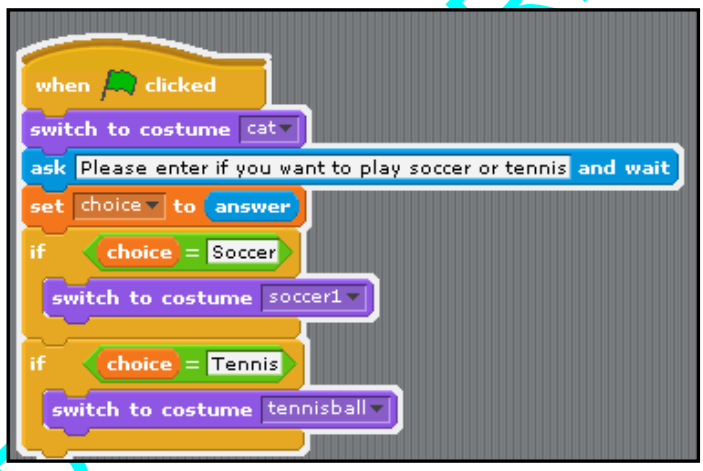
Step 1: Create a variable called choice

Step 2: The cat sprite should appear as a default sprite

Add a soccer ball costume from the “Things” folder.

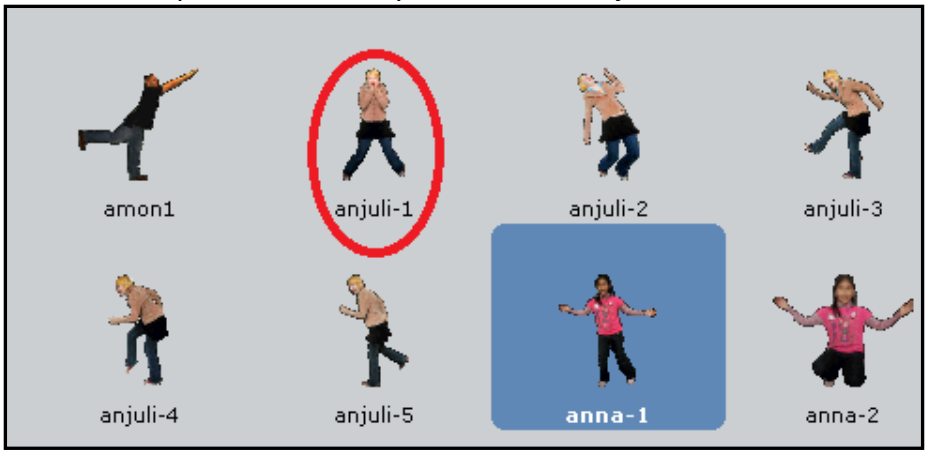
Add a tennis ball costume from the “Things” folder.

Step 3: Write a script that look s like this



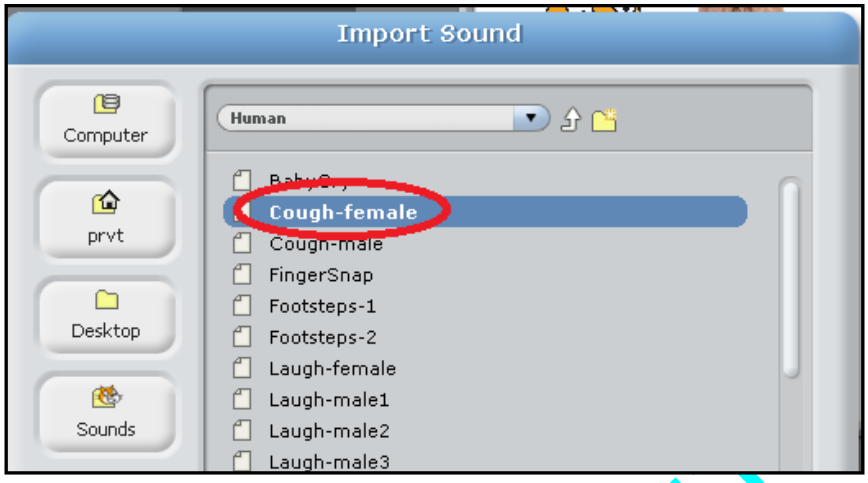
Example 2

Create a project to do the following to play sounds corresponding to the response of the user:
Choose two sprites: The cat by default and “anjuli-1”.



Import the following sounds from the folder called “Human”.

Cough-female, Sneeze-female and Laugh-female



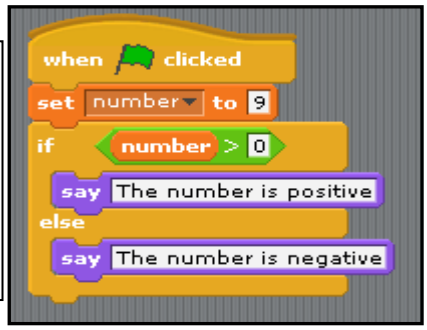
Write a script so that the cat asks “anjuli” how she feels today. If the answer is happy, display the “Laugh-female” sound. If the answer is sick, display the “Sneeze-female” and the “Cough-female” sounds.

Example 3

Create a project to set a number to the value 9. Display a suitable message indicating if the number is positive or negative.

Step 1
Make a variable called number

Step 2
See script ⟶



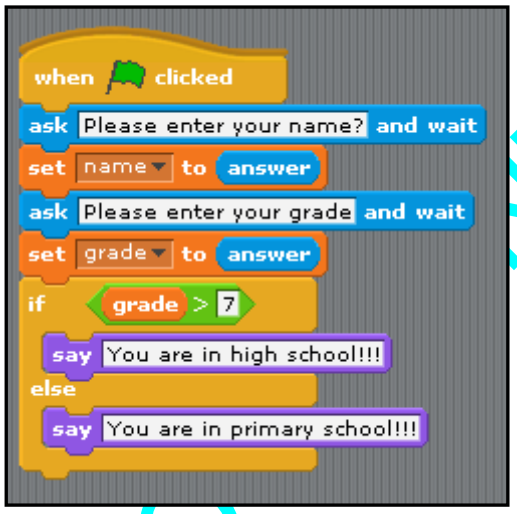
Example 4

Create a project to prompt the user to enter a learners' name and grade from 1 to 12. Display a suitable message indicating if the learner is in high school or primary school.

High school: Grade 8 - 12

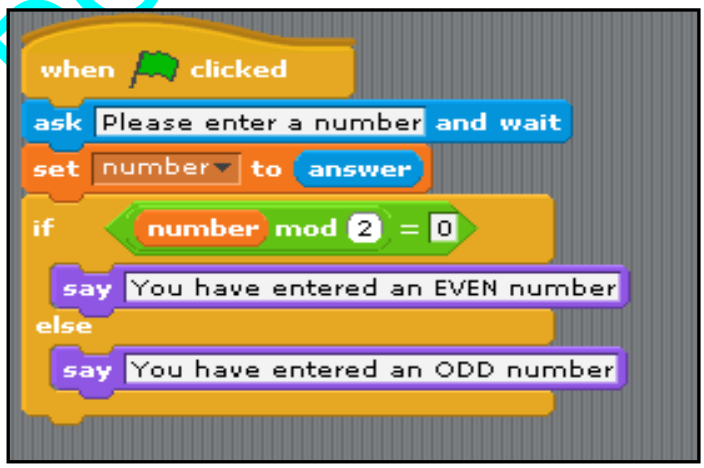
Step 1: Make two variables for name and grade.

Step 2: Write a script that looks like this.



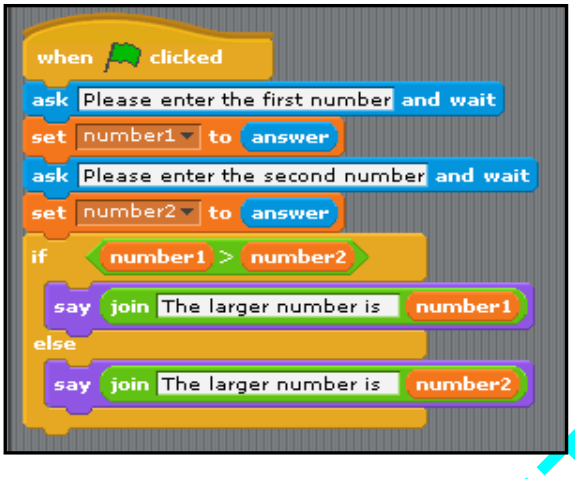
Example 5

Create a project to prompt the user to enter a number. Display a suitable message indicating if the number is even or odd.



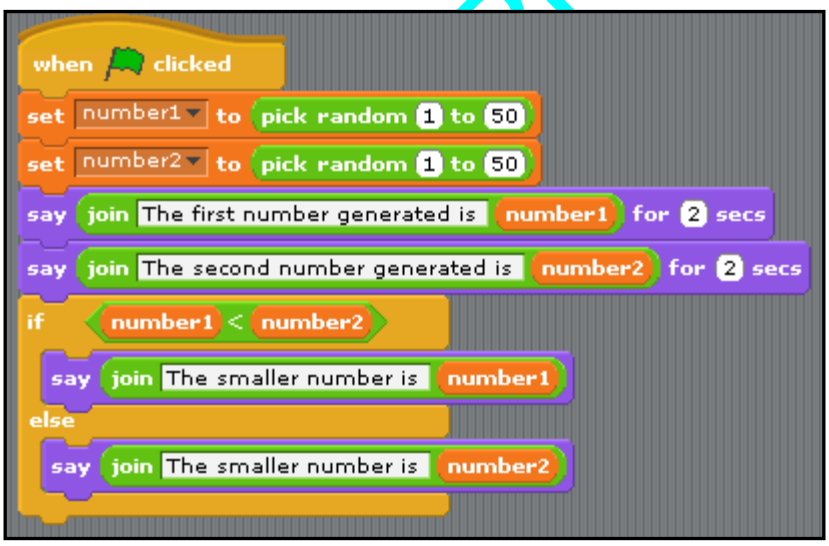
Example 6

Create a project to enter two numbers. Display the larger number with a suitable message.



Example 7

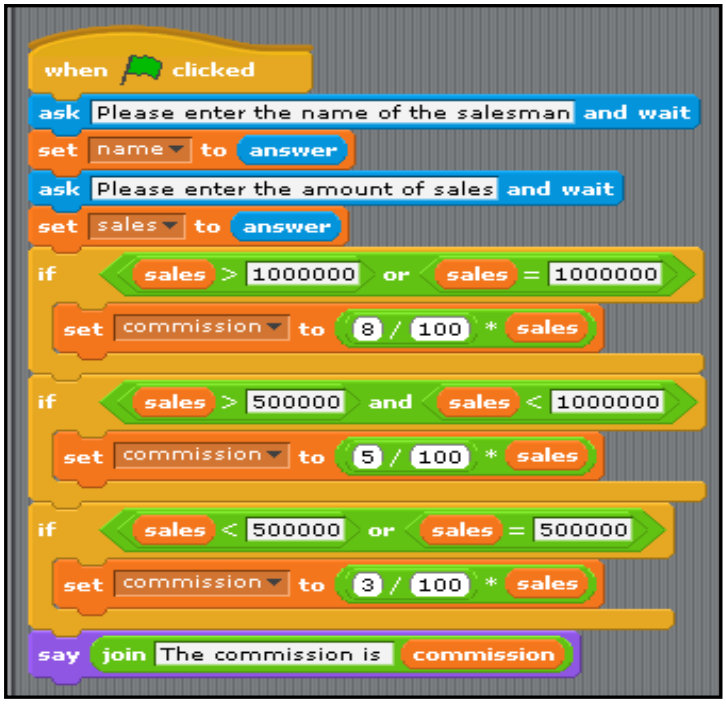
Create a project to randomly generate two numbers in the range 1 – 50. Display both numbers generated and display the smaller number with a suitable message.



Example 8

A salesman earns a commission based on the amount of sales achieved. Create a project to enter the name of the salesman and the sales. Calculate and display the commission as follows:

Sales	Commission
>=1000 000	8%
> 500 000 and < 1000 000	5%
<=500 000	3%



Exercise

1. Create a project to enter your name and your age. Display a message indicating if you are an adult or a child. Adults are 18 years and older.
2. Create a project to randomly generate two numbers in the range 100 – 999. Display the larger of the two numbers.
3. The price of a movie ticket is R52.00. Your project must enter the number of movie tickets required. Calculate and display the total. A 5% discount is awarded if the number of tickets required is greater than 4. Display the discount and the final price of the tickets.
4. Create a project to enter the amount spent at a supermarket. A discount is awarded as follows:

Amount	Discount
>=5000	10%
>2500 AND < 5000	8%
>=1000 AND <=2500	5%
<1000	0

Display the original amount, the discount and the final amount.

5. Create a project to enter two numbers called num1 and num2. Determine and display using a suitable message if num1 is a factor of num2.

   Example 1: num1 = 3 and num2 = 6. The display will be as follows:
   3 is a factor of 6.

   Example 2: num1 = 7 and num2 = 20. The display will be as follows:
   7 is not a factor of 20.

Event Handling

An event can be seen as an instruction that needs to be carried out. Event handling is a process whereby an object responds to an instruction triggered off either by the user or by another part of the program.

Reasons for event handling

- Communication between sprites
- Synchronisation to coordinate multiple tasks
- Sending / Receiving messages

"Broadcast" blocks are used to signal or trigger events while "When" blocks are used to handle or carry out the event.

Block	Description
	The event that must be carried out if the space key is pressed. The arrow keys, letters and numbers can also be selected by clicking on the drop down arrow.
	The event that must be carried out if Sprite1 is pressed. Sprite1 can be used as a button. When you can click on the button, a task will be carried out.
	A control block used to send a message. A new message can be created by clicking on the drop down arrow. Allows for one to one / one to many communication.
	A control block used to send a message. A new message can be created by clicking on the drop down arrow. This is similar to the broadcast block above except the sprite that is broadcasting the message waits until the scripts it activated have run completely. It then continues with the rest of the current script. Allows for one to one / one to many communication.
	Allows a sprite to receive a message and respond according to the relevant script.

Example 1

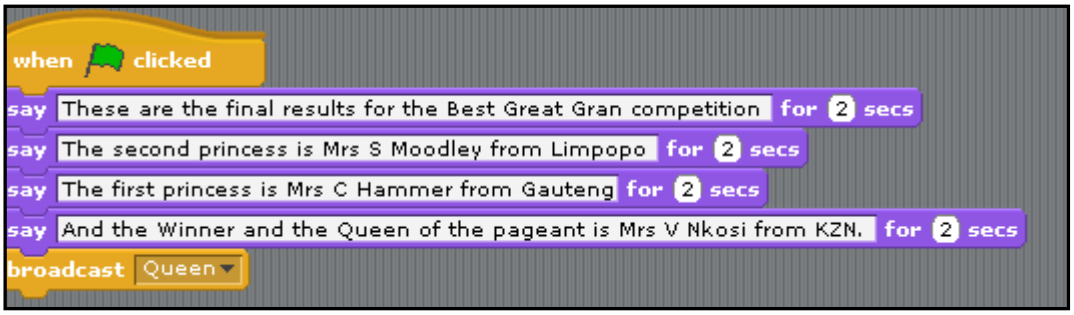
Create a project to release the results for the Best Great Gran competition.

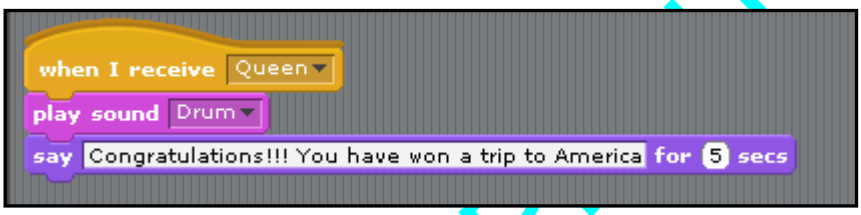
Announce the second princess, the first princess and then queen.

Broadcast the message “Queen” at the end of the announcement.

When the message “Queen” is received, play the drum sound and display

a suitable message indicating she has won a trip to America.





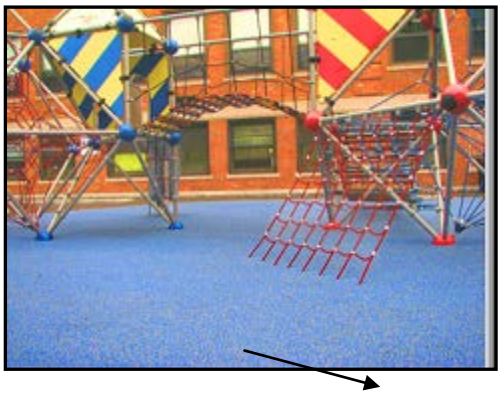
Example 2

Create a project to assist at the drama playhouse at your school. Messages will be sent to you to adjust the background appropriately. The background is either indoor playground or outdoor playground.

Step 1: Select the two backgrounds and the cat as a default sprite.

Backgrounds can be selected by clicking on the stage, and then on the import command.

Indoor playground



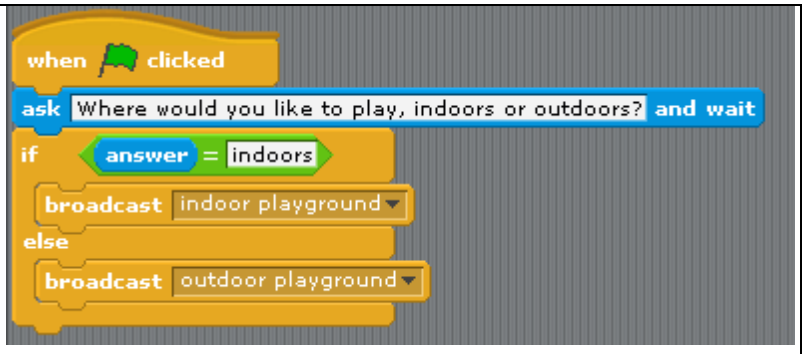
Known as atom-playground

Outdoor playground

Known as playing-field

Step 2:

Write the required script.

Sprite1	
Stage	

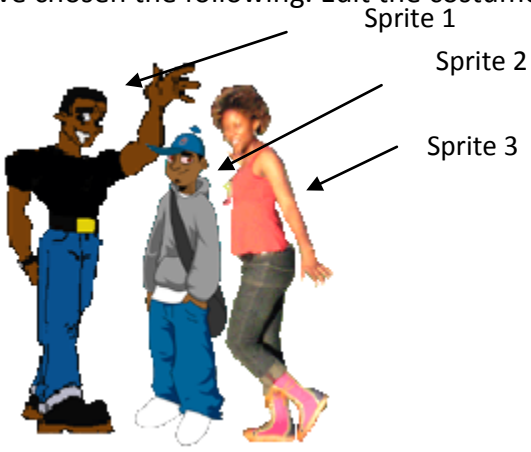
Example 3

Your mum has organised a surprise birthday party for your dad. She asks you to stand outside the house and await dad so that you can tip her off as he enters. You need to say “Hi Dad How was your day?”. As soon as mum hears the message, she will play the Drum machine sound and scream out “Surprise” for 2 seconds and say “Happy Birthday Sweetheart”. You only need to wait for five seconds when dad arrives.

Step 1

Choose three sprites, a sprite to represent mum, a sprite to represent dad and a sprite to represent yourself (the little boy or girl).

I have chosen the following. Edit the costume for direction and size where necessary.



Step 2

Write scripts for each sprite.

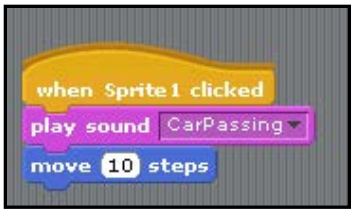
Sprites	Scripts	
Sprite 1		The project will start when the green flag is clicked Hide sprite1 at the start of the project Wait for 5 seconds after the project is activated Show sprite1
Sprite 2		The project will start when the green flag is clicked Show sprite2 at the start of the project Wait for 5 seconds after the project is activated Say “Hi dad. How was your day?”, this block simply prints this statement on the stage Sends the message to all the other sprites. Used to trigger Sprite3.
Sprite 3		The project will start when the green flag is clicked Hide sprite3 at the start of the project How to respond when this message is received Show Sprite3 Play the DrumMachine sound Sprite3 says “Surprise”. This statement is printed on the stage. Sprite3 says “Happy Birthday Sweetheart!!!!”. This statement is printed on the stage.

When Sprite Clicked 

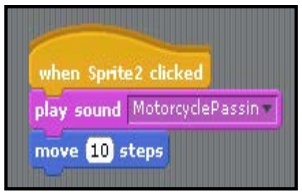
Example

Create a project to select a picture of the following as sprites. Select a car, bicycle, boat and an aeroplane. Select an appropriate sound for each sprite. When a sprite is clicked on, an appropriate script must be written to move the sprite 10 steps forward and play the required sound. Remember to click on each sprite separately, import a sound and then write the script. Choose the sounds before starting the scripts.

Script for Sprite1
Choose the CarPassing sound
from the Effects folder



Script for Sprite2
Choose the MotorcyclePassing sound
from the Effects folder



Script for Sprite3
Choose the Ripples sound from the
Effects folder

Script for Sprite4
Choose the Peculiar sound from the
Electronics folder

When key pressed 

This block allows you to perform an action if the following keys are pressed; the arrow keys, the space key, the letters of the alphabet and numbers from 0 – 9. You can select other keys by clicking on the drop down arrow near the word “space”.

Example

Create a project to paint Sprite1 as a red ball.
Write a script that uses the arrow keys to move the ball 10 steps up, down, left or right depending on the arrow key that is pressed. See Figure 1

Figure 1

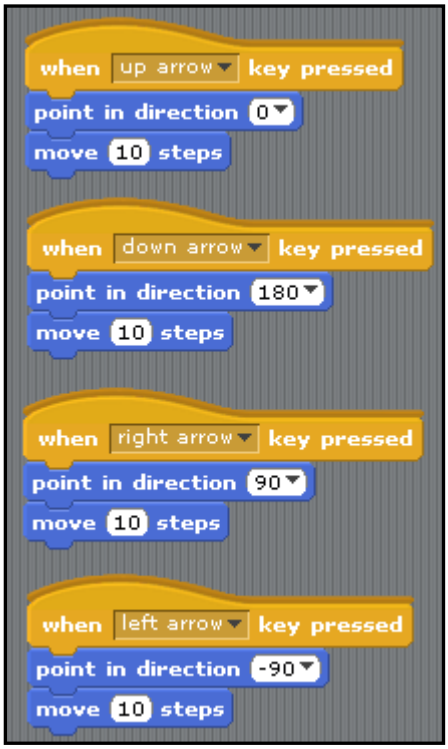
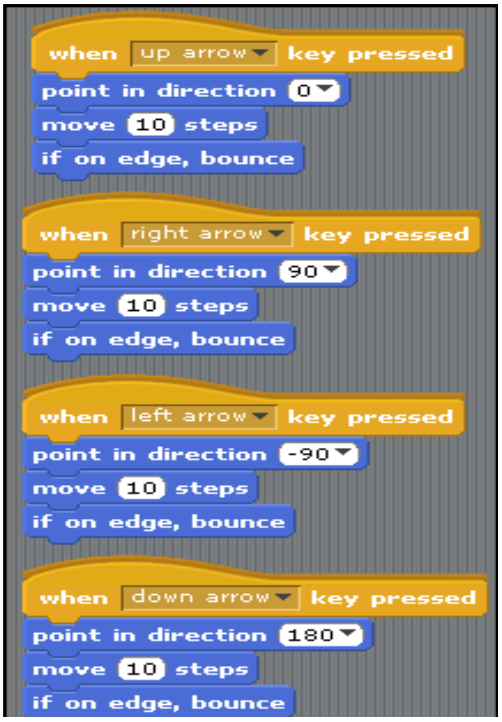


Figure 2



You would notice that if you continue pressing on any of the arrow keys, the ball might move out of the stage area.

How to keep the sprite within the stage area 

This block allows the sprite to remain within the stage area.

Add this block to the end of each "When key pressed" block in the previous example. See Figure 2.

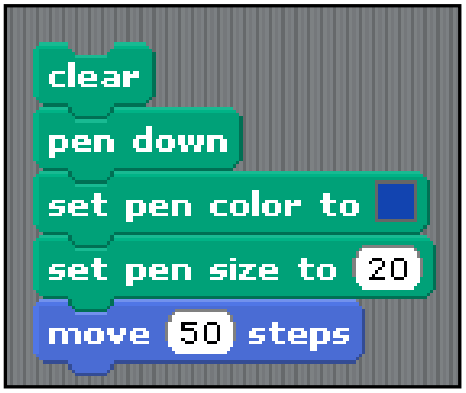
Exercise

1. Create a project using two "people" sprites to conduct a conversation on the topic on drugs. Use a suitable background, sound, colour and different costumes.
2. Create a project that uses two "fantasy" sprites. Use the arrow keys to move one of the sprites up, down, left or right. The project must be terminated when sprite1 touches sprite2. Also display "time out" when the sprites touch.
3. Create a project that uses a sprite to prompt the user to enter their name. Allow the user to choose between two animals the "unicorn" and the "dragon". Switch to the appropriate costume with an appropriate sound depending on the choice made by the user.
4. Create a project that uses many sprites and a suitable background to synchronise people walking in a group and having a conversation about school.
5. Create a project that produces a dance item with three "people" sprites. Use sound, colour and a variety of costumes.

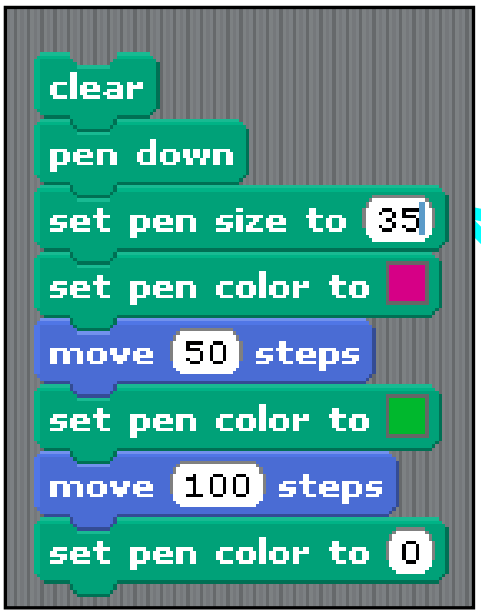
6. Create separate projects each using sound, a variety of costumes, at least two stages and colour for each of the following titles.

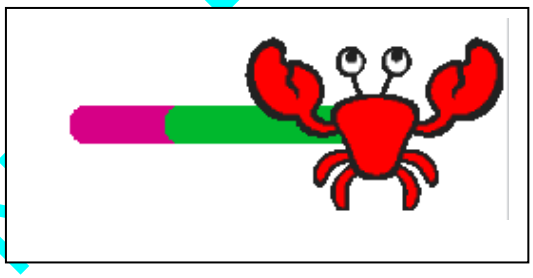
   6.1 A scary night alone
   6.2 A day at the beach
   6.3 My family grocery store

Pen



This script will clear the stage, prints in the selected colour using a specified pen size which indicates the thickness of the line.





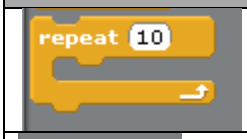
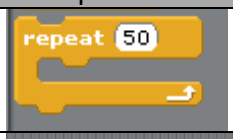
Repetition / looping

Let me take you back to a previous example on Anna doing her exercise routine. Her routine was created once, but she wanted to carry out the routine 10 times. So in order to do that, we created a script with the routine once and we added a 

block from the Control category in

order to execute the routine 10 times. If Anna wanted to execute the routine 20 times, all we need to do is change the number from 10 to 20 in the repeat block.

The following blocks can be used for repetition

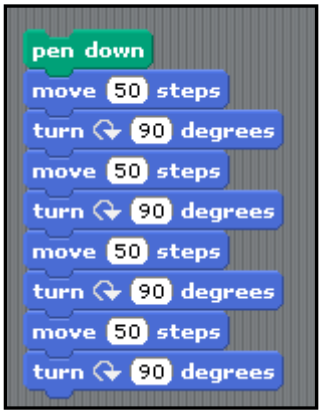
Block	Example	Description
		Executes the blocks inside the “repeat” block 50 times
		Executes the blocks inside the “forever” block continuously. Will move the Sprite continuously in a circular motion, 10 steps forward, changes direction by 15 degrees, clockwise
		Executes the blocks inside the “forever if” block continuously if the condition stated is true. Only if the sprite is not touching the edge will the blocks get executed.
		Executes the blocks inside the “repeat until” block until the space key is pressed. Allows the user to control the stopping of an action / event.

An example to illustrate how we can use looping to avoid the use of unnecessary blocks

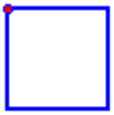
Let’s say we want to draw a square with side 50 steps.

Create a “dot” sprite, that looks like • .

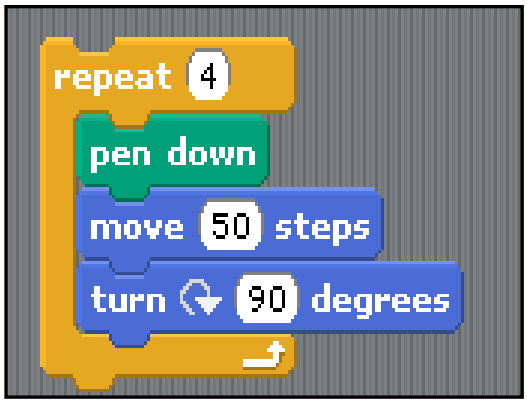
The script you can use without looping:



Sample Stage

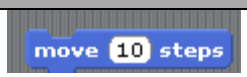
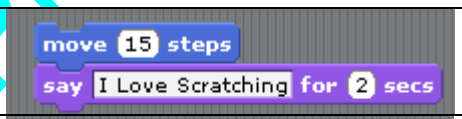
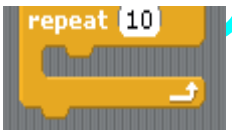


Shorter script using looping to avoid repetition



Example 1

Create a project that makes the cat sprite move 15 steps forward, says "I Love Scratching" for 2 seconds and waits for 1 second. This event must take place 10 times.

Steps	Select Blocks	Project Progress
Step 1	Select  from Motion	
Step 2	Select  from Looks	
Step 3	Select  from Control	
Step 4	Repeat the event 10 times. Enclose the project inside this block. Select  from Control	

Example 2

Create a project to make the cat sprite turn 180 degrees in a clockwise, circular motion, 20 times. After each time, the cat must wait for 1 second. Change the colour effect of the cat by 25.

Example 3

Create a project to select sprite dan-1 from the People Folder.

Add costumes dan-2, dan-3, dan-4, dan-5 and dan-6.

Create a dance routine for Dan by switching costumes from dan-1 to dan-6.

Repeat this dance routine continuously, meaning non – stop.

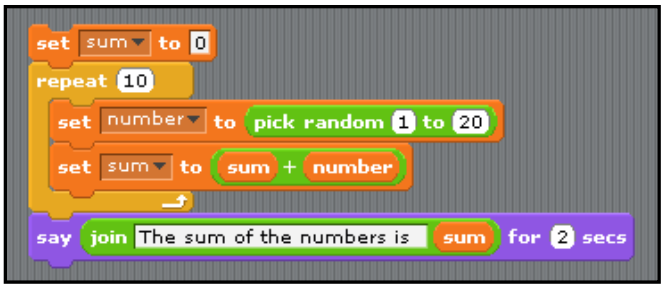
If you want to create a block to terminate the dance routine, you can use a button like the space bar to stop.

You can also add sound to the routine to make it more interesting.

Where to place blocks when using loops? If you need the block to be executed once, the block must be placed out of the loop. If you need the block to be executed repeatedly, then the block must be placed inside the loop.

Example 4

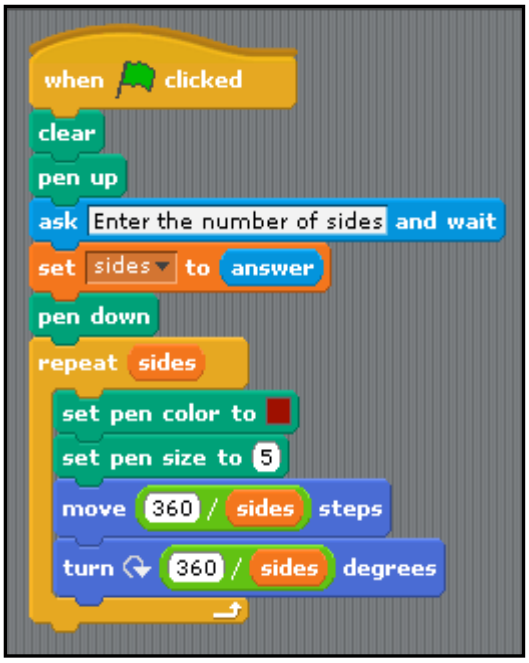
Create a project to randomly generate 10 numbers in the range 1 – 20. Calculate and display the sum of the numbers. Create two variables called number and sum.



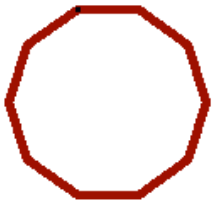
The variable sum must be set to 0 at the beginning to initialise the memory box. Every time the program is executed it will start sum from 0 otherwise it will add the new sum to the previous sum.

Example 5

Create a project to enter the number of sides of a figure. Draw a figure with the number of sides entered.



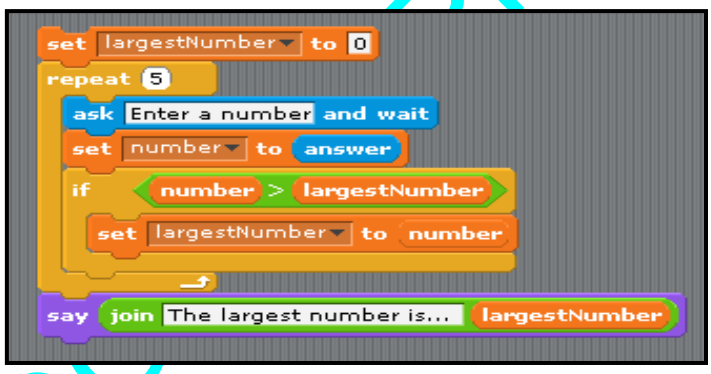
The output for sides = 10



The output for sides = 4

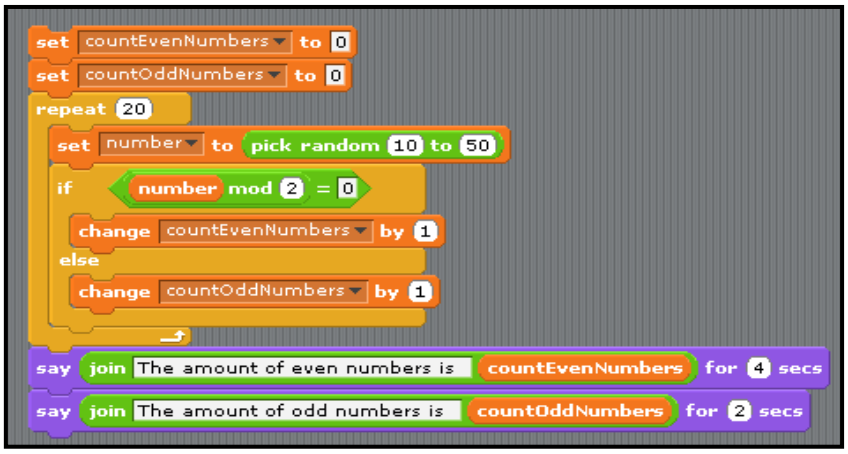
Example 5

Create a project to enter 5 positive numbers. Determine and output the largest number.



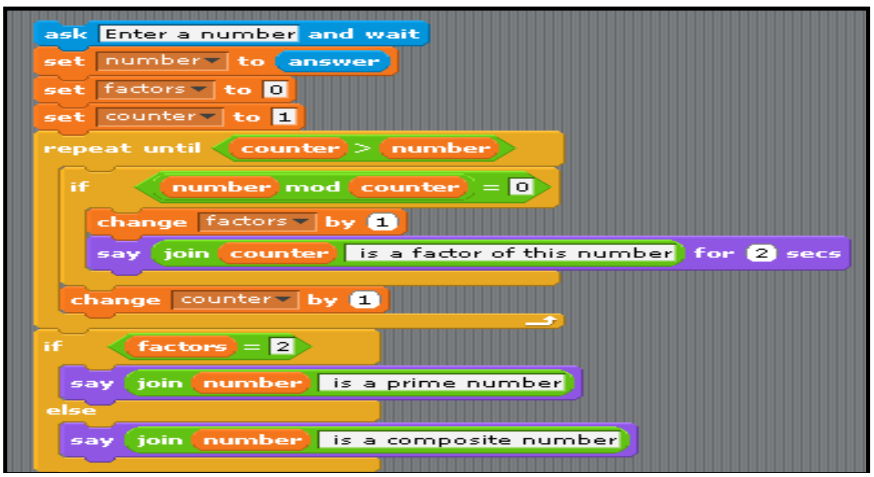
Example 6

Create a project to randomly generate 20 numbers in the range 10 – 50. Determine and output the amount of even numbers and the amount of odd numbers.



Example 7

Create a project to enter a number. Display a suitable message indicating if the number is prime or composite. A prime number has 2 factors, 1 and itself.

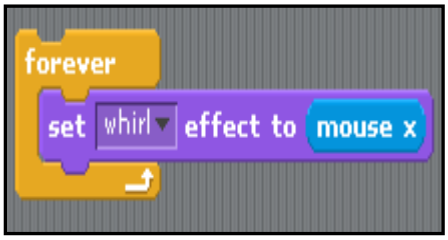


Exercise

1. Create a project that uses the sea as a background. Select 4 types of fish that must be allowed to move randomly all over the stage. The fish must move continuously, that is, use the "forever" block.
2. Create a dance routine for a sprite. The dance routine must be terminated when the space is pressed. Add sound and a variety of costumes.
3. Create a project to enter 10 numbers. Find and display the average number.
4. Create a project to randomly generate 6 numbers in the range 1 – 100. Display the amount of numbers that are divisible by 2 and 3.
5. Create a project to randomly generate 4 three digit numbers. Display the largest and the smallest number.
6. Create a project to make a square with side = 100.
7. Edit the script above to make 5 squares with the smallest square having a side of 100. Increase the size of each square thereafter by 50. Use a different pen colour for each square.
8. Create a project to enter a number. Display the factorial of the number.
   Example if 4 is entered; the factorial is 1 x 2 x 3 x 4 = 24. Display only the answer 24.
9. Create a project to display the first 10 numbers in the Fibonacci sequence.
   1 1 2 3 5 8 13 21 34 55
   Edit your solution to enter any number. Display that number of terms in the sequence.
10. A perfect number is a number that is the sum of its factors other than itself.
    Create a project to enter a number. Display a message indicating if the number is a perfect number. Example 6 is a perfect number since its factors are 1 + 2 + 3 = 6.
    Do not add the number itself.
11. Create a project to enter 3 names and six marks for each name. Display the name of the person with the highest total mark. (Note you would require 2 repeat blocks; the one to control the amount of names and the other to control the amount of marks).

More on Looks and Sensing

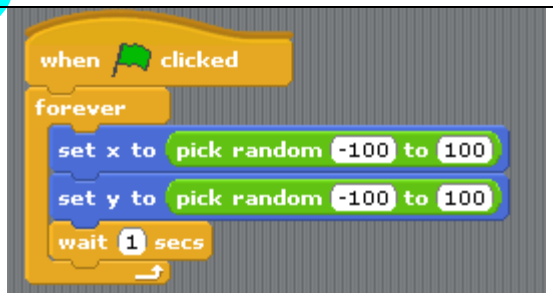
1. Distorting images



Will create a distorted sprite body. The shape changes according to the movement of the mouse.

2. Catching Game

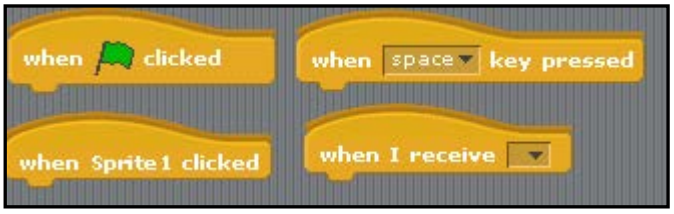
Select two sprites a cat and a dog. Write a script each for the cat and the dog where they are allowed to move about randomly within an area of x = -100 to 100 and y = -100 and 100. Stop the game when the dog touches the cat. The dog must say "Caught" when it touches the cat.



Threads

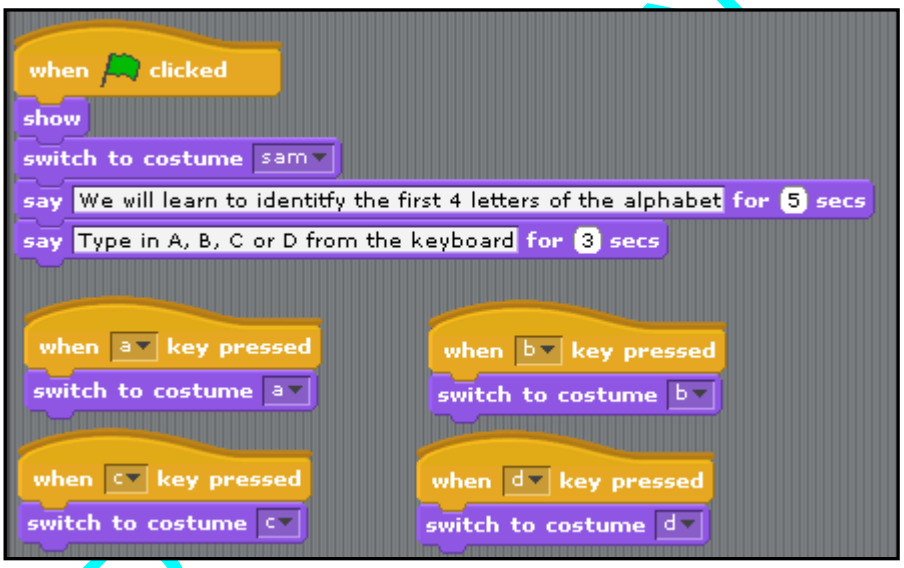
A thread is the simultaneous execution of more than one stack of blocks. This simply means that a thread can also be seen as a small program inside a bigger program where all threads can be

executed at the same time. Any block that starts with "when" indicates the start of a new thread. These are some examples of how to start a thread.



Example 1

Create a project to select sprite "Sam"  and 4 costumes which are the letters A, B, C and D. You project must allow Sam to say "We will learn to identify the first 4 letters of the alphabet", "Type in A, B, C or D from the keyboard". Display the corresponding letters on the stage.



Exercise

1. Create a project to choose an animal sprite. Import the numbers from 0 – 9 as costumes. Show the selected number on the stage as pressed on the keyboard. Also add a different sound for each number selected.

2. Create a project to select five animal sprites. Reduce the size of all sprites so they fit neatly horizontally on the stage. Import a sound for each animal sprite. When the animal is selected the appropriate sound of the animal must be played.

3. Create a project that shows a people sprite and select four types of sport balls as sprites. Work out the co-ordinates of the foot of the people sprite. When any one of the balls is selected, the selected ball must glide to the foot of the people sprite. You may label each ball from A to D.

Lists

A list is a data structure that can store more than one value. A traditional variable like “number” can store only a single value while a list can store multiple values. Each value in the list has a position. The position is called an index. The value at the first position is 1, the value at the second position is 2 etc. A list can also be referred to as an array.

Example

Create a project that makes a list called teamList. Put the following names in the list.

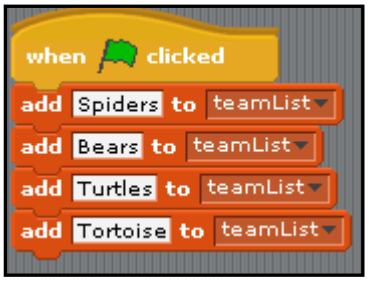
Spiders, Bears, Turtles and Tortoise.

Solution:

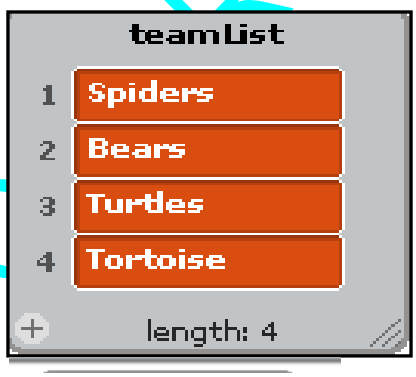
Click on Variables, Click on Make a list and name the list “teamList”.

Add the following to teamList. Use the appropriate block to form the following script.

Script



Stage



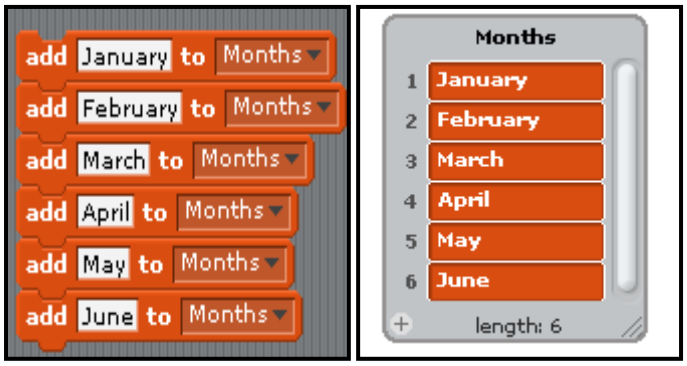
The list is dynamic, that is it has no definite size but will change as the tasks performed changes. The length of the list in Scratch allows for dynamic change by the user, unlike other programming languages where the size of the array must be pre-defined.

Block	Description
delete 1 of teamList	Delete the entry at position 1 of the list
add thing to allNames	Add an entry to the end of a list
insert thing at 1 of teamList	Add an entry at a specified position of the list
replace item 1 of teamList	Replace an entry at a specified position of the list
item 1 of teamList	Extract item 1 in a list
length of teamList	Used to determine the length of the list
teamList contains thing	Used to check if the list contains a specified value

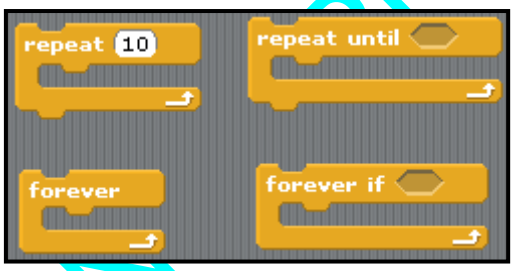
Example 2

Create a project to create a list called months. Add the first six months to the list.

Script **Stage**

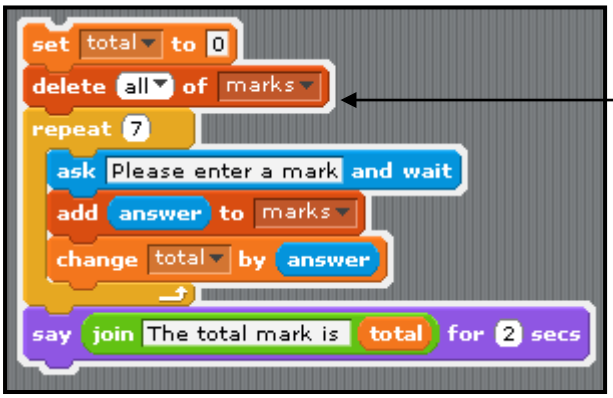


We traditionally use a list when we have many entries that need to be stored, example if we need to store 100 marks or 500 scores etc. Our project will be lengthy if we use an add block for each number that needs to be entered. We therefore use the looping blocks for input and output of data in a list.



Example 3

Create a project that asks the user to enter seven marks. Display the total of the marks. Make a variable called total and make a list called marks.

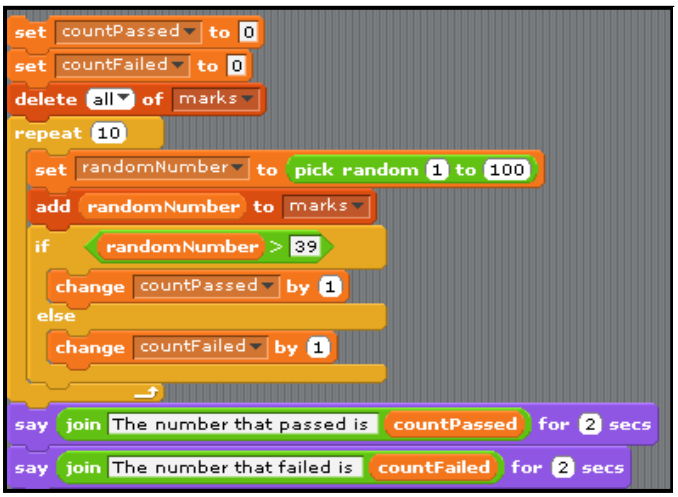


Ensure that you include this block so that the project will always start entry at index 1, otherwise it will add to the end of the list every time the script is clicked on.

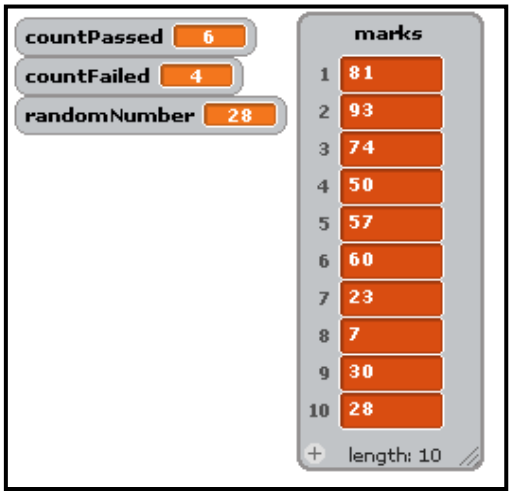
Example 4

Create a project to allow the user to randomly generate 10 marks in the range 1 – 100. Determine and display the amount of marks >= 40 and the amount of marks < 40. Make 3 variables called randomNumber, countPassed and countFailed. Make a list called marks.

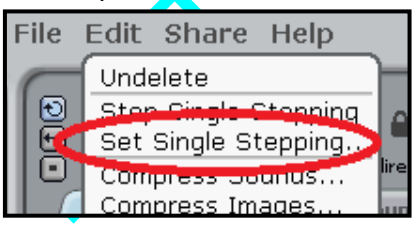
Script



Stage



NB. You would notice that when you execute this script, it runs very fast to give the output. If you want to slow down the process of the output, click on Edit, select Set Single Stepping.



The output will change for every run as the numbers are being randomly generated.

Exercise

1. Create a project to make a list called numbers. Allow the user to enter 5 numbers. Find and display the average number.
2. Create a project to make a list called temperature. Randomly generate the temperature for 10 days. Find and display the highest temperature.
3. Create a project to make a list called days. Store the seven days of the week. Swap Monday with Thursday.
4. Create a project to make a list called careers. Store at least fifteen careers in the list. Pick a random index from 1 to 15. Display the career at this index from the careers list.
5. Create a project to make a list called names and add five names. Add menu options to add, delete or replace a name. Create a script to satisfy each menu option.
6. Create a project to pick 10 random numbers in the range 1 – 100. Display the smallest number and the index or position of the smallest number.
7. Create a project to store 8 marks as follows in markList: 56; 87; 99; 24; 55; 45; 49; 80. Sort the marks so that they are arranged from lowest to highest.

Parallel Lists

Data in two or more arrays correspond using the index or position of a value.

Example: Create a project to store the following names and marks of learners.

Index / Position	Names	Marks
1	Adrian	89
2	Brian	76
3	Amy	90

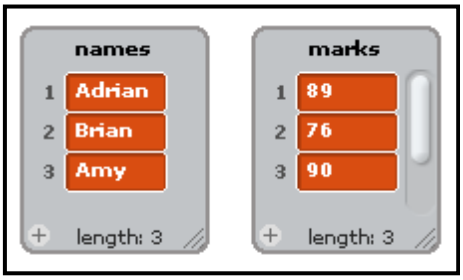
A parallel list means that the mark at position / index 1 corresponds to the name at index 1. The mark at index 2 corresponds to the name at index 2 etc. The order of entry is important and must match.

Solution

Make two lists, one for names and the other for marks.

Script

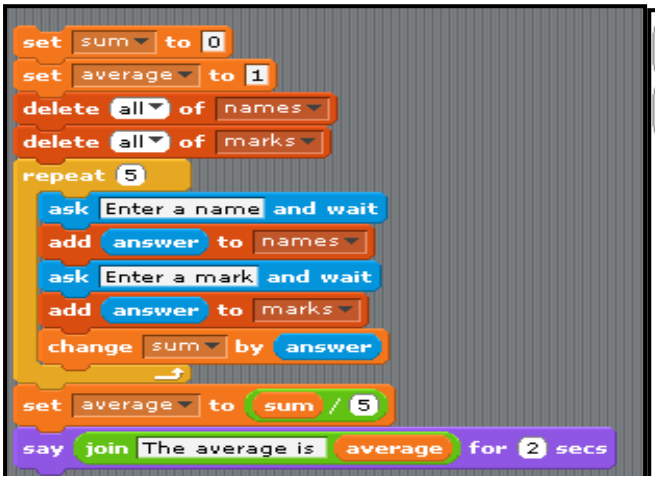
Stage



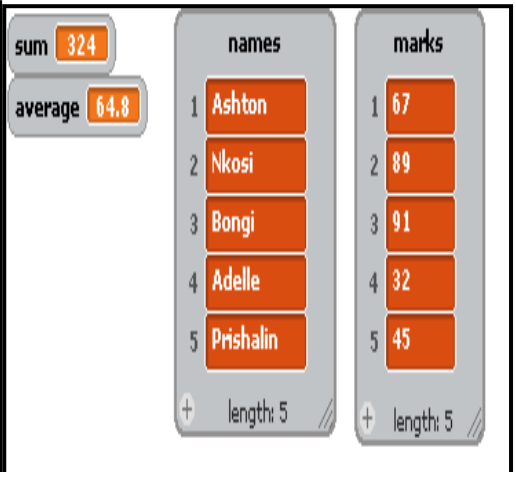
Example 1

Create a list to allow the user to enter 5 names and marks for the respective names. Display the average mark.

Script



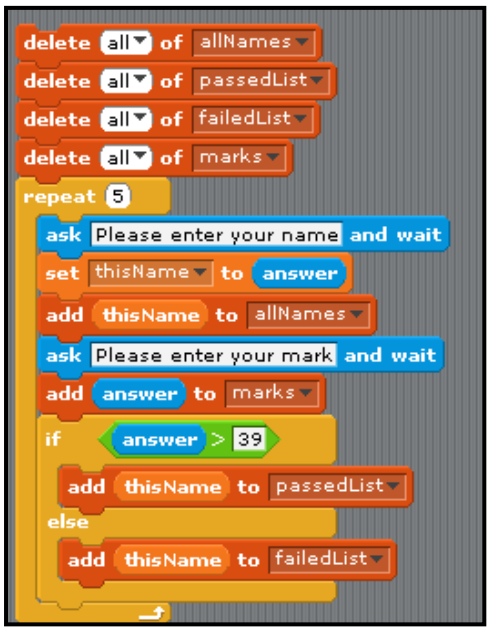
Stage



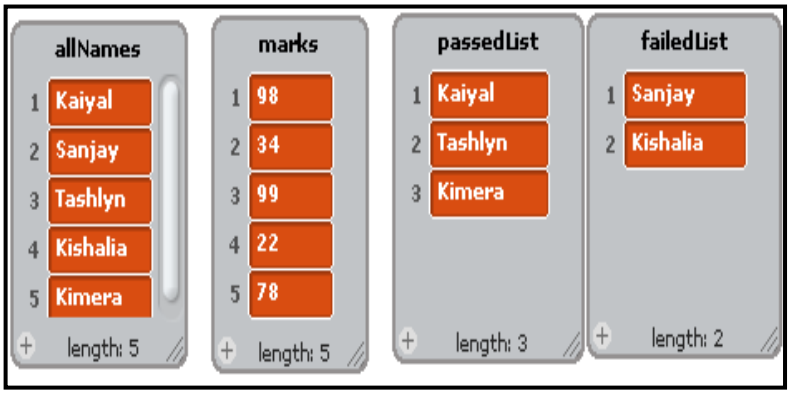
Example 2

Create a project that allows the user to enter 5 names and a mark for each name. Extract and store the names of all learners that have passed and all learners that have failed in two separate lists. Create a variable called thisName and 4 lists, allNames, marks, passedList and failedList.

Script



Stage



Exercise

1. Create two lists, one called itemsList and the other called pricesList. Add the following items and the corresponding prices. Shirt=R99.99, Pants = R160, Dress = R125, Tie = R30. Display the total of the shirt, pants and tie.

2. Create two lists, one called monthList and the other called rainfallList. Store the twelve months in the monthList. Randomly generate rainfall readings in the range 50 – 150. Find and display the average rainfall reading. Also count and display the number of months that had a rainfall reading less than the average rainfall reading.

3. Create two lists, one called animalsList and the other called lifeSpanList.
   Store these values. Sheep: 15 Wolf: 18 Tiger: 22 Lion: 35
   Allow the user the option of adding an animal and the lifespan, deleting or changing the lifespan of an animal.

4. Create three lists, one called computerAccessoriesList, one called pricesList and one called expensiveAccessoryList. Allow the user to enter five accessories and its prices. The accessories that are more than R1000 must be placed in the third list.

   Sample data: speakers: R349 hard drive: R1500 printer: R2000
   Webcam: R499 flash drive: R60

5. Create two lists called nameList and totalMarkList.
   Enter 5 names and store these names in the nameList. Pick 7 random marks in the range 1 – 100 for each name. Calculate the total of the 7 marks and store in the totalMarkList.

6. Create four lists namely itemList, priceList, vatList, totalList.
   Enter 6 items into itemList and its corresponding prices in pricelist.
   Calculate and store 14% vat of each item in vatList. Add the corresponding values from pricelist and vatList and store the result in totalList.

   Sample data: Milk 18.99 Bread 13.98 Fruit juice 26.79
   Yoghurt 23.49 Salad 37.90 Almonds 56.50

7. Create two lists called teamList and scoreList. Enter the names of 5 teams and a score for each team.
   7.1 Re-arrange the team names and scores in descending order of the scores.
   7.2 Allow the user to enter a team name to search for. If the team name is found; update the score for the team by asking the user to enter the correct score. Display a suitable message if the team name is not in the team list.

   Sample data: Pilots 78; Jelsea 61; Citi 90;
   Snill 82; Pool 79

8. Create a list called numberList. Add the following numbers to the list.
   67; 56; 34; 67; 55; 23; 55; 51; 67;65
   Make another list called removeDuplicateList to extract and show the values from numberList without duplicates.

Strings

A string is alphanumeric, that is, it can contain letters of the alphabet, numbers and special characters.

The following blocks are commonly used to manipulate strings

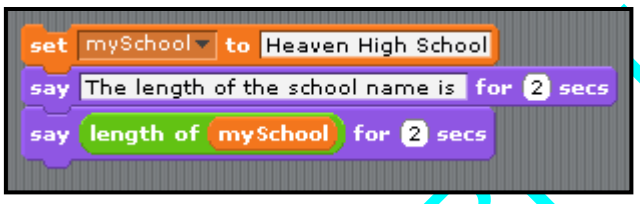
	Used to join, combine or concatenate two strings.
	Indicates the letter or character at a specified position in a string
	Indicates the number of characters in a string
	Used to check if the list contains a specified string

Example 1

Create a variable called mySchool and add the relevant blocks as given in the illustration. The project will display the first, third and sixth letters of the school name.

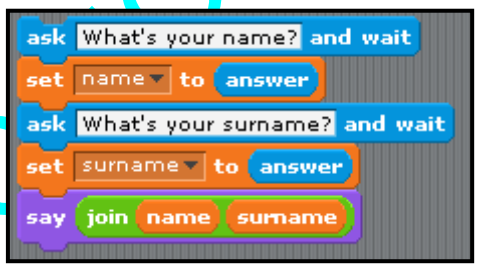
Task 1: Add blocks to the project to display each letter of the word "High" in the school name.

Example 2: Using the same school name, display the length of the name of the school.

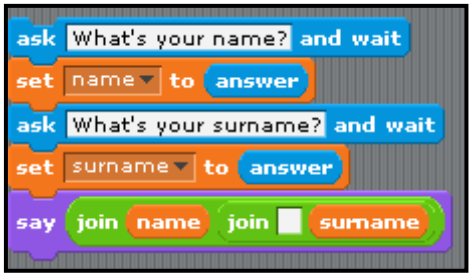


Task 2: Create a variable called myBestFood. Set this variable to "Chicken pasta". Display the length of myBestFood.

Example 3: Create two variables called name and surname. Allow the user to enter the name and surname from the keyboard. Join the name and surname to display the full name.

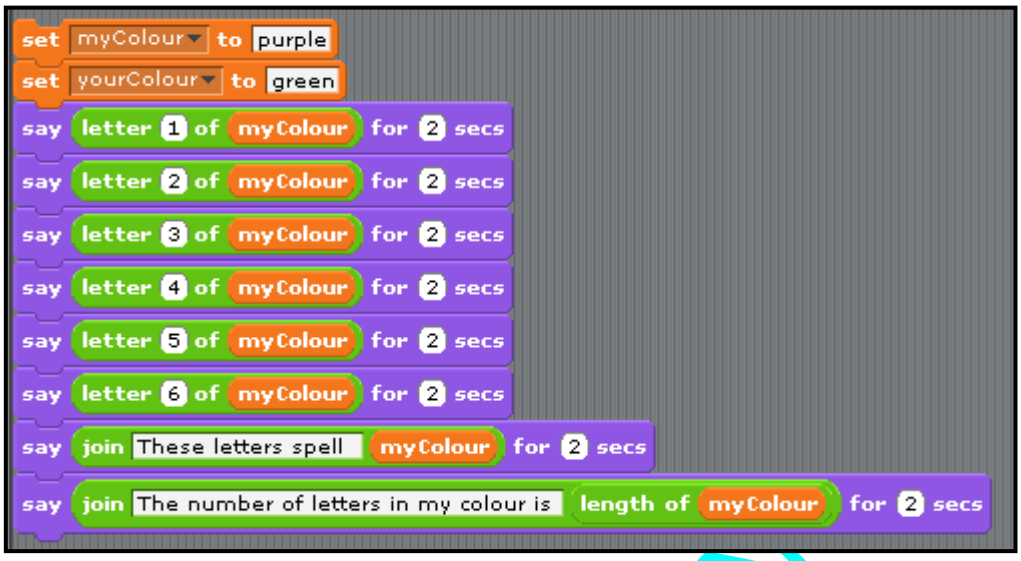


If the name entered is Keelan and the surname entered is Jenkins, the display will be KeelanJenkins. How do we create a space so that the display will be Keelan Jenkins? You need to add another join block as follows:



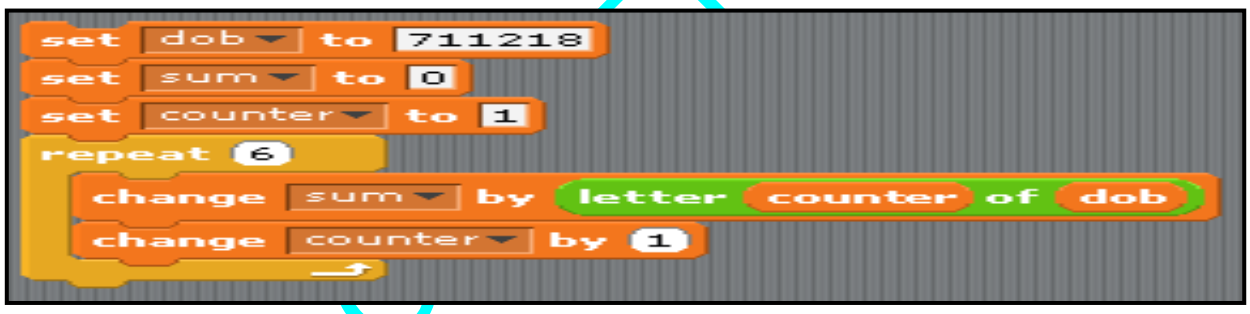
Task 3: Create a project to allow the user to enter their street number and then the street name. Display the street address with a space between the number and name.

Example 4: Create a project that uses two variables mycolour and yourColour. Set myColour to purple and yourColour to green. Display each letter of myColour and then the length of myColour.



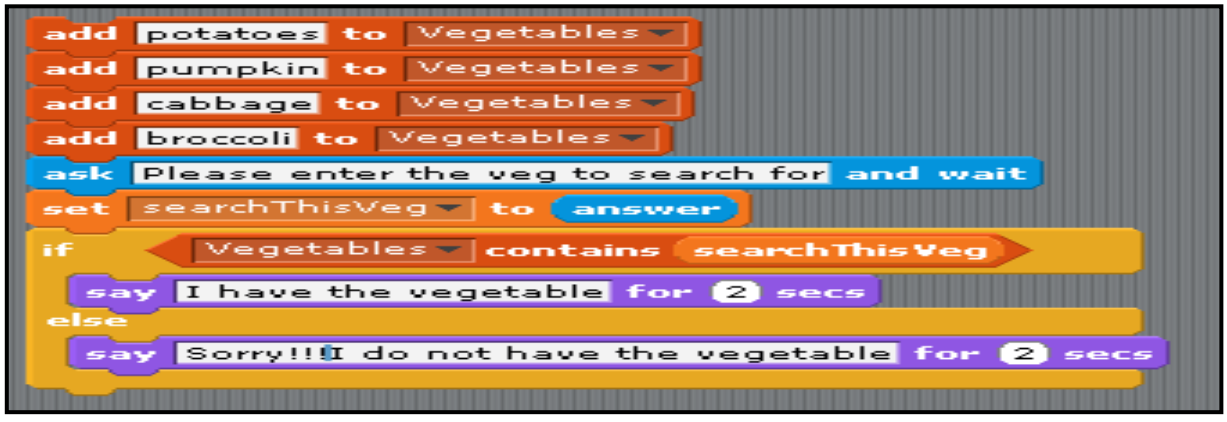
Task 4: Add blocks to the code above to display the same information for yourColour.

Example 5: Assign the following date of birth to a variable called dob. Find and display the sum of the digits of the date of birth.



Example 6:

Create a list called vegetables and a variable called searchThisVeg. Add the following to the list, namely, potatoes, pumpkin, cabbage and broccoli. Ask the user to enter the vegetable to search for and display a suitable message indicating if the vegetable is found or not.



Task 6:

Create a list called fruit and a variable called addFruit. Add the following to the list namely, apples, bananas, oranges, strawberries and kiwi. Ask the user to enter the name of a fruit to add to the list. Display a suitable message if the list already contains this fruit otherwise add the fruit to the list.

Exercise

1. Create a project to enter your address. Display the fourth to the eighth characters one at a time.

2. Create a project to enter a first name and a surname. Display the surname and the initial.

3. Create a project to simulate a code. Enter your name, surname and date of birth in the format yyyy/mm/dd. The code must be made up of the following. The first letter of the name, the last letter of the surname, the month of birth and the symbol "#".

4. Create a project with a list called groceryList. Add at least five items. If the list has the item called fruit, display a message "Approved" otherwise delete all the items from the list and re-enter the 5 items..

5. Create a project that allows the user to enter an id number.
   Display the sum of the digits of the id number.
   Calculate the age by entering the current year.

6. Create a project that allows the user to enter a sentence.
   Display the number of characters in the sentence. (The length)
   Display the number of words in the sentence.

7. Create a project that allows the user to enter a sentence.
   Display the number of vowels in the sentence.
   Display the sentence in reverse order.

8. Create a project that allows the user to enter a sentence.
   Display the sentence without the vowels.

9 Create a project that allows the user to enter a word.
   Display a message indicating if the word is a palindrome or not.
   A palindrome is a phrase that spells the same forwards or backwards; e.g. madam, level

10. Create a project that allows the user to enter a sentence.
    Replace all spaces with a star(*). Replace the letter Z with a hash(#).
    Display the sentence with the replaced characters.

11. The format of a South African ID is:

{YYMMDD} {G} {SSS} {C} {X} {Z}

- {YYMMDD} is the date of birth
- {G} is gender. 0-4 is female and 5-9 is male
- {SSS} is a sequence number based on the sequence of registration.
- {C} is citizenship. 0 for South Africans and 1 for other citizens
- {X} is usually 8 or 9 – a part of the check digit
- {Z} is a check digit calculated from the other digits

So for the ID number 9109075012083 we know certain things about the person:

- Born on September 7th , 1991
- Male
- 12th male born on that date to get an ID number
- South African citizen

Create a project to do the following:

Enter an ID number.

Display the date of birth in the format 01 January 2014.

Display the gender

Display the sequence of registration

Display if a South African citizen or not.

Verify that the 12th digit is either an 8 or a 9.