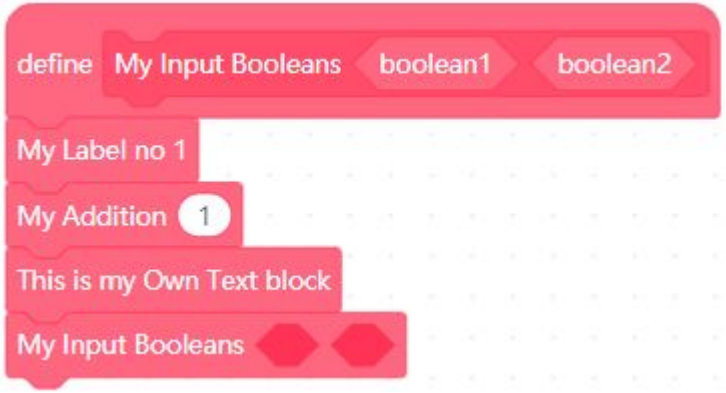
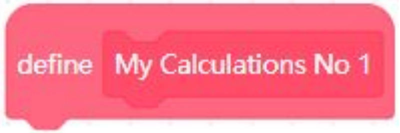
Lesson 10 – Making Our Own Blocks (functions)



```
define My Input Booleans boolean1 boolean2
My Label no 1
My Addition 1
This is my Own Text block
My Input Booleans
```

My blocks allow us to create new blocks for a sprite.
We give the block a name and then define what the new block does using other Scratch blocks.

Ex the block is defined & named Calculation n



define My Calculations No 1

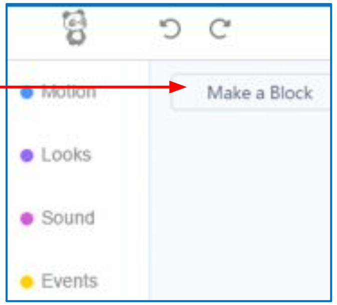
Under it are the other blocks I need for doing the calculations.

My Blocks

These blocks are created by us.
They statements for the selected sprite.
They are color-coded **pink or Purple.**

Before any statements is created
it is empty, except for a **"Make a Block"** box.

On selecting, procedure for making block statements starts.



Uses of My Blocks. There are **two** distinct areas for using My Blocks.

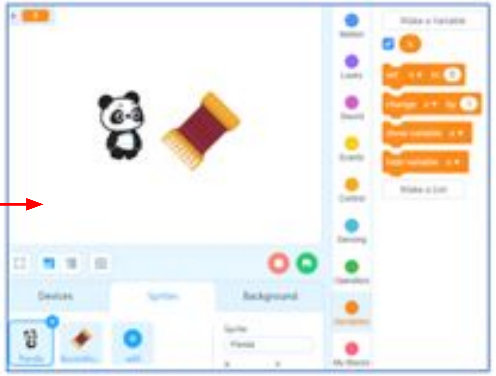
- Use 1. To **convert** multiple lines of certain types of **codes in to one line.**

- Use 2. To **make a function** and then call it in one line using the **name** allocated to that function.

Let us **understand** them one by one. Before that kindly examine note below.

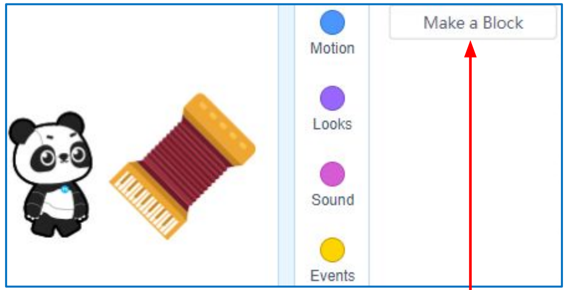
Important Note

Variables blocks have no block statements.
They have to be made by us.
However, in the case of variables,
statements made are **usable by all the sprites.**



In case of My Blocks, the statements are made
& are **usable by a specific sprite** in a project.

If our project has multiple sprites,
we have to make **separate** statements for each.

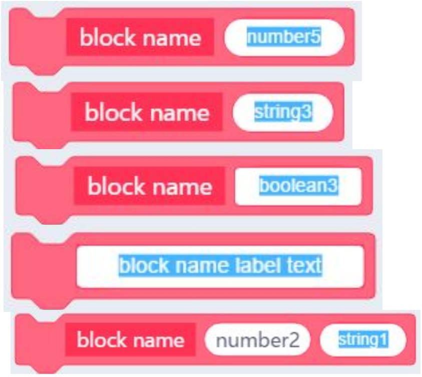


Thus when sprite is selected, it says **Make a block.**

Types of Block Statements

We can create following block statements:

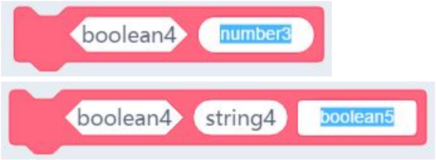
- A number statement.
- A Text or String statement.
- A Boolean statement
- A Label statement
- Mixed statement containing two or more of above.



Types of Boolean Statements

They need to be made:

- As per coding requirements.
- Separate for each sprite.



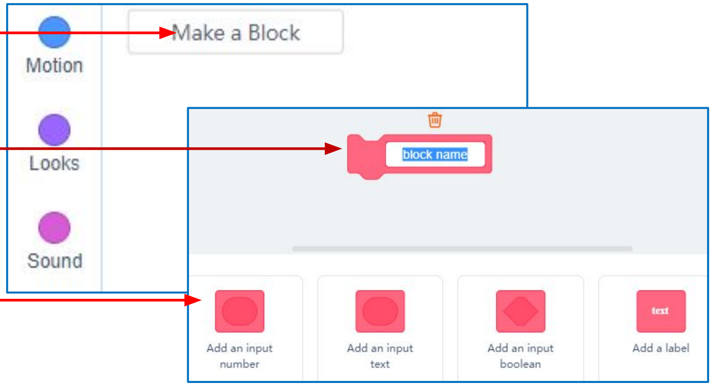
Creating your own blocks

To make a block select my blocks.
In the window that opens select Make a Block.

This opens a dialogue box with a **Skeleton block.**

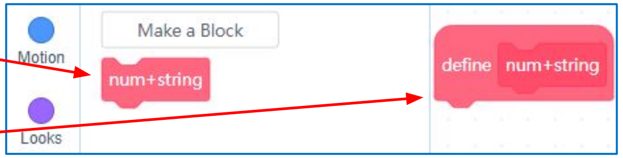
It gives four options to **Add** numbers, text
Boolean & label inside skeleton block.

Let us add a function **num+string.**



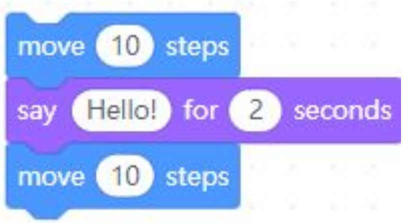
On pressing OK
the **num+string block**
appears in the area called Palette

& a **Definition** block appears in script area.
This **needs to be later defined.**



Making Block Statements

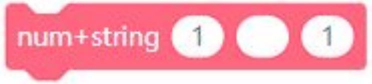
Say we have a **three line** code:



```
move 10 steps
say Hello! for 2 seconds
move 10 steps
```

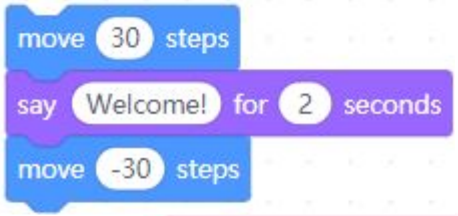
- Two blocks of **numbers** (num).
- One block of **Text** (string).

Using My Blocks we can convert them into **one block** containing all three:



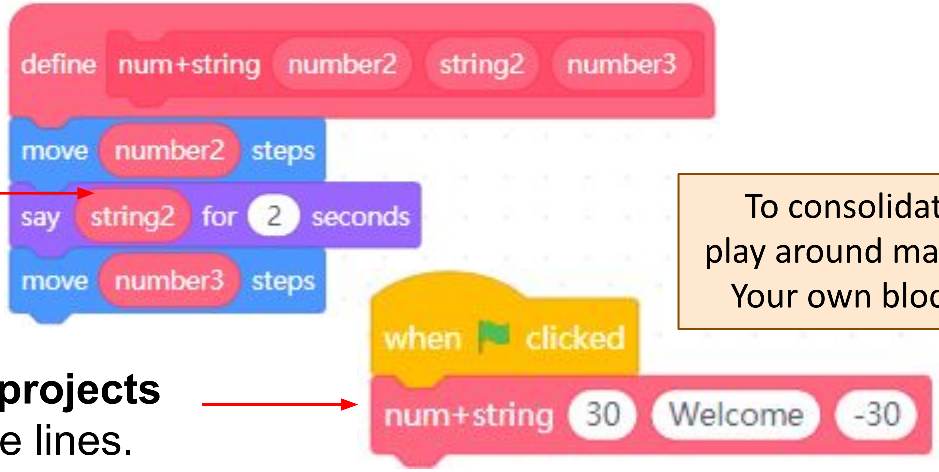
```
num+string 1  1
```

To do so, change parameters.



```
move (30) steps
say (Welcome!) for (2) seconds
move (-30) steps
```

Drag & drop num+string below define.
Add the three reporters.



```
define num+string (number2) (string2) (number3)
move (number2) steps
say (string2) for (2) seconds
move (number3) steps
```

```
when 🏴 clicked
num+string (30) (Welcome) (-30)
```

The block thus created can now be used in projects like any other block. It shall execute all the three lines.

To consolidate
play around making
Your own blocks

A word about functions

In programming:

- Function is a block of **organized, reusable code** that is used to perform a single, related action.
- Functions provide better **modularity** for your application and a high degree of **code reuse.**

In Scratch.

- A function block is like any other block, except that you are the one who **decides what it does.**
- You have to **input values** in the block.
- **My block makes the function blocks in scratch**.

My Block Coding Project

Project – School Annual Day

The story line is simple:

Your school is having its annual day. Lots of parents will be coming.
You are part of the Reception team.
You will be standing at the school entrance.
When any parent comes close to you. One of you will take a step forward.
You will then welcome then and give them the directions to the auditorium.

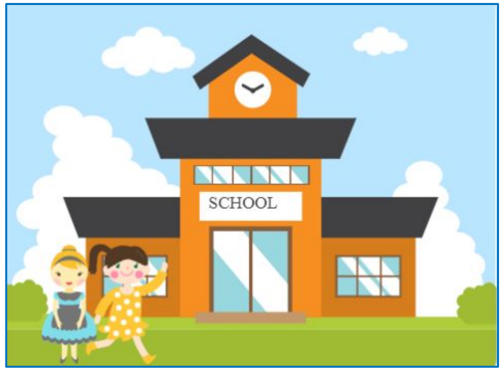
As part of the annual day interaction, you are required to code this story and present the final Code to the audience of parents.

Preparation for Coding - 1

Review the project story line in your mind.

Select the backdrop & sprites from mBlock 5 library.
For this project we require:

- One **backdrop.**
- Two sprites as the **reception team.**
- **Two parents** to be welcomed and guided in.
  Code for remaining parents will be similar.



Preparation for Coding - 2

- Identify the **route** the parents will take to enter the school.
- Identify the **best position** on the stage for locating the reception team.
- Finalise the **number of steps** one of them should take to welcome them.
- Finalise the **welcome messages**.
- Finalise how parents will **enter the school** after the welcome.

Coding Procedure

B**reak the story** into a number of action steps.
Number of steps depend on **your visualisation** of how the story would run?

In this case:

- The coding team has to be in its **position prior** to the arrival of the guests.
- In deciding this position you have to decide from **which direction** the guests will come and in which direction they will go.

- Thereafter, you have to decide the **start point** where the guest appear on the stage. Their **stop** in front of the reception team, and their entry in the school.
- Finally the **trigger for each set of code.**

Suggested Coding Steps

Following steps are recommended.

- Code **actions** of Reception team.
- Code actions of guest 1.

- Create **functions** for Reception team.
- Create functions for Guest 1.

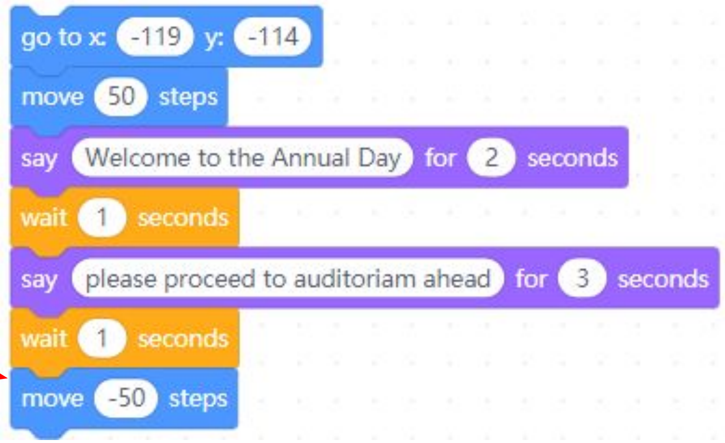
- Trigger the **arrival** of guest 1 at the reception team & later their entry in the school.
- Trigger the **responses** of the reception team to the arrival of the guests.

- Repeat above actions for **other guests with separate triggers.**

Let us Start Coding

Step 1 – Coding actions of Reception team

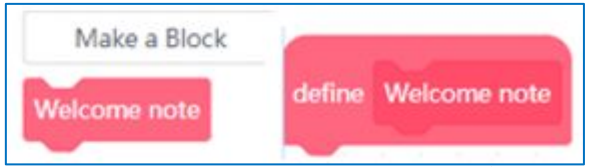
- Decide the **x & y co-ordinates** of the team location.
- The child welcoming takes a **step forward.**
- Delivers message 1.
- After a little wait, delivers message 2.
- Note this wait acts as control for next line.
- After the guests leave, takes a **step back.**



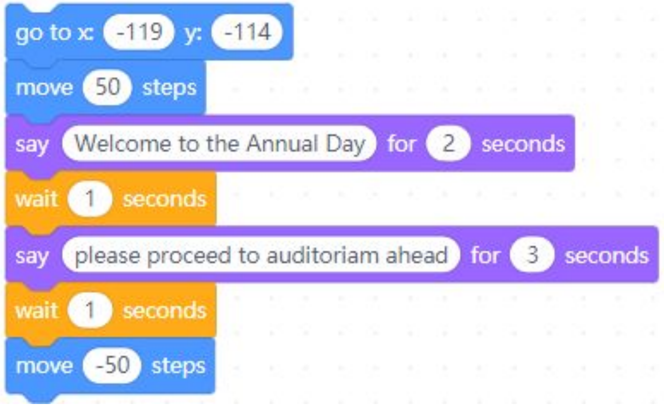
```
go to x: -119 y: -114
move 50 steps
say Welcome to the Annual Day for 2 seconds
wait 1 seconds
say please proceed to auditoriam ahead for 3 seconds
wait 1 seconds
move -50 steps
```

Step 2 – Creating a Function for Reception Team

- Select My Bloc. We learnt its procedure above.
- Remember we need to give name to the function.
- This enables its re-call.
- **Functions name** is **welcome note.**



- The **function definition (actions)** are:
- **Actions to be performed by the function** when called, are stored below **function's Event block.**

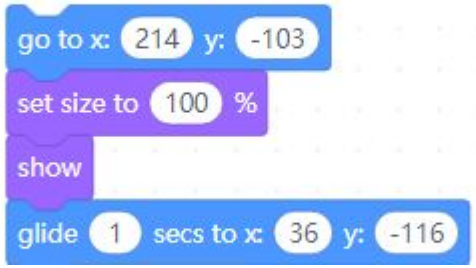


```
go to x: -119 y: -114
move 50 steps
say Welcome to the Annual Day for 2 seconds
wait 1 seconds
say please proceed to auditoriam ahead for 3 seconds
wait 1 seconds
move -50 steps
```

Step 3 – Coding actions of Guest Arrival & Entry in the School

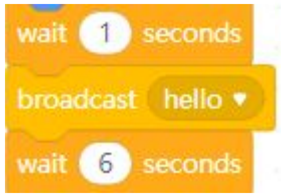
This code has three parts:

- First part codes **guests arrival** from edge of stage at the location of reception team.



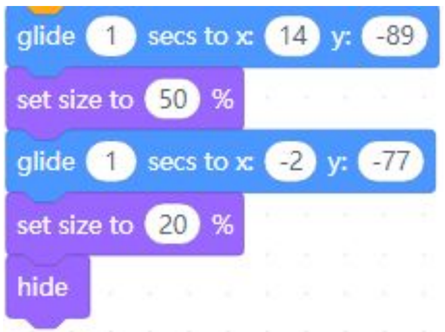
```
go to x: 214 y: -103
set size to 100 %
show
glide 1 secs to x: 36 y: -116
```

- Second **triggers response** of reception team.



```
wait 1 seconds
broadcast hello ▾
wait 6 seconds
```

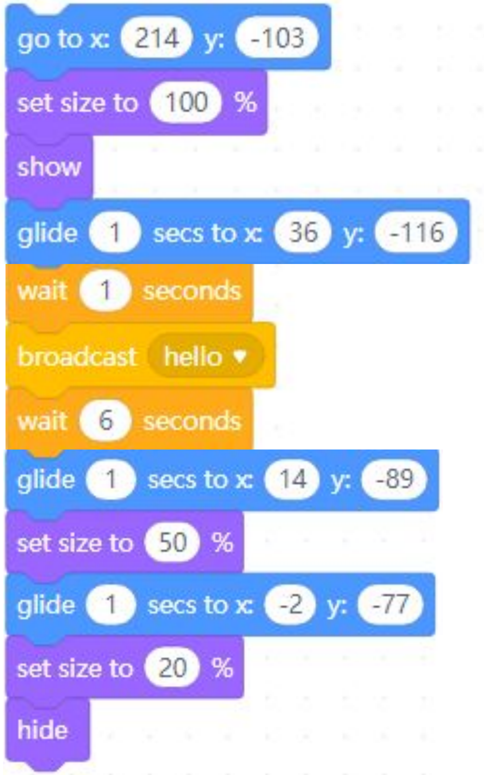
- Third part shows **guests moving to & entering** in the school auditorium



```
glide 1 secs to x: 14 y: -89
set size to 50 %
glide 1 secs to x: -2 y: -77
set size to 20 %
hide
```

- Note: the three have to come **below one another** to make one single code.
- These have to be placed below the **functions event block** as in step above.

.

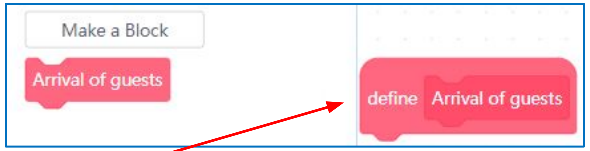


```
go to x: 214 y: -103
set size to 100 %
show
glide 1 secs to x: 36 y: -116
wait 1 seconds
broadcast hello ▾
wait 6 seconds
glide 1 secs to x: 14 y: -89
set size to 50 %
glide 1 secs to x: -2 y: -77
set size to 20 %
hide
```

Step 4 – Creating a Function for the Guests

- The procedure for this is similar to step 2.

- **Name** the function **Arrival of Guests.**
- It’s block statement are in the above slide.
- They have to come below the function definition.

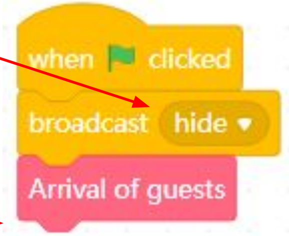


- Since we have two guests, we have to make **functions for both of them** in the same way.
- In reality, this function is **made once & re-used** for all the guests.

Step 5 – Triggering Arrival of Guests

- The **trigger** for guest 1 is - **when green flag clicked.**
- This has to be followed by a message that the guest needs to **send as a broadcast.**
- We have given the message **hide.** It could be any other.
- This hides other guest so that they appear **one by one.**

- Below this we must add steps this guest needs to follow.
- For this we add the block statement **Arrival of guests.**



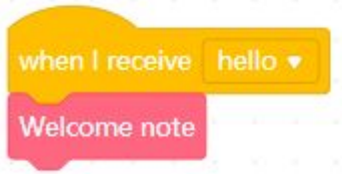
```
when 🏁 clicked
broadcast hide ▾
Arrival of guests
```

- When triggered this function (arrival of guests) will be **automatically recalled.**
- These also need to be made for all the guests.

Step 6 – Triggering Response of Reception Team

- Trigger to start welcome message is **when I receive hello.**
- When guest arrive near reception team they say hello.

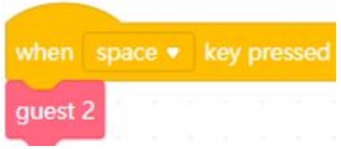
- On receipt, reception team **starts its welcome procedure.**
- To do this we need to call the function named **Welcome note.**
- This will them run all the steps placed in that function.



```
when I receive hello ▾
Welcome note
```

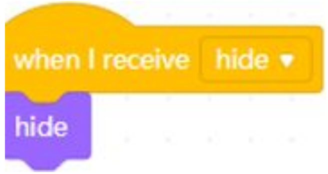
Step 7 – Repeating the Code including its trigger for Guest 2

- The trigger for guest 2 is **when space key pressed.**
- After this comes **hide broadcast message** that the guest must send to other guests yet to come.
- For this we add the **calling function** named guest 2.
- When triggered this function will be called **executed.**



```
when space ▾ key pressed
guest 2
```

Note: Guest 2 is receiving this message from guest 1. This enables him to hide.



```
when I receive hide ▾
hide
```

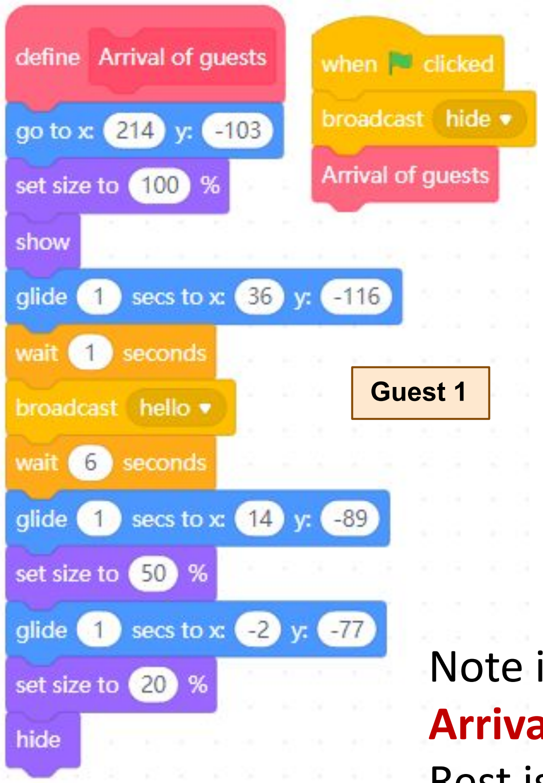
Step 8 – De-Bugging

De-bugging is the process of **testing your code,** observing the faults and rectifying them.

Every code may require certain amount of de-bugging. The most common areas where you could go wrong are:

- **Time between actions.** They may require an increase or decrease.
- **Positioning.** This may require adjustment to give a good look.
- **Actions of Hide and Show.** This may require some adjustments.
- **Wrong broadcast message** done for the reception team (RT) and the guests. So you have to be careful in receiving correct message for RT and next guest

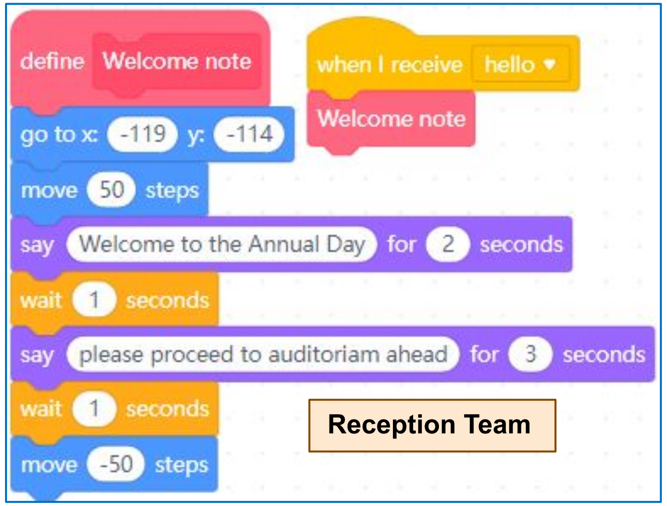
Final Project Codes



```
define Arrival of guests
go to x: 214 y: -103
set size to 100 %
show
glide 1 secs to x: 36 y: -116
wait 1 seconds
broadcast hello ▾
wait 6 seconds
glide 1 secs to x: 14 y: -89
set size to 50 %
glide 1 secs to x: -2 y: -77
set size to 20 %
hide
```

```
when 🏴 clicked
broadcast hide ▾
Arrival of guests
```

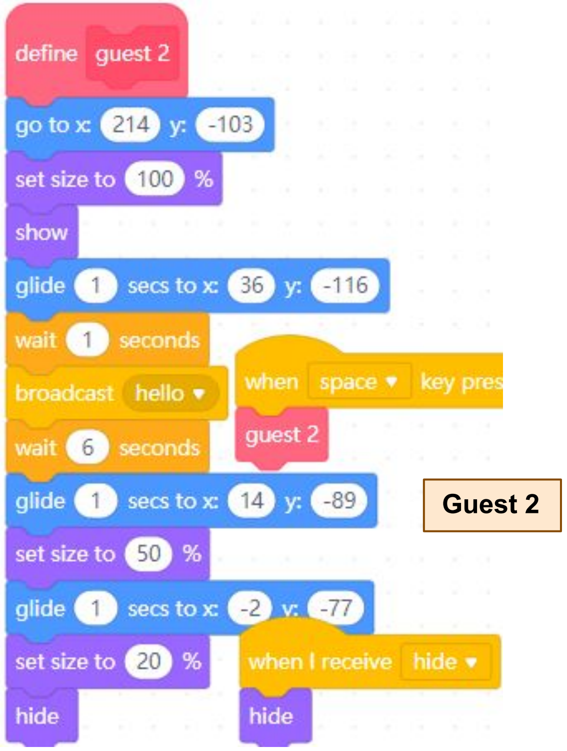
Guest 1



```
define Welcome note
go to x: -119 y: -114
move 50 steps
say Welcome to the Annual Day for 2 seconds
wait 1 seconds
say please proceed to auditoriam ahead for 3 seconds
wait 1 seconds
move -50 steps
```

```
when I receive hello ▾
Welcome note
```

Reception Team



```
define guest 2
go to x: 214 y: -103
set size to 100 %
show
glide 1 secs to x: 36 y: -116
wait 1 seconds
broadcast hello ▾
wait 6 seconds
glide 1 secs to x: 14 y: -89
set size to 50 %
glide 1 secs to x: -2 y: -77
set size to 20 %
hide
```

```
when space ▾ key pres
guest 2
```

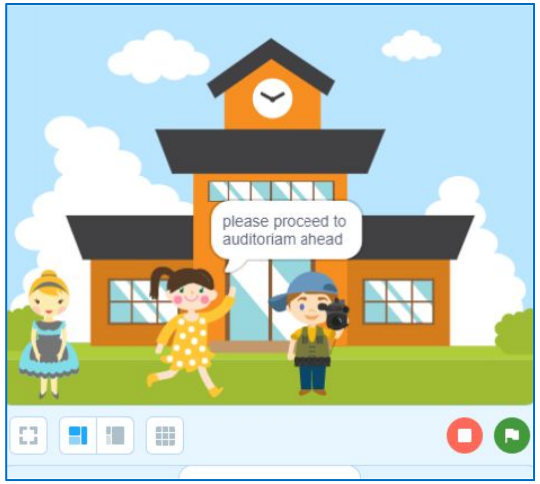
```
when I receive hide ▾
hide
```

Guest 2

Note in this we have created three functions –
Arrival of guests, Welcome note & Guest 2
Rest is similar to other projects.

Running the Code

To Run Code
Click green flag

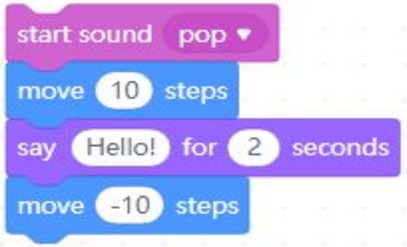


& see the result
This is a small part

Consolidation of this Project & its learning

- Go over the procedure of the above project a few times.
- First by seeing the code and then all by yourself. Keep the code available for ready reference.

- Make three simple projects as per your imagination.
- Start with motion, sound and looks blocks for these small codes.
- Play with positioning & timings.



```
start sound pop ▾
move 10 steps
say Hello! for 2 seconds
move -10 steps
```

- Store the project codes as different functions. Try calling by using different triggers.
- Place them all under one umbrella.
- Move to complicated projects. You will make mistakes, but you will learn.

Now Try Making your Own Blocks.

Code Karega India Badhega