# Sprite predicts your fortune!

Vineet Srivastava

# In this lesson, we will …

- Introduce a very important programming construct called LISTS.
- We will get familiar with the basic operations on a list.
- All this, as we will build a fortune teller game!

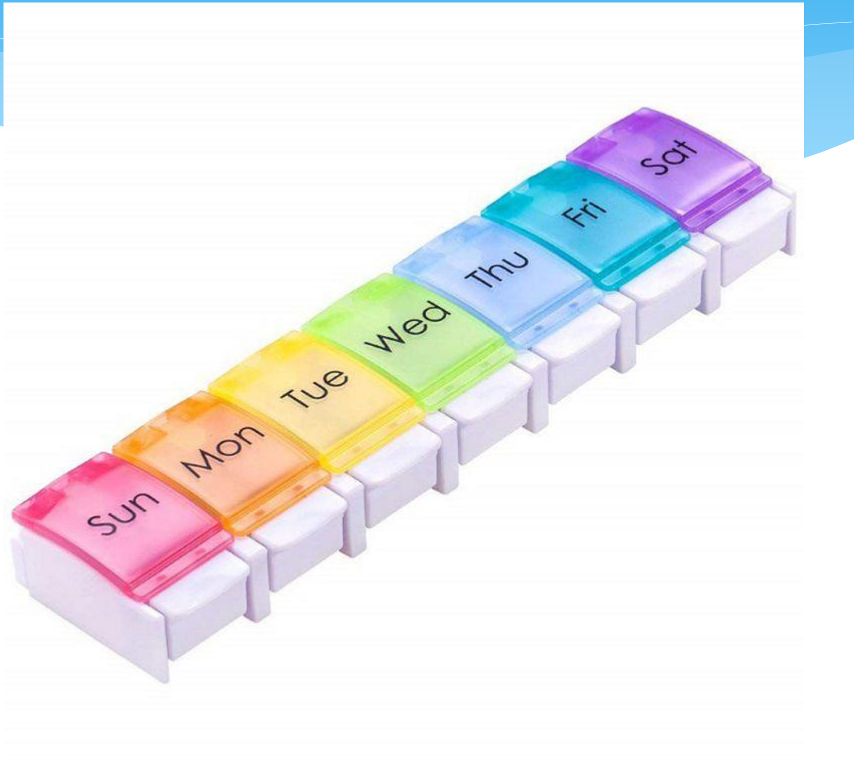
# What is a list?

- You can think of lists as *big brothers* of variables.
- Recall, variables were like containers that helped sprite 'remember' pieces of information.
- Lists are like 'multi-box' containers using which sprites can remember many related things at once. These could be numbers, names, words etc.
  - A bit like a shopping bag.

# What are lists used for?

- Lists are an amazingly powerful concept – using lists our games and projects can really go to the ‘next’ level in terms of complexity and features.
- For now, we will build a FORTUNE TELLER game, just to get familiar with the list blocks and also to have fun ☺.

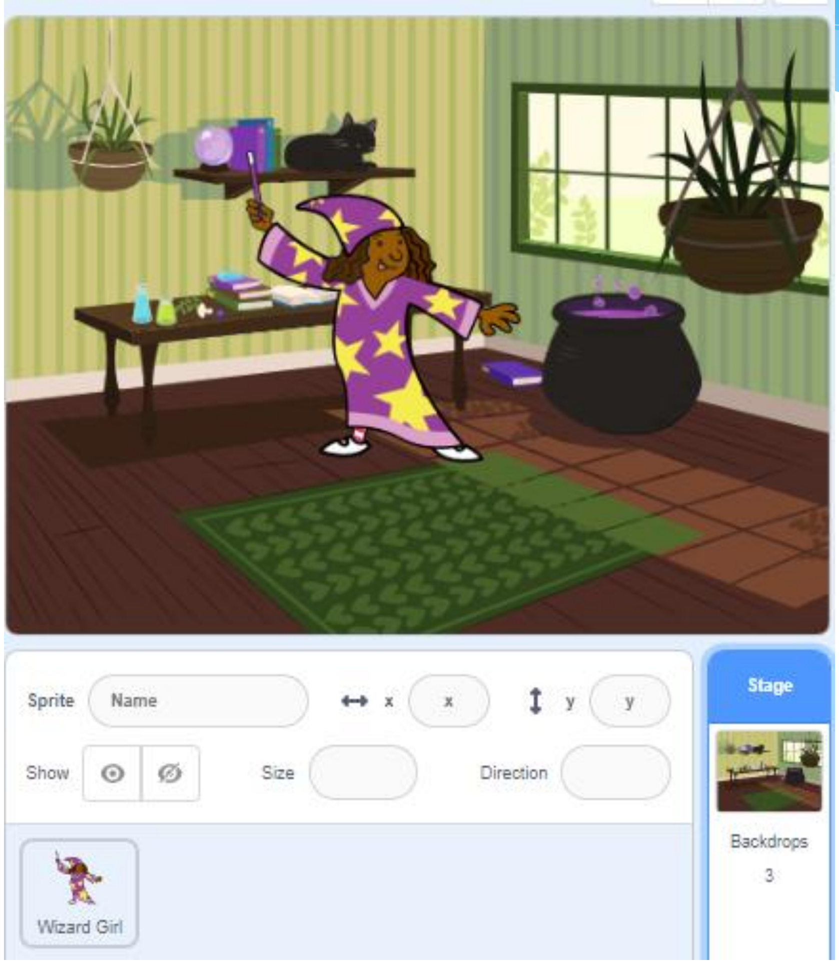
# Variables vs Lists



**Variables**
Separate containers

**Lists**
Several Containers connected to each other, each can be accessed by itself.

# Fortune Teller Game

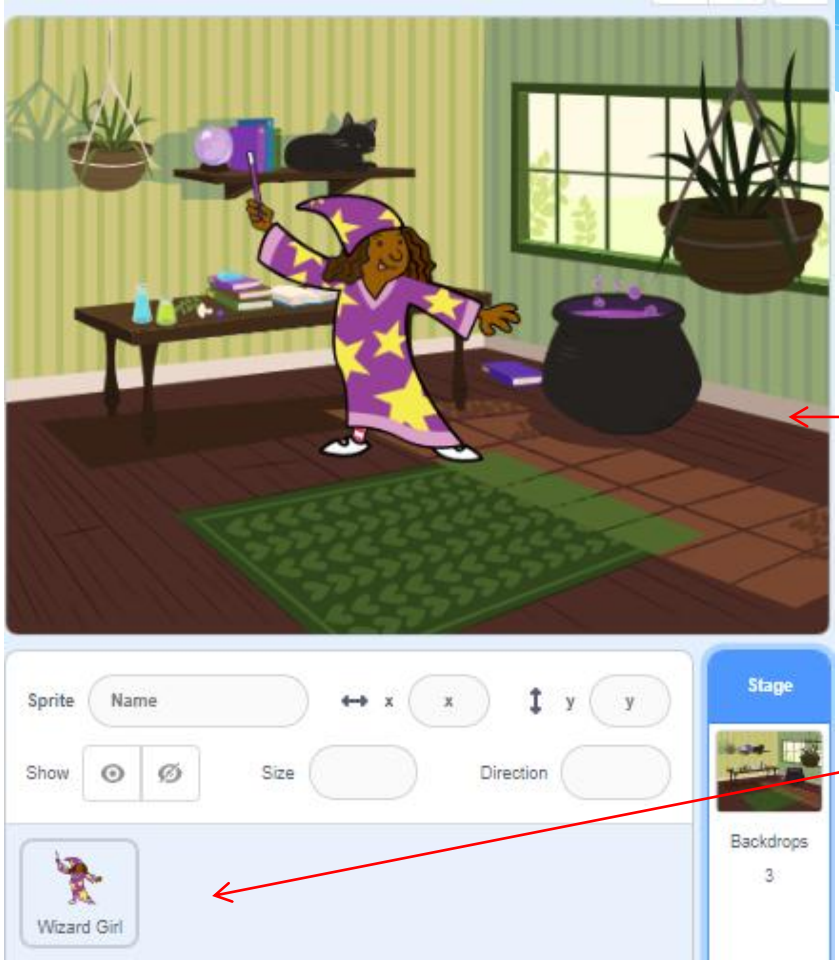


A Wizard girl will answer our questions relating to our ‘fortune’.

It will take the help of a ‘crystal ball’ to get these answers.

(While the game is fun, underneath we use lists to create the answers. So in a way this game also conveys some very deep programming ideas.)

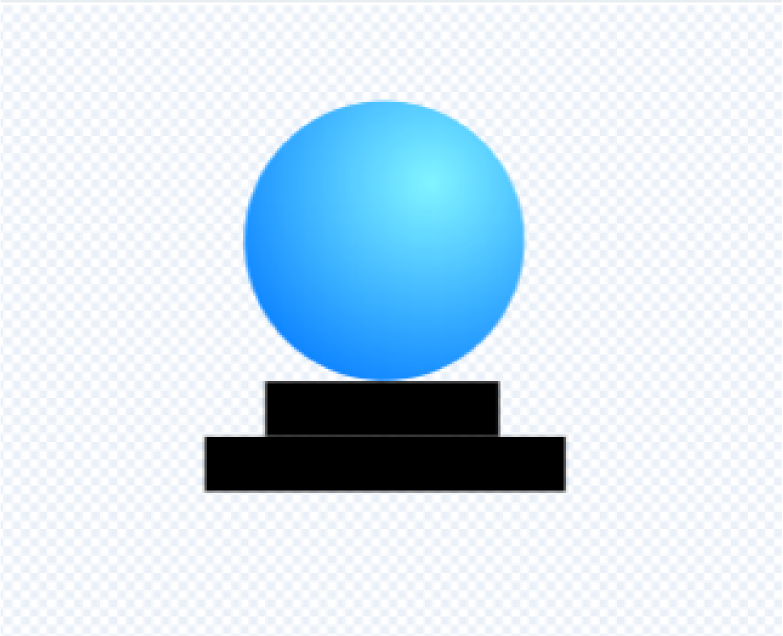
# Sprites and Background

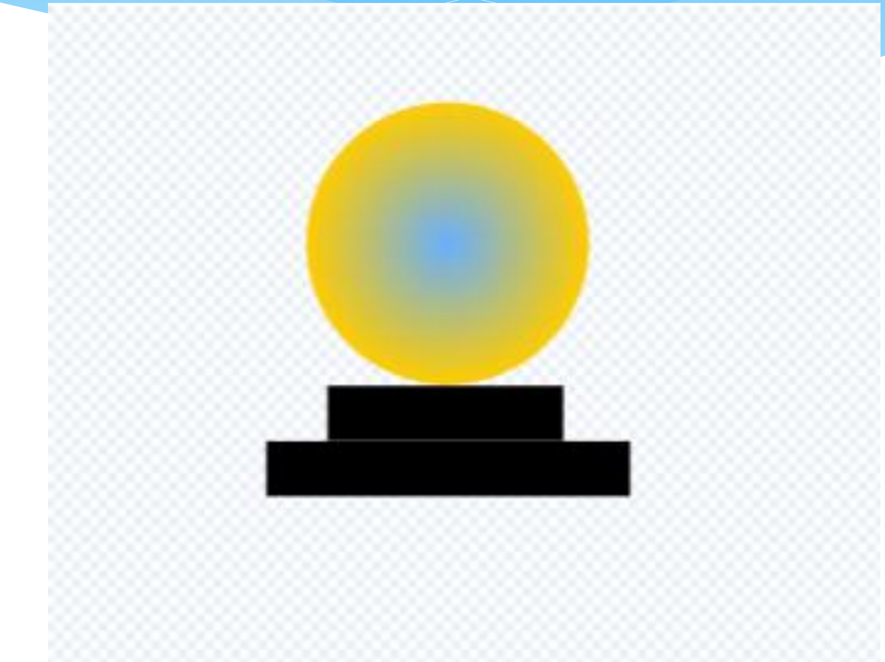


We have used a backdrop called the ‘WITCH HOUSE’

We have used a sprite called ‘WIZARD GIRL’

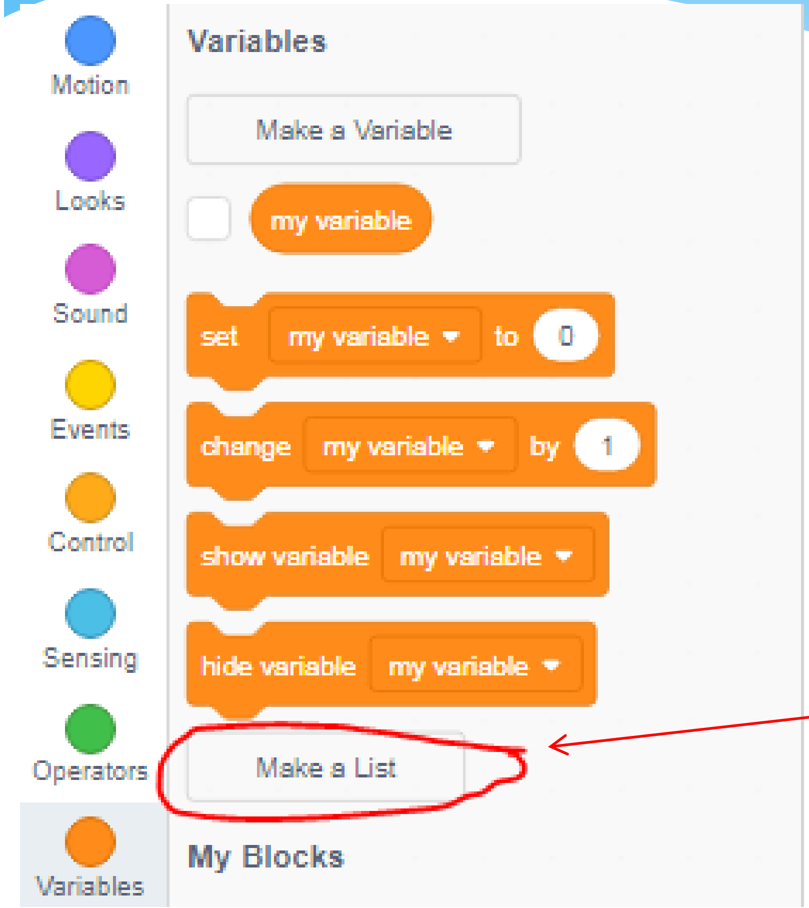
# Add a crystal ball





We can use a combination of existing costumes and paint costumes to create a crystal ball. We have given two costumes to the crystal ball.

# Let us create a list of ‘Wizard Answers’



Variables

Make a Variable

□ my variable

set my variable ▾ to 0

change my variable ▾ by 1

show variable my variable ▾

hide variable my variable ▾

Make a List

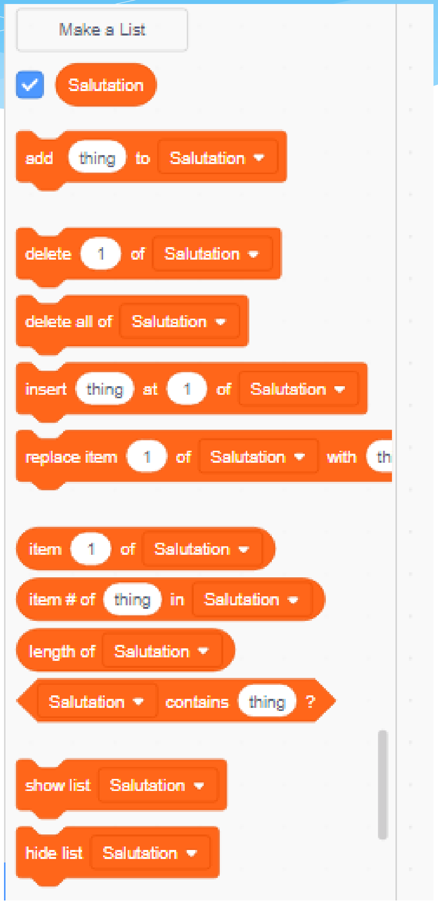
My Blocks

Wizard will pick its answers from a ‘pre-determined’ set of answers. These answers will be stored in a list called ‘Wizard Answers’.

- Click on VARIABLES → MAKE A LIST

# When you create ‘Wizard Answers’

- A lot of new blocks appear …
- These blocks basically contain the different operations that we can perform on the list variables, as we will see now.



```
Make a List
☑ Salutation
add thing to Salutation ▾
delete 1 of Salutation ▾
delete all of Salutation ▾
insert thing at 1 of Salutation ▾
replace item 1 of Salutation ▾ with th
item 1 of Salutation ▾
item # of thing in Salutation ▾
length of Salutation ▾
Salutation ▾ contains thing ?
show list Salutation ▾
hide list Salutation ▾
```

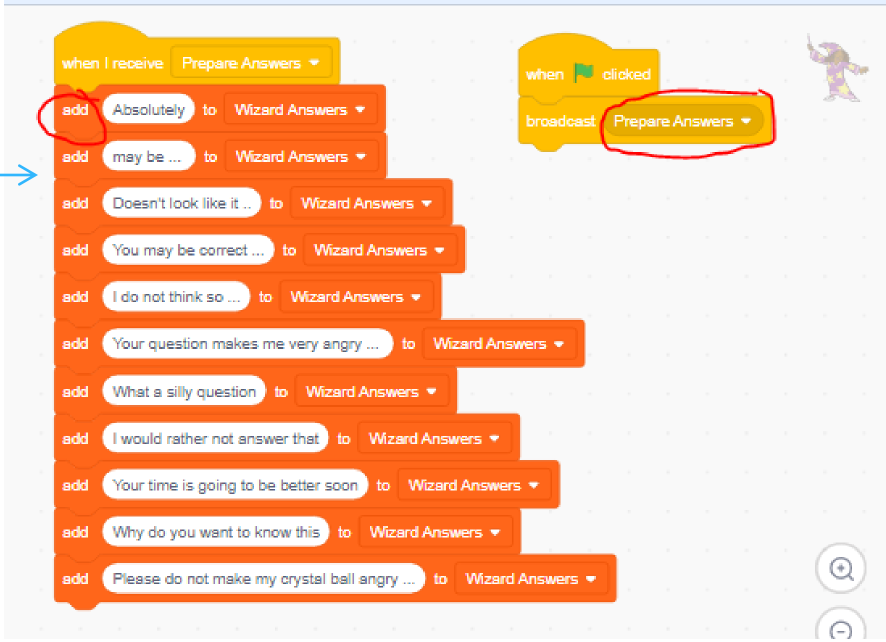
# Populate the list ‘Wizard Answers’

- We will now write a bit of code to populate the list called ‘WIZARD ANSWERS’

We keep adding the answers into the LIST called WIZARD ANSWERS.

ADD is somewhat like the SET in case of variables.

*You can be AS creative here as you want and add as many elements as you like.*

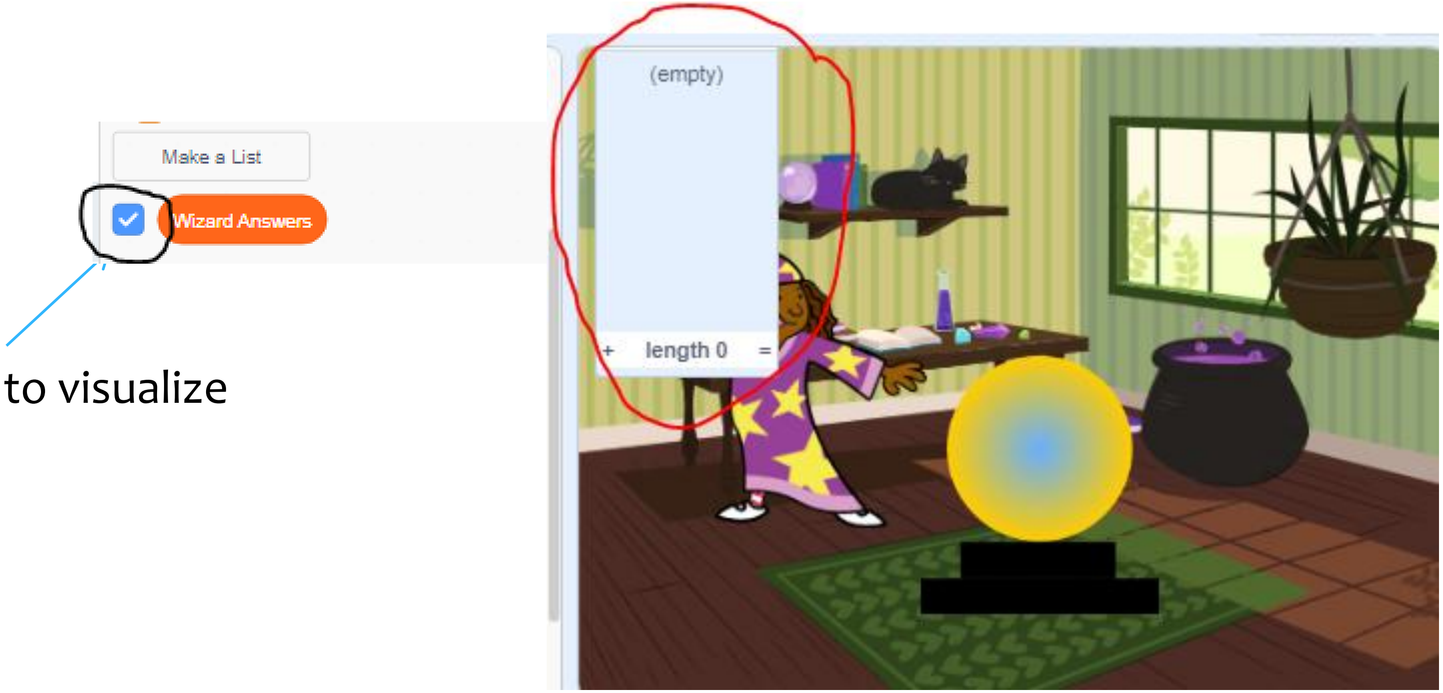


```
when I receive Prepare Answers
add Absolutely to Wizard Answers
add may be ... to Wizard Answers
add Doesn't look like it .. to Wizard Answers
add You may be correct ... to Wizard Answers
add I do not think so ... to Wizard Answers
add Your question makes me very angry ... to Wizard Answers
add What a silly question to Wizard Answers
add I would rather not answer that to Wizard Answers
add Your time is going to be better soon to Wizard Answers
add Why do you want to know this to Wizard Answers
add Please do not make my crystal ball angry ... to Wizard Answers
```

```
when 🏁 clicked
broadcast Prepare Answers
```

# Can we visualize this process?

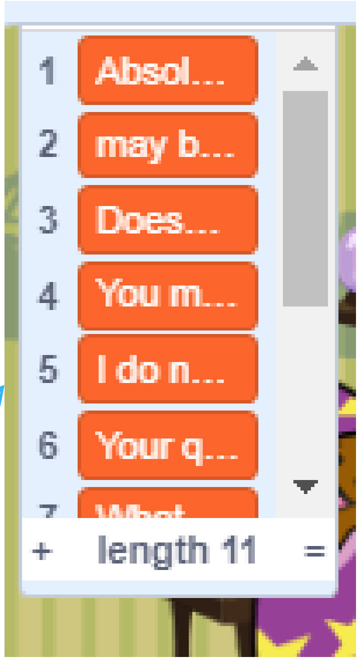
- Click the little tick mark adjacent to the WIZARD ANSWERS



Click here to visualize

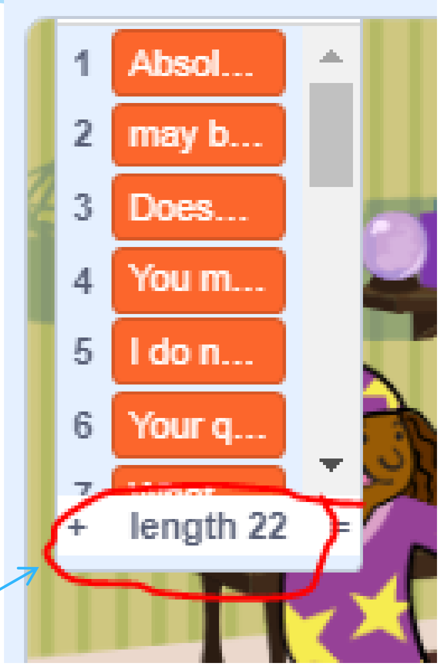
# ‘Visualize’ the list getting populated

- When I click FLAG


1 Absol...
2 may b...
3 Does...
4 You m...
5 I do n...
6 Your q...
7 What...
+ length 11 =

Notice, all the sentences that we were adding to ‘WIZARD ANSWERS’ are here.

- When I click FLAG again

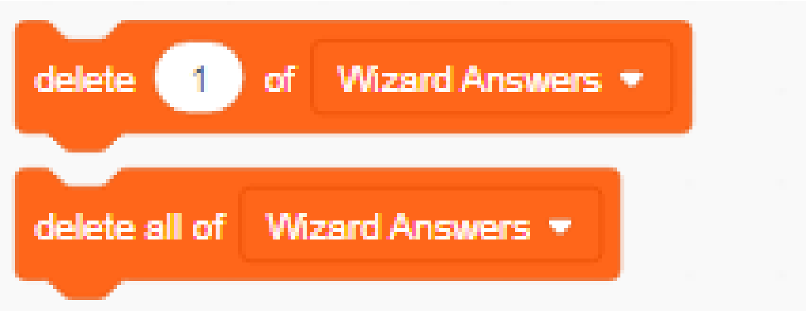

1 Absol...
2 may b...
3 Does...
4 You m...
5 I do n...
6 Your q...
7 What...
+ length 22 =

Notice, all the sentences got added ONCE again. And now the length of the list is 22!

# Preventing this situation!

- To prevent this situation, we have to use the ‘DELETE’ blocks.
- There are two options:

Delete ITEM number 1 from the list. (Instead of 1, we can put another number or a variable).



delete 1 of Wizard Answers ▾

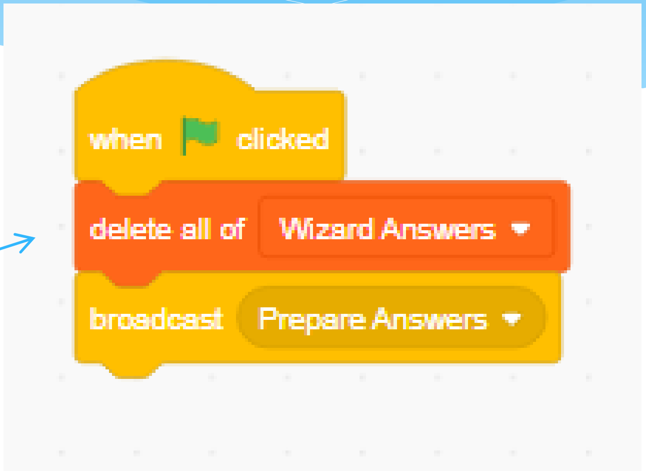
delete all of Wizard Answers ▾

Delete ALL items of the list. We usually need to do this at the beginning of the code.

WIBYTE

# Updating the code

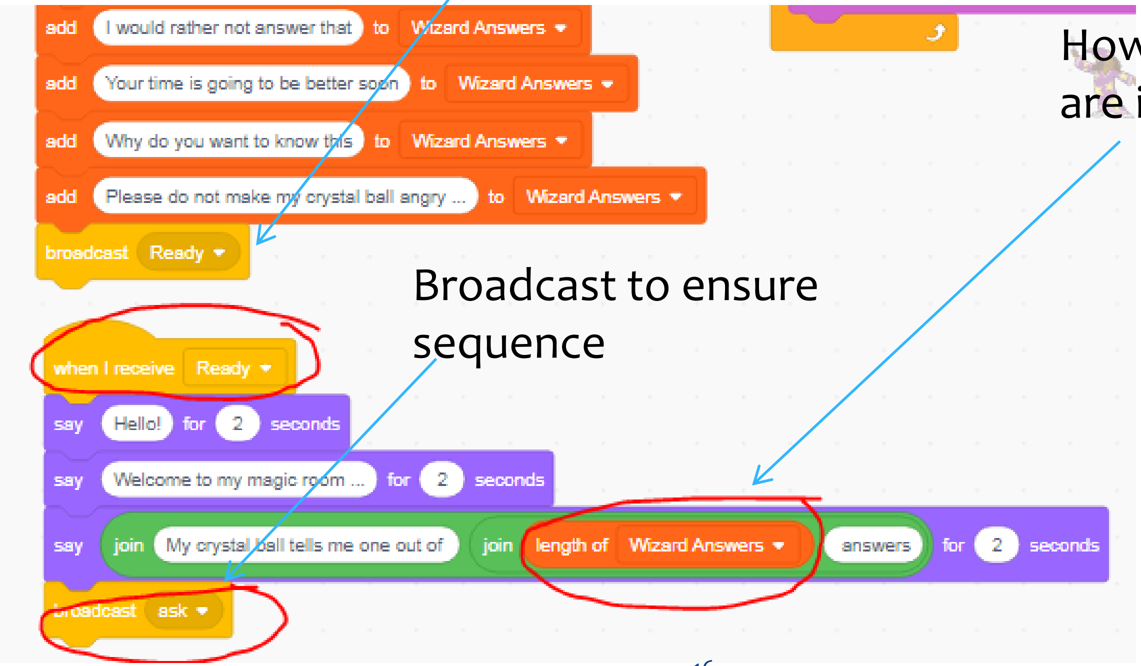
Before starting to populate the list, clear it up. (Delete all of its contents).



```
when 🏴 clicked
delete all of Wizard Answers ▾
broadcast Prepare Answers ▾
```

# Wizard greets you!

- Use broadcast to ensure that the list is fully populated BEFORE the other parts of the program start.



```
add (I would rather not answer that) to [Wizard Answers ▾]
add (Your time is going to be better soon) to [Wizard Answers ▾]
add (Why do you want to know this) to [Wizard Answers ▾]
add (Please do not make my crystal ball angry ...) to [Wizard Answers ▾]
broadcast (Ready ▾)

when I receive [Ready ▾]
say (Hello!) for (2) seconds
say (Welcome to my magic room ...) for (2) seconds
say (join (My crystal ball tells me one out of) (join (length of [Wizard Answers ▾]) (answers))) for (2) seconds
broadcast (ask ▾)
```

How many elements are in the list?

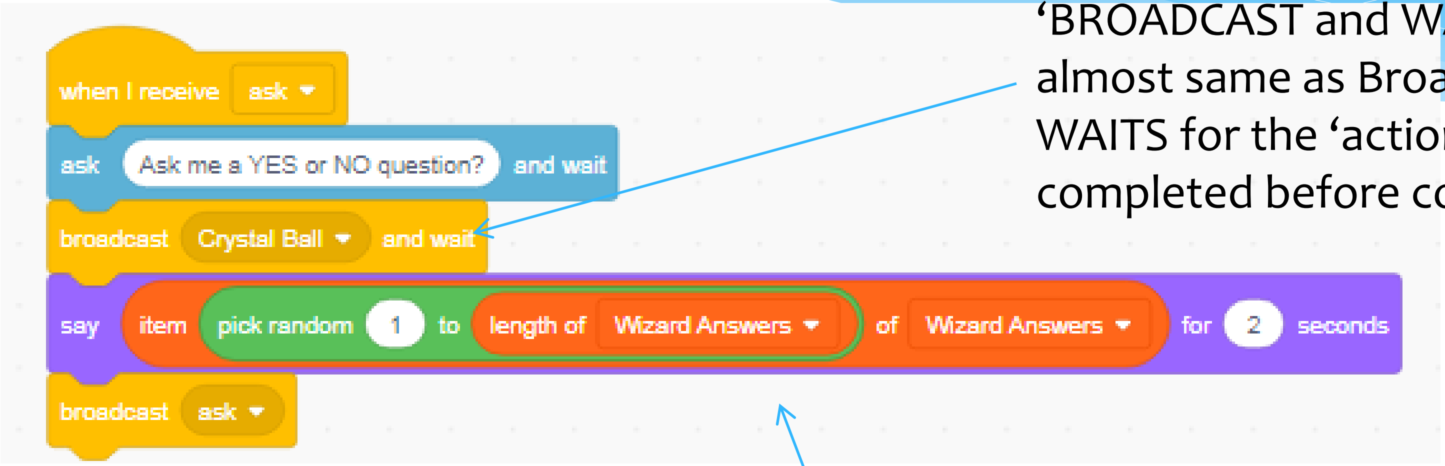
Broadcast to ensure sequence

# Next Features

- Wizard will ask us to ask a question.
  - We will build this capability including wizards response.
- Wizard will keep track of which questions we have asked previously and if we ask a repeated question, the wizard will disappear.
- We will build these two independently, and then merge.

# Wizard's question and response

Effect on CRYSTAL Ball
Notice. We have intentionally used 'BROADCAST and WAIT' here. This is almost same as Broadcast , but it WAITS for the 'action' to get completed before continuing.

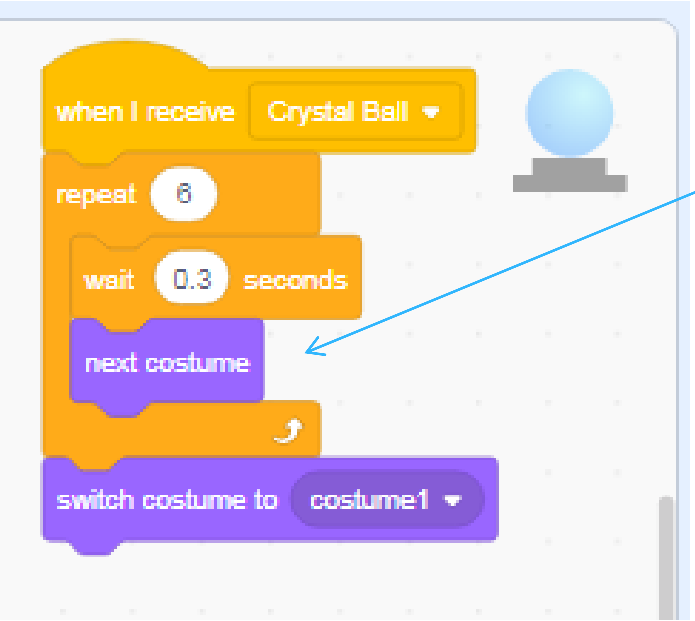


```
when I receive [ask ▾]
ask (Ask me a YES or NO question?) and wait
broadcast [Crystal Ball ▾] and wait
say (item (pick random (1) to (length of [Wizard Answers ▾])) of [Wizard Answers ▾]) for (2) seconds
broadcast [ask ▾]
```

Ask a question again

Wizard's response (See Slide 18)

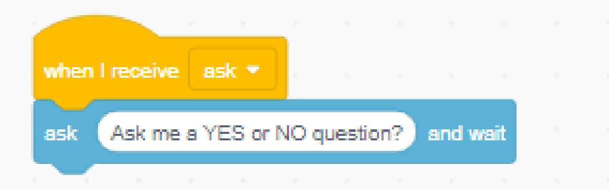
# On the CRYSTAL BALL



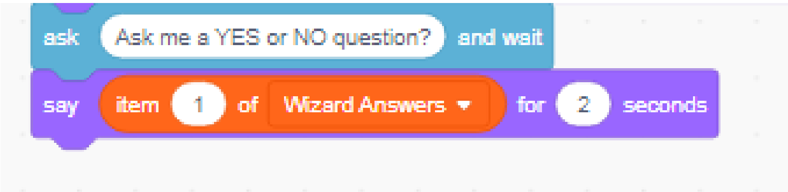
```
when I receive Crystal Ball ▾
repeat 6
    wait 0.3 seconds
    next costume
↺
switch costume to costume1 ▾
```

Create a blinking effect by using next costume.

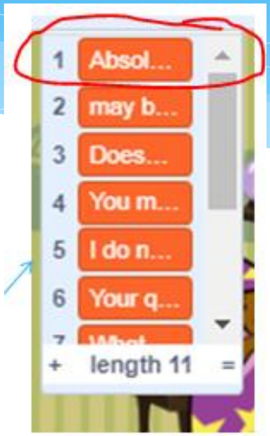
# Look at Wizard’s response again



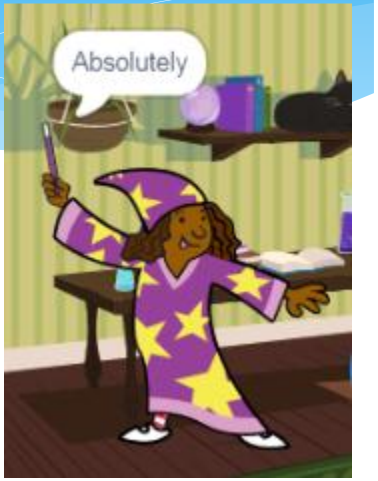
```
when I receive ask
ask Ask me a YES or NO question? and wait
```



```
ask Ask me a YES or NO question? and wait
say item 1 of Wizard Answers for 2 seconds
```

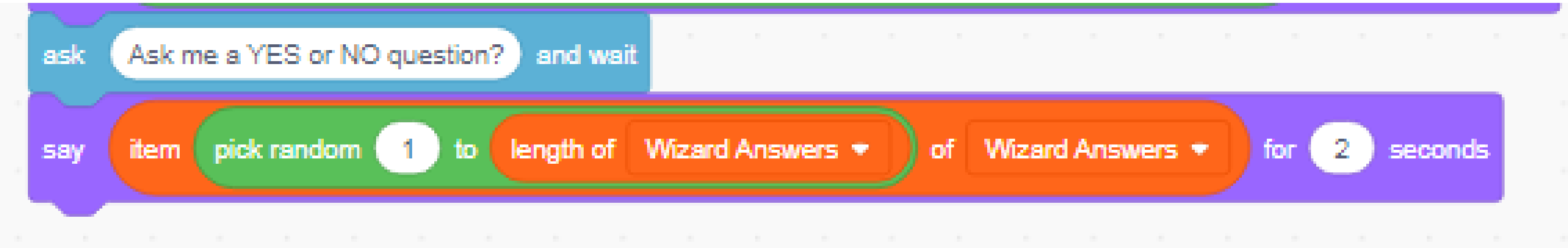


```
1 Absol...
2 may b...
3 Does...
4 You m...
5 I do n...
6 Your q...
7 What...
+ length 11 =
```



Let’s say we had used the above code. In this case, the WIZARD picks up item # 1. Hence always answers ‘Absolutely’

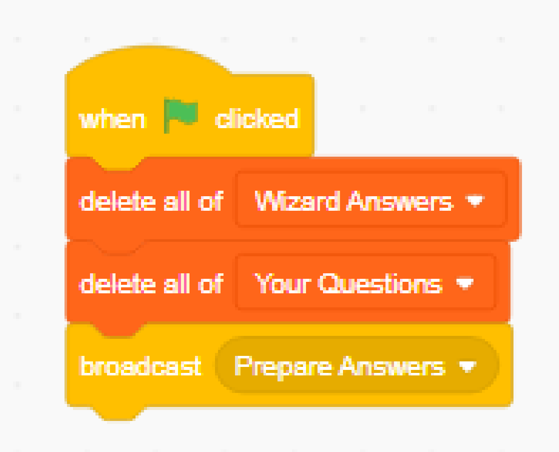
But, to make it more interesting, we have made the wizard pick a ‘random’ answer from the list of answers .



```
ask Ask me a YES or NO question? and wait
say item pick random 1 to length of Wizard Answers of Wizard Answers for 2 seconds
```

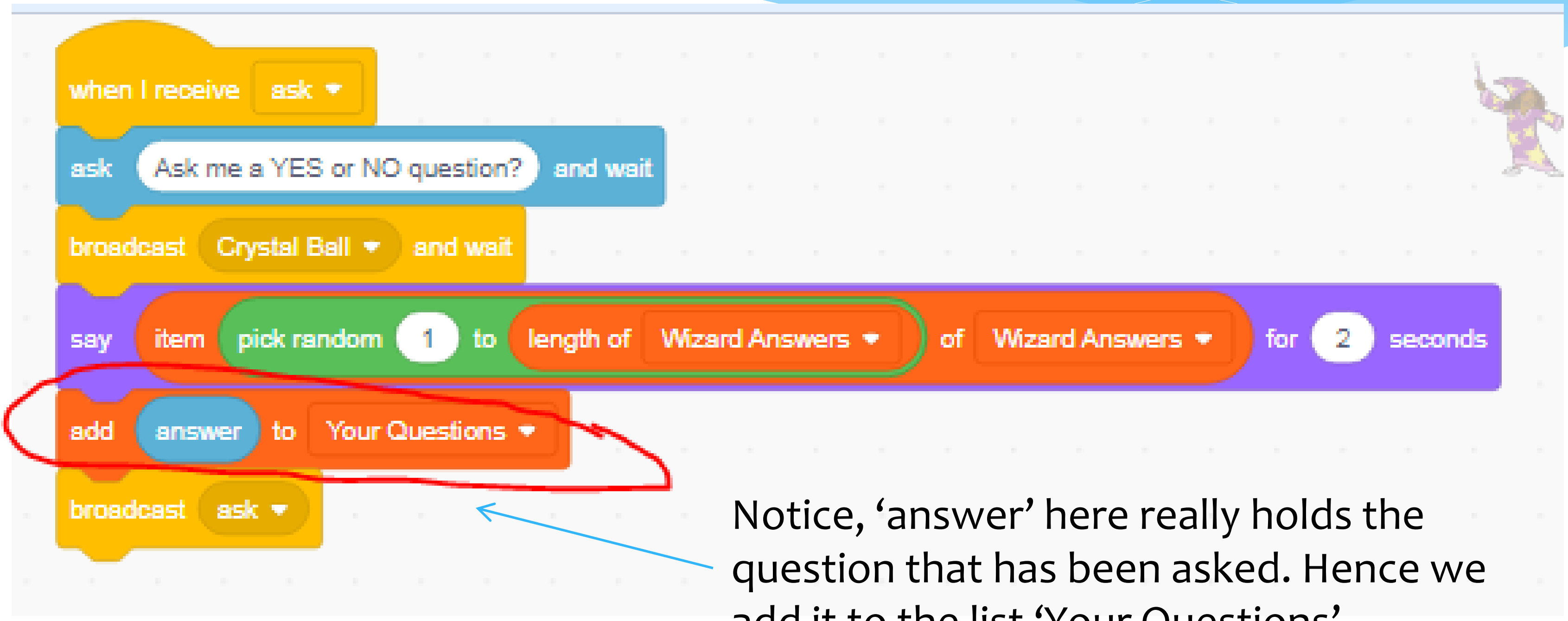
# Keeping track of your questions

- Wizard can keep track of the questions that you are asking.
- It can do so by adding the questions to a list called ‘YOUR QUESTIONS’.
- Create a list called ‘YOUR QUESTIONS’.  Delete all of it in the beginning.



```
when 🏴 clicked
delete all of Wizard Answers ▾
delete all of Your Questions ▾
broadcast Prepare Answers ▾
```

# Updating the ‘YOUR QUESTIONS’ list

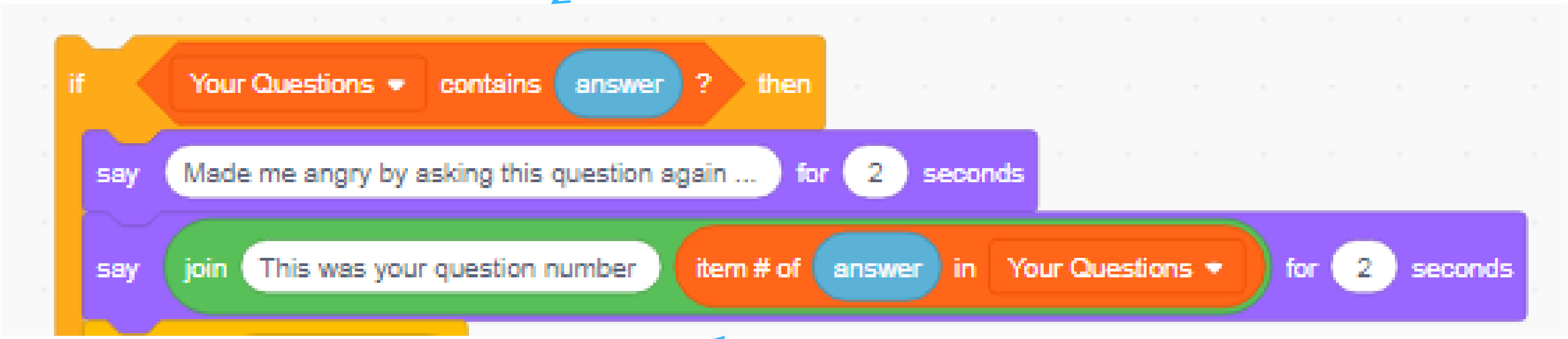


```
when I receive [ask ▾]
ask (Ask me a YES or NO question?) and wait
broadcast (Crystal Ball ▾) and wait
say (item (pick random (1) to (length of [Wizard Answers ▾])) of [Wizard Answers ▾]) for (2) seconds
add (answer) to [Your Questions ▾]
broadcast (ask ▾)
```

Notice, ‘answer’ here really holds the question that has been asked. Hence we add it to the list ‘Your Questions’

# Finding out if the question has been asked before

Does the list already contain this question?

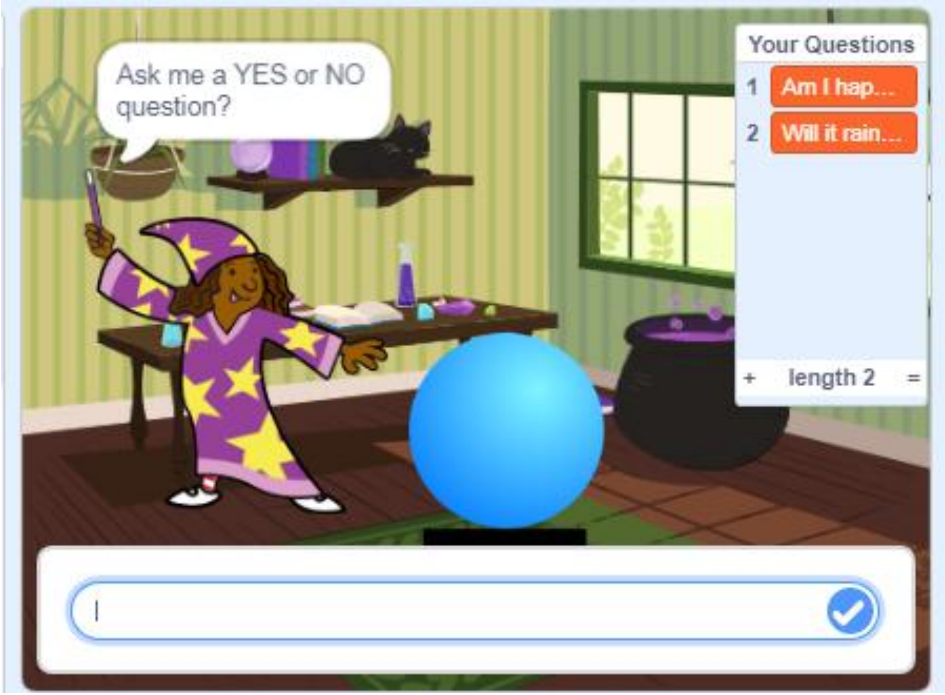


```
if <[Your Questions ▾] contains (answer) ?> then
    say [Made me angry by asking this question again ...] for (2) seconds
    say (join [This was your question number] (item # of (answer) in [Your Questions ▾])) for (2) seconds
```

If YES, then at what item number?

WIBYTE

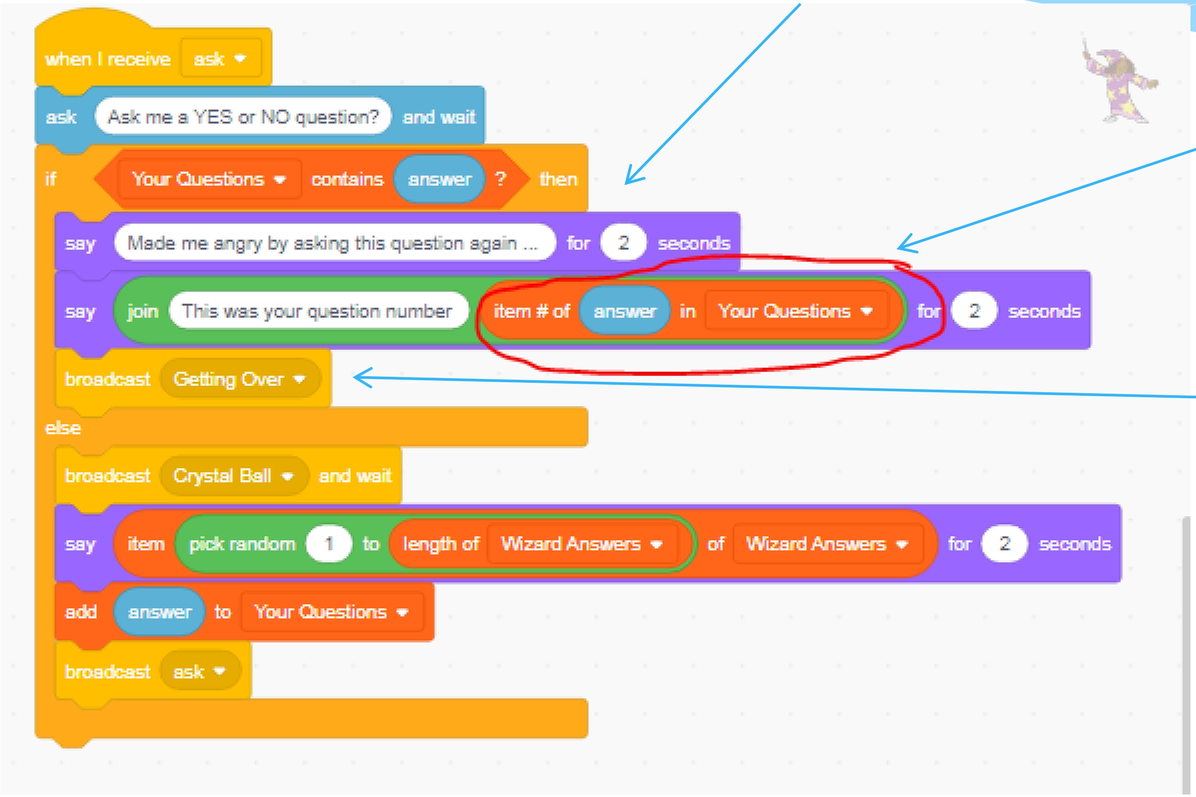
# Example



Notice, all the questions that we are asking are now getting stored in this list.

Hence for a new added question, we can check if the question has previously been asked.

# Putting it all together



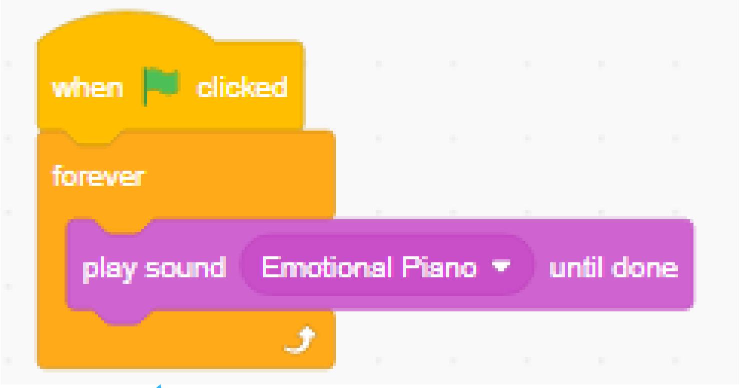
```
when I receive ask ▾
ask Ask me a YES or NO question? and wait
if Your Questions ▾ contains answer ? then
    say Made me angry by asking this question again ... for 2 seconds
    say join This was your question number item # of answer in Your Questions ▾ for 2 seconds
    broadcast Getting Over ▾
else
    broadcast Crystal Ball ▾ and wait
    say item pick random 1 to length of Wizard Answers ▾ of Wizard Answers ▾ for 2 seconds
    add answer to Your Questions ▾
    broadcast ask ▾
```

You had previously asked this question

The item number for this question.

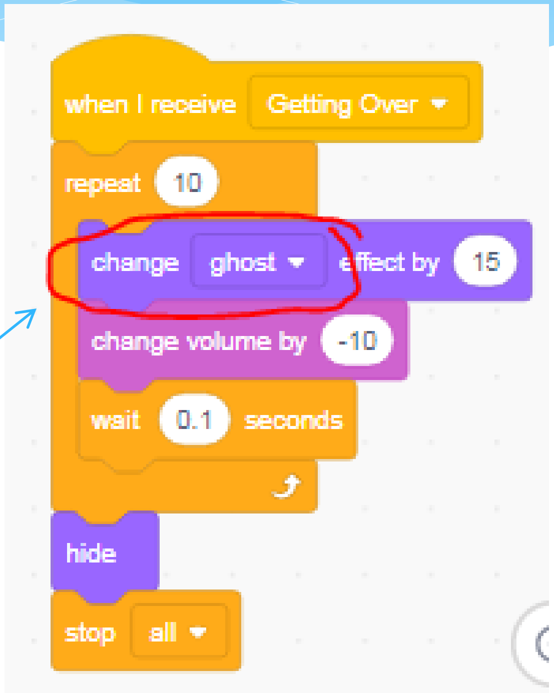
Getting over

# Background Music and Ending Conditions



```
when 🏁 clicked
forever
    play sound Emotional Piano ▾ until done
```

Background Music



```
when I receive Getting Over ▾
repeat 10
    change ghost ▾ effect by 15
    change volume by -10
    wait 0.1 seconds
hide
stop all ▾
```

Slowly Disappear

# You are all set!

- Lists are a very powerful concept. But it takes some practice to get used to the different options they provide.
- This basic activity teaches us the uses of lists in a fun way.
- Try out this activity. Apart from concept of lists, this will also be a good practice of the concepts of cloning/broadcasting etc.

# Extra Innings

# Ideas to spice up the game!

- Use multiple lists for the wizard answers. Combine their elements randomly.
- Create an innovative storyline. May be the answers the wizard provides can be related to a theme.

# Add vs Import

- There is also a feature in SCRATCH to import a list from outside.
- This allows us to populate lists by filling the lists elsewhere (e.g. in notepad).
- However, this cannot be done ‘while’ the program is running. Hence it is somewhat limited in use.
- In general, lists will be used to store information about sprites (e.g. their locations, sizes etc) *during* the program, in which case we have to use add, like we have done in the class.