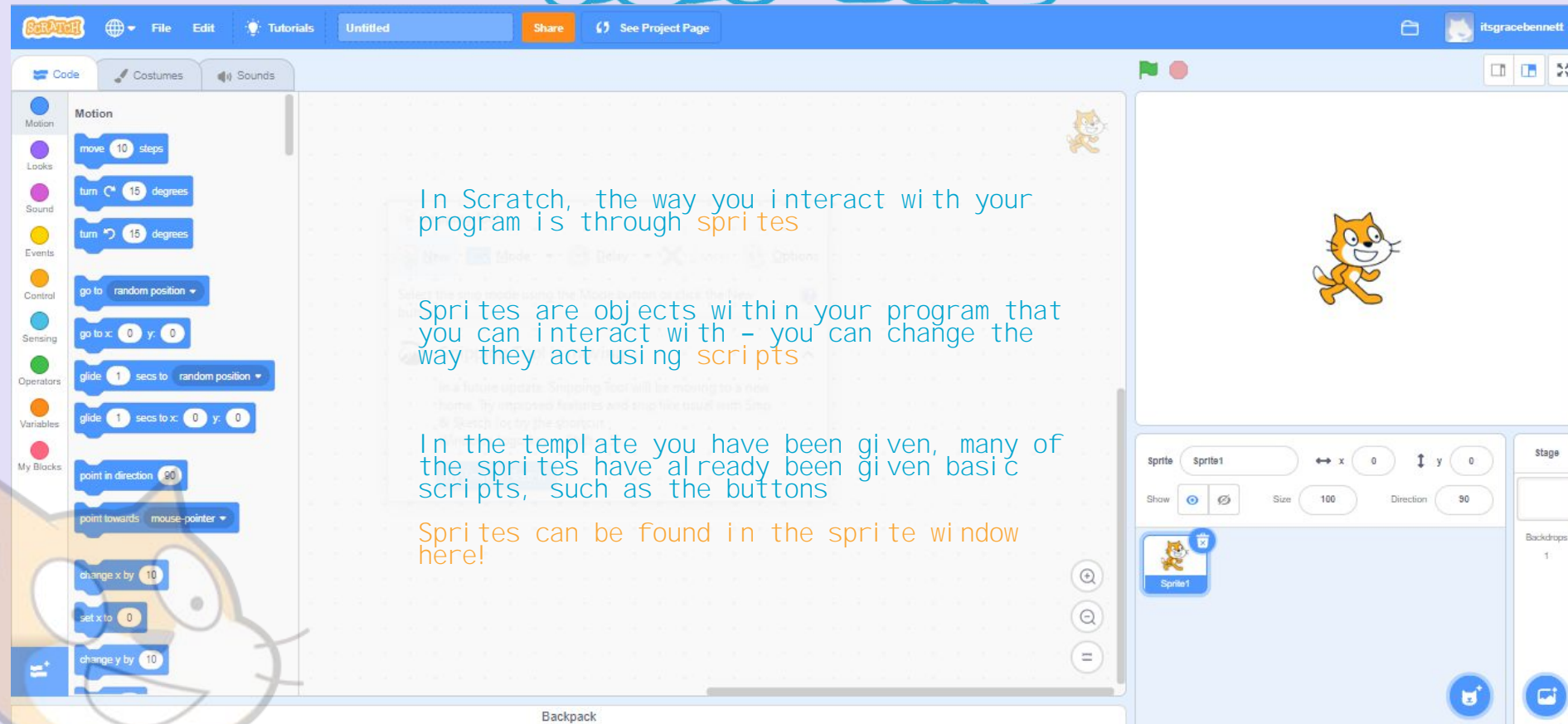


# Dentistry Booking System

*BEGINNER*

# How does SCRATCH work?



The screenshot shows the Scratch web interface. At the top is a blue header with the Scratch logo, a globe icon, and menu options: File, Edit, Tutorials, Untitled, Share, and See Project Page. Below the header are three tabs: Code, Costumes, and Sounds. The left sidebar contains a vertical menu of categories: Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, and My Blocks. The main workspace is a large white area with a grid background. In the center, there is a semi-transparent text box containing the following text:

In Scratch, the way you interact with your program is through **sprites**

Sprites are objects within your program that you can interact with - you can change the way they act using **scripts**

In the template you have been given, many of the **sprites** have already been given basic **scripts**, such as the buttons

Sprites can be found in the **sprite window** here!

The right side of the interface features a stage window showing a white stage with a small orange cat sprite in the center. Below the stage is a control panel for the selected sprite, labeled 'Sprite1'. It includes fields for x and y coordinates (both set to 0), a 'Show' button, a 'Size' field (set to 100), and a 'Direction' field (set to 90). At the bottom right of the control panel are icons for 'Backdrops' and 'Sprites'.

# How does SCRATCH work?

The image shows a screenshot of the Scratch web-based programming environment. The interface is divided into several sections:

- Code Section (Left):** A vertical sidebar containing various code blocks categorized by function (Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, My Blocks). A purple arrow points to this section with the text: "This is the code section! Here is where we can drag and drop blocks of code. Blocks are colour coded depending on their function".
- Display Window (Center):** A large white area where the sprite (Scratch Cat) is visible. An orange arrow points to this area with the text: "This is our display window, where our sprites will appear and interact with the user".
- Sprite Window (Bottom Right):** A panel showing the selected sprite (Sprite1) and its properties (x: 0, y: 0, Size: 100, Direction: 90). An orange arrow points to this window with the text: "Sprites can be found in the sprite window here!".

The Scratch logo is prominently displayed in the top center of the interface.

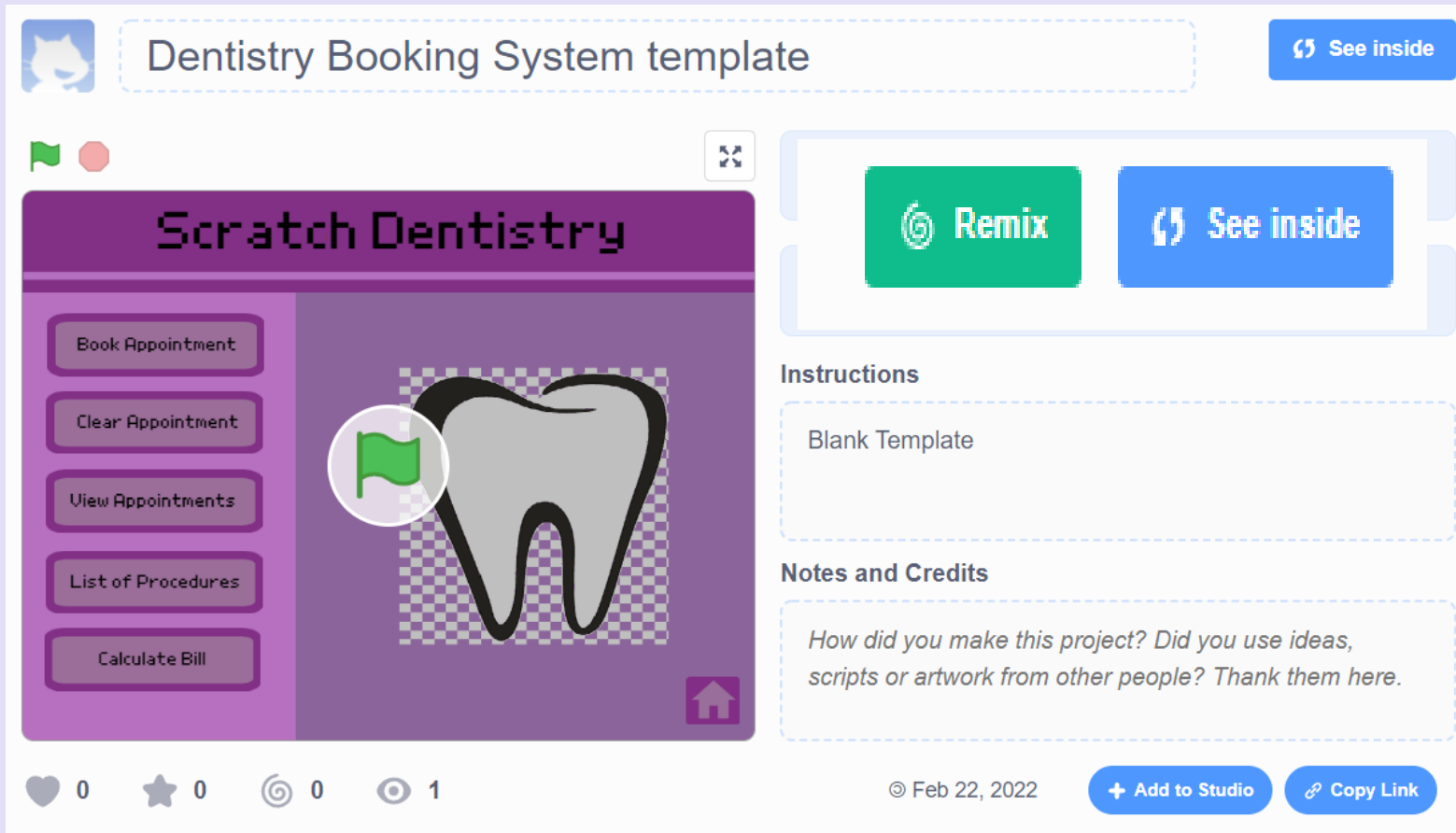
# Activity scenario summary:

In this example we will code a project, which replicates a simple Dentistry Booking System, used in a dental practice.

This project will help you develop an understanding how coding is used in dentistry. As part of the activity, participants will design a software in which the dentistry staff will be able to;

- a) Book Appointments
- b) Clear Appointment slots
- c) View Booking Lists
- d) View Procedure Lists
- e) Calculate Bill

First, we are going to program a basic appointment check-in and out system for our dental practice



Once you have logged into Scratch...

Click [here](#) to access the template to our project!

Click REMIX to get started with your project

By the end of this lesson, your project should look like [this](#)!  
(Use this if you get stuck)

# The Menu:

---

This is the menu! In the beginner section we will be adding functionality to the first two buttons.



The screenshot displays the Scratch IDE interface for a project titled "Dentistry Booking System". The code editor shows a sequence of six "when I receive" blocks, each followed by a "hide" block. The stage area shows a "Scratch Dentistry" interface with a large tooth sprite and several buttons: "Book Appointment", "Clear Appointment", "View Appointments", "List of Procedures", and "Calculate Bill". The sprite palette on the right shows the "Tooth" sprite selected, with a purple arrow pointing to it. The interface also includes a "Backpack" area at the bottom.

This window should look something like this!

Make sure that you have selected the **tooth** sprite.

# Booking an Appointment



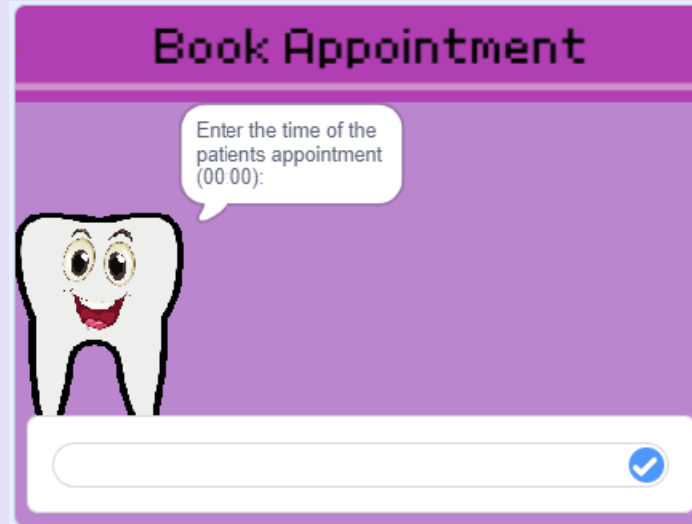
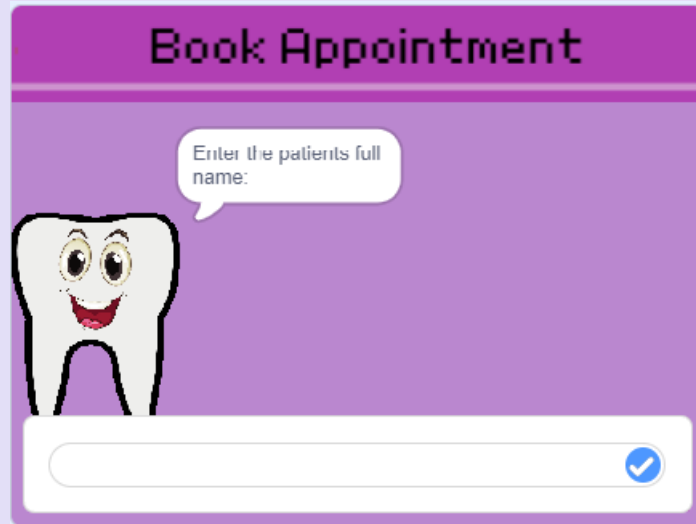
This button will allow the user to book an appointment to the dentist.

It will ask the user to input their full name and the time that they would like to go in for.

If the appointment has been booked successfully then a confirmation message will come up.



# What it will look like..

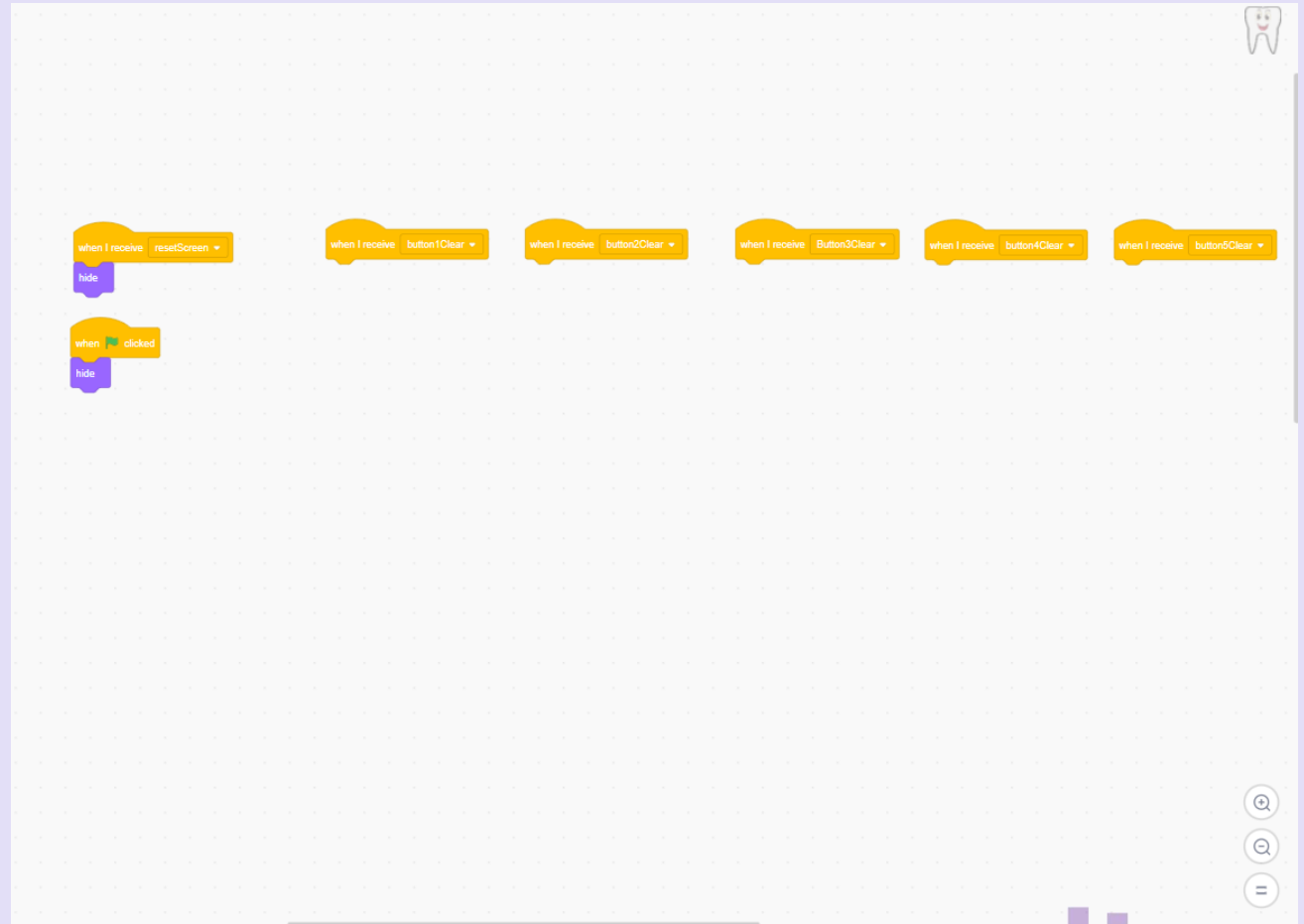


# First Look:

When we first open the tooth sprite it should look like this.

All of the code for Book Appointment will be attached to button1Clear block.

This means that when the button is clicked ONLY the code for Book Appointment will run.



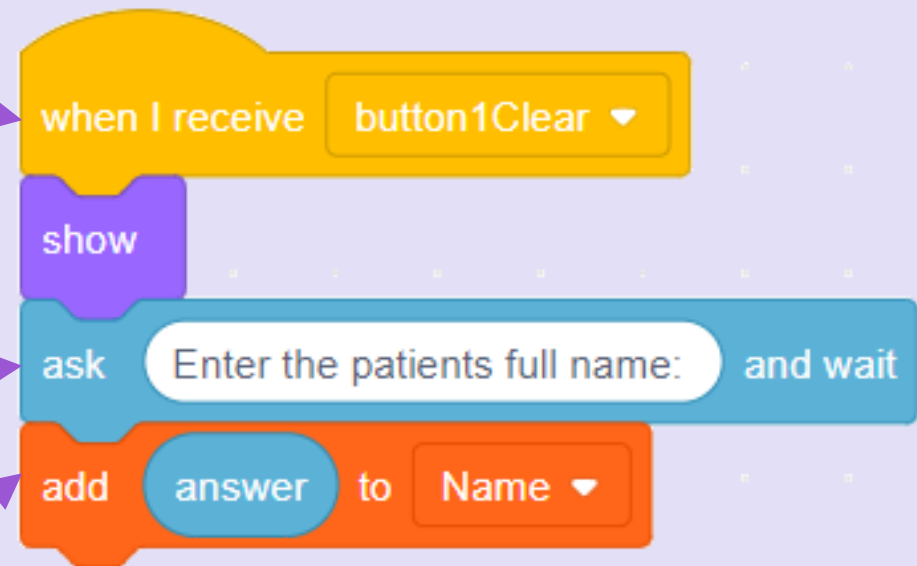
# Step 1

Once 'Book Appointment is clicked

When the button is clicked this code will run and the screen is cleared.

The ASK block will ask the user to enter the patients full name and store their input into the answer block and wait.

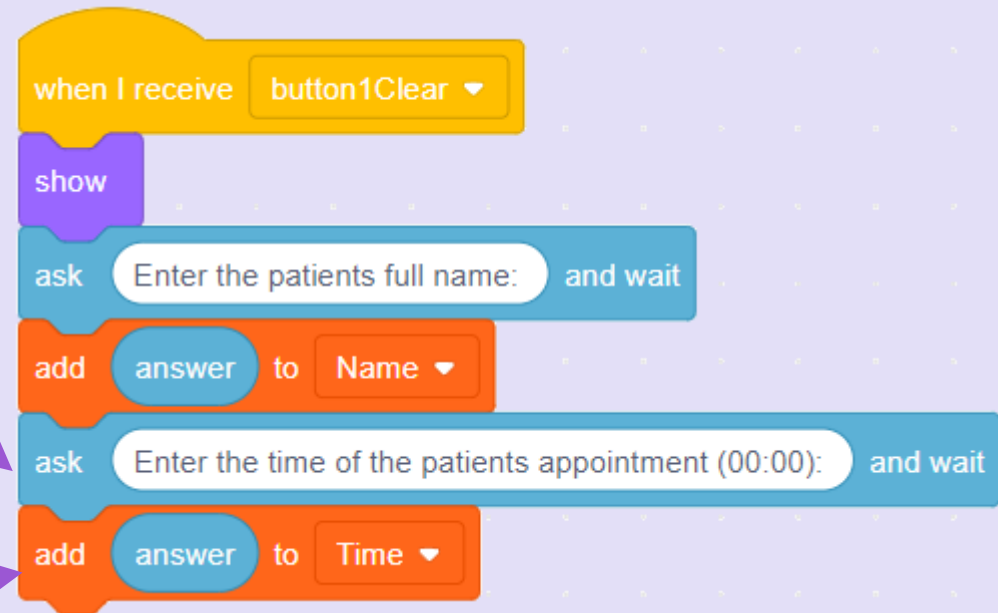
Then the answer is added to the list name.



# Step 2

Another ASK block has been used to ask the user what time they want the appointment for.

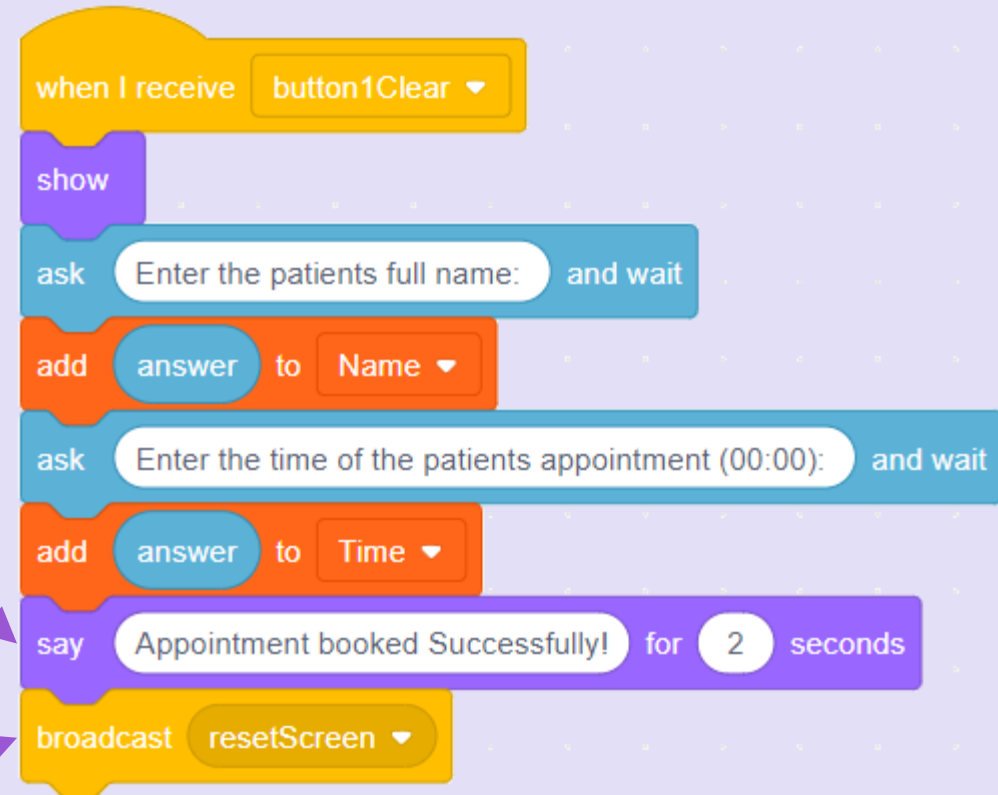
The information input by the user is stored in the answer block again and stored in the list time.



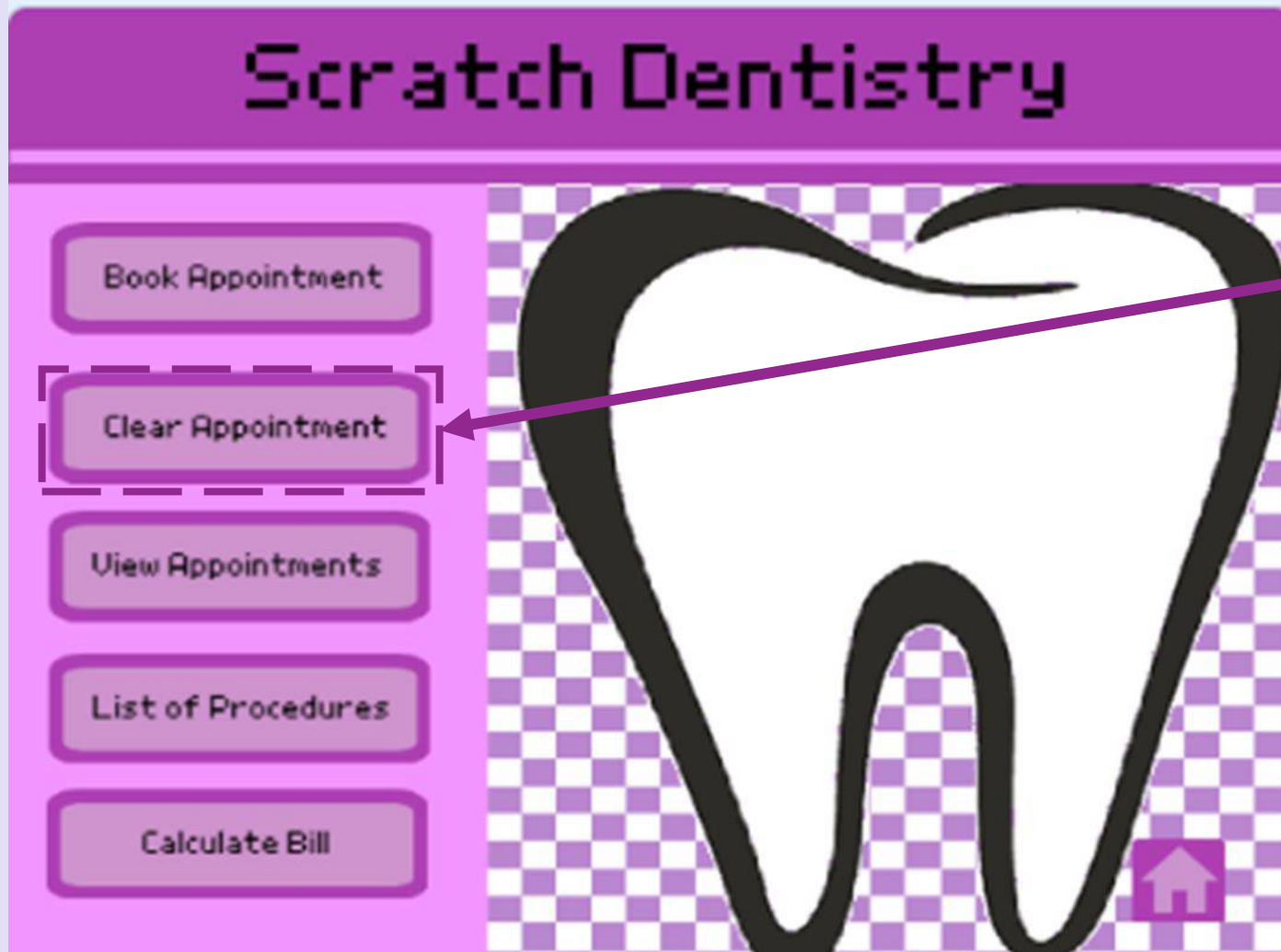
# Step 3

We can then display a message to confirm the booking.

Then the screen resets back to the home screen ready for the next command.



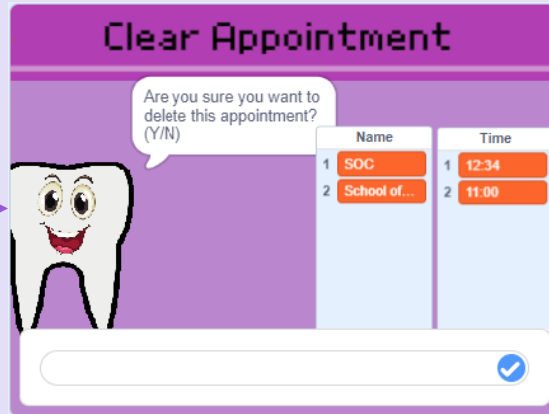
# Clear Appointment



This button will allow the user to clear an appointment from the current booking list.

It will ask what name you would like to delete and once input into the program the whole appointment will be gone.

# What it will look like..



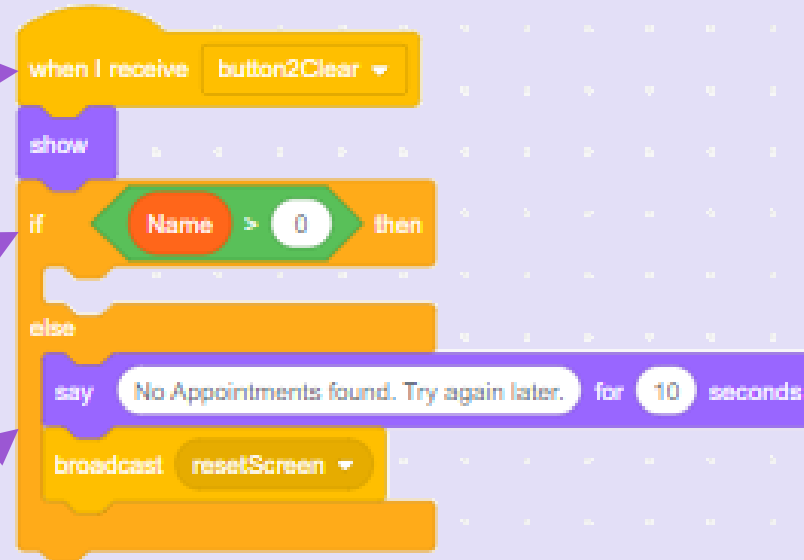
# Step 1

Once 'Clear Appointment' is clicked

When the button is clicked, this code will run, and the screen will be cleared.

The IF ELSE block checks to see if the list of names is greater than zero then all the code inside the IF block will run.

If the list isn't greater than zero then the code within the else statement will run.



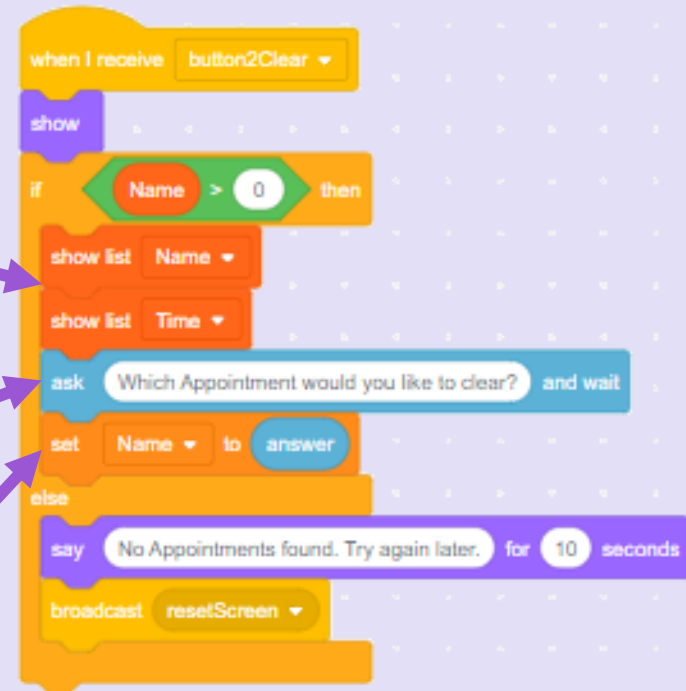


# Step 2

Both lists are shown so the user can see the name and time of the appointment.

The user is then asked which appointment they would like to clear.

The answer is then stored inside a variable called Name.

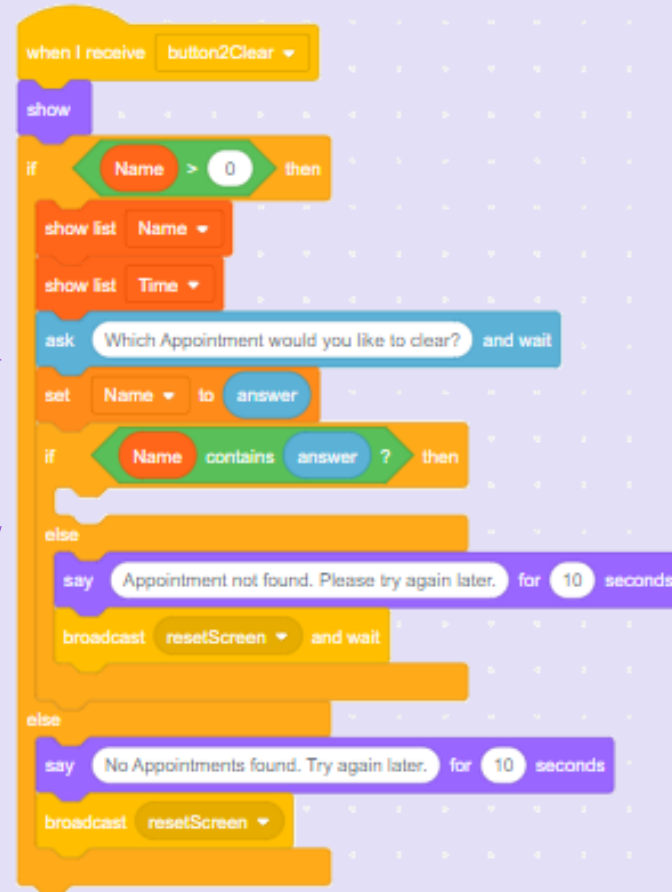


```
when I receive button2Clear
  show
  if Name > 0 then
    show list Name
    show list Time
  ask Which Appointment would you like to clear? and wait
  set Name to answer
else
  say No Appointments found. Try again later. for 10 seconds
  broadcast resetScreen
```

# Step 3

If the list 'Name' contains the same name as the one the user has input. If there is a match then the code within the second IF ELSE block will run.

If there is no match then a message saying to try again later will show. Then the screen will reset onto the home screen.



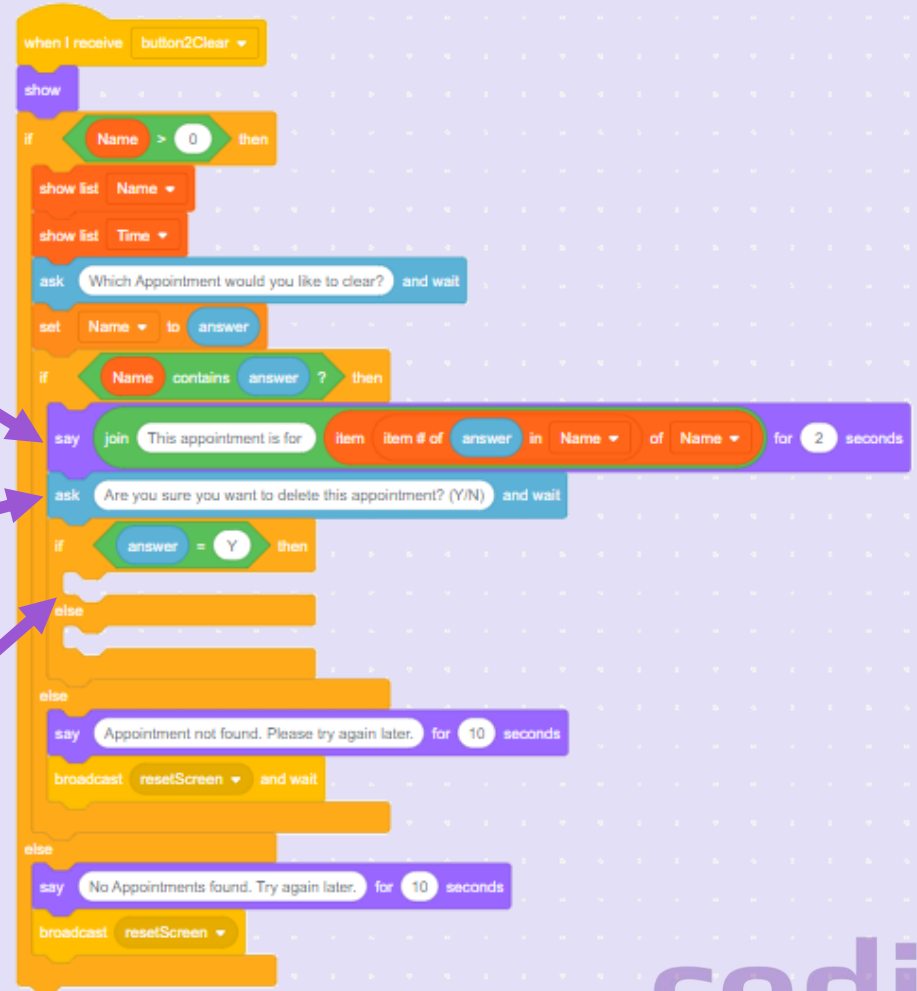
```
when I receive button2Clear
show
if Name > 0 then
  show list Name
  show list Time
  ask Which Appointment would you like to clear? and wait
  set Name to answer
  if Name contains answer ? then
  else
    say Appointment not found. Please try again later. for 10 seconds
    broadcast resetScreen and wait
  else
    say No Appointments found. Try again later. for 10 seconds
    broadcast resetScreen
```

# Step 4

This block displays a message confirming who the appointment is for.

The user is then asked if they are sure that they want to delete that appointment.

This IF ELSE block then tells us what we should do if the user inputs 'Y' or if they input 'N'.



```
when I receive button2Clear
show
if Name > 0 then
  show list Name
  show list Time
  ask Which Appointment would you like to clear? and wait
  set Name to answer
  if Name contains answer ? then
    say join This appointment is for item item # of answer in Name of Name for 2 seconds
    ask Are you sure you want to delete this appointment? (Y/N) and wait
    if answer = Y then
      // empty block
    else
      // empty block
    else
      say Appointment not found. Please try again later. for 10 seconds
      broadcast resetScreen and wait
    else
      say No Appointments found. Try again later. for 10 seconds
      broadcast resetScreen
```

# Step 5

If the user inputs 'Y' then the name and time of the appointment is deleted

A confirmation message then pops up whilst showing the new lists to the user.

If the user inputs 'N' then a message saying that the appointment wasn't deleted is displayed and then the screen resets.

```
if Name contains answer ? then
  say join This appointment is for item item # of answer in Name of Name for 2 seconds
  ask Are you sure you want to delete this appointment? (Y/N) and wait
  if answer = Y then
    delete item # of Name in Name of Time
    delete 1 of Name
    say Appointment deleted successfully! for 10 seconds
    broadcast item # of Name in Name
    broadcast resetScreen
  else
    say Appointment was not deleted. for 10 seconds
    broadcast resetScreen
```

# Congratulations!

You have completed the beginner section of the dentistry booking software.

Find the code for this lesson [here!](#)