Capstone Class: Voice Selfie App

What is our GOAL for this MODULE?

We learned how to process a text to trigger an action. We completed building our voice controlled selfie app.

What did we ACHIEVE in the class TODAY?

- Used an **if** condition to check whether the spoken words match our criteria.
- Built the voice controlled selfie app.

Which CONCEPTS/ CODING did we cover today?

- Added an **if** condition to check whether "**take my selfie**" was said.
- Wrote **take_snapshot()** function.
- Delayed the **take_snapshot()** function by 5 seconds.
- Wrote the **save()** function.

How did we DO the activities?

Divide the code in parts:

Part 1 -

- Add an **if** condition to check if the user says **"take my selfie"**.

Part 2-

- If we have said "**take my selfie**", then it should not repeat it, instead it should say **"Taking selfie in 5 seconds"**.

Part 3 -

- Writing code to take a selfie.

Part 4 -

- Learning and understanding how to delay some code.
- Delay the take selfie code (5 seconds) using the timer function.

Part 5 -

- Writing code for downloading the selfie file.

Part 1-

Our first aim is to put an **if** condition that checks if what we have said is **“take my selfie”**.

Continue the code in the **main.js** file.

First, add code for checking if what we have said is “take my selfie”. Put this if condition inside the **onresult** function which we did in class 98. The **onresult** function is used to convert speech to text, so it will be the right place to add the code for checking if what we have said is “take my selfie”. Add code inside **onresult** function like this:

```
var SpeechRecognition = window.webkitSpeechRecognition;

var recognition = new SpeechRecognition();

function start()
{
    document.getElementById("textbox").innerHTML = "";
    recognition.start();
}

recognition.onresult = function(event) {

 console.log(event);

var Content = event.results[0][0].transcript;

    document.getElementById("textbox").innerHTML = Content;
    console.log(Content);
      if(Content =="take my selfie")
      {
        console.log("taking selfie --- ");
        speak();
      }
}
```

Explaining the above code:

1. If you remember, the **content** variable holds the text that is converted from our

speech.

2. So now we know the **content** variable holds the text of our speech, so we can easily do a condition check that if the **content** variable has "**take my selfie**", then console a text saying anything like "**taking selfie**", this console is done just to verify that if condition works.

```
if(Content =="take my selfie")
{
  console.log("taking selfie --- ");
```

- Output on console screen:

```
take my selfie                                                    main.js:18
```

3. Now move the **speak()** function inside the if condition.

```
if(Content =="take my selfie")
{
  console.log("taking selfie --- ");
  speak();
}
```

- The purpose of doing this is, as we saw in the output, the speaking from the system was only done when we said "**take my selfie**".

Now if you will run the code, and say something, nothing will happen other than the speech getting converted into text.
And when you say, "**take my selfie**" at that time, the system will repeat the same and open the webcam live view.

Part 2 -

In this part change where the system is repeating the voice command we gave, to speak "**Taking selfie in 5 seconds**".

Change the **speak()** function which we did in the last class.

Previous class code - **speak()** function:

```
function speak(){
    var synth = window.speechSynthesis;

    speak_data = document.getElementById("textbox").value;

    var utterThis = new SpeechSynthesisUtterance(speak_data);

    synth.speak(utterThis);
    Webcam.attach(camera);
}
```

If you remember, the **speak_data** variable used to hold the text taken from the textarea (this textarea has the speech to text converted value).
And then we passed the **speak_data** variable to get it converted back to speech, and then we triggered the system to speak it out.

1. Replace the above marked code with the below code:

```
speak_data = "Taking you Selfie in 5 seconds";
```

- So **speak()** function will look like this:

```
function speak(){
    var synth = window.speechSynthesis;

    speak_data = "Taking you Selfie in 5 seconds";

    var utterThis = new SpeechSynthesisUtterance(speak_data);

    synth.speak(utterThis);
    Webcam.attach(camera);
}
```

- So now the **speak_data** variable will always have the text as "**Taking selfie in 5 seconds**". So when the speak() function is called it will say **Taking selfie in 5 seconds.**

It is not compulsory to mention 5 seconds specifically, you can mention whatever seconds you feel like.

Now if you will run the code, and say anything, nothing will happen other than the speech getting converted into text.
And when you say "**take my selfie**" at that time, the system will say "Taking selfie in 5 seconds", and will open the webcam live view.

Part 3 -
Write code for taking a selfie. Add this code after the webcam code, like this:

```
Webcam.set({
    width:360,
    height:250,
    image_format : 'png',
    png_quality:90
});
camera = document.getElementById("camera");

function take_snapshot()
{
    Webcam.snap(function(data_uri) {
        document.getElementById("result").innerHTML = '<img id="selfie_image" src="'+data_uri+'">';
    });
}
```

Zoom in look:

```
function take_snapshot()
{
    Webcam.snap(function(data_uri) {
        document.getElementById("result").innerHTML = '<img id="selfie_image" src="'+data_uri+'">';
    });
}
```

Explaining the above code:

1. Define the function.

```
function take_snapshot()
{
```

2. Webcam.snap() is a predefined function of **webcam.js** which is used to take a selfie,

this function contains data_uri that can be used to show preview of the image which generates after taking a snapshot. So first define Webcam.snap():

```
function take_snapshot()
{
    Webcam.snap( );
```

3. Now write a function inside Webcam.snap(), and pass data_uri inside it. And use this data_uri to display the selfie.

```
function take_snapshot()
{
    Webcam.snap( function(data_uri) { } );
```

4. Now update the div which we had made for the purpose of holding the selfie in **voice_selfie_app.html**, with this data_uri which has the selfie taken:

```
Webcam.snap(function(data_uri) {
    document.getElementById("result").innerHTML = '<img id="selfie_image" src="'+data_uri+'">';
});
```

- `document.getElementById("result").innerHTML =` this we already know.
- Define **img** tag: `= '<img`
    - Then we give an **id** to this **img** tag `= '<img id="selfie_image"` . We are giving **id** to **img** tag, so that later we can take this image from this **img** tag and save it.
- Now in **src** of the img tag pass data_uri. So that this image gets updated with the selfie taken and gets displayed.

```
= '<img id="selfie_image" src="'+data_uri+'">';
```

Be careful while closing and opening double quotes and single quotes.

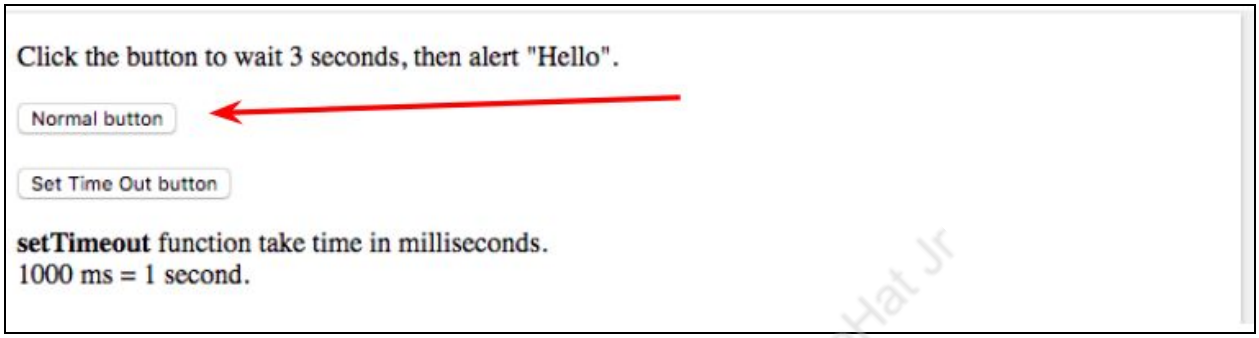
Part 4 -
To learn how to delay some part of the code use the **set timeout** function of javascript.

Run https://www.w3schools.com/code/tryit.asp?filename=GDME156RVQSU, and follow the

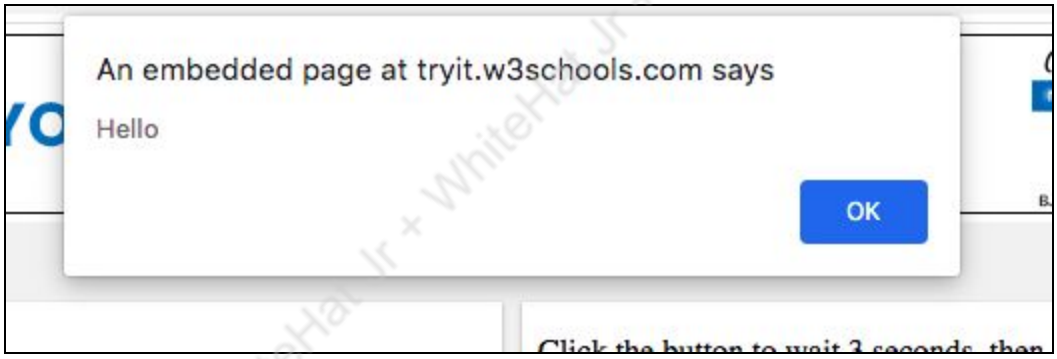
steps:

1. Press the RUN button: 

2. Now click the “Normal button”:



3. And you will notice as soon as you press the normal button, an alert box will come:



- Code for “Normal button”:

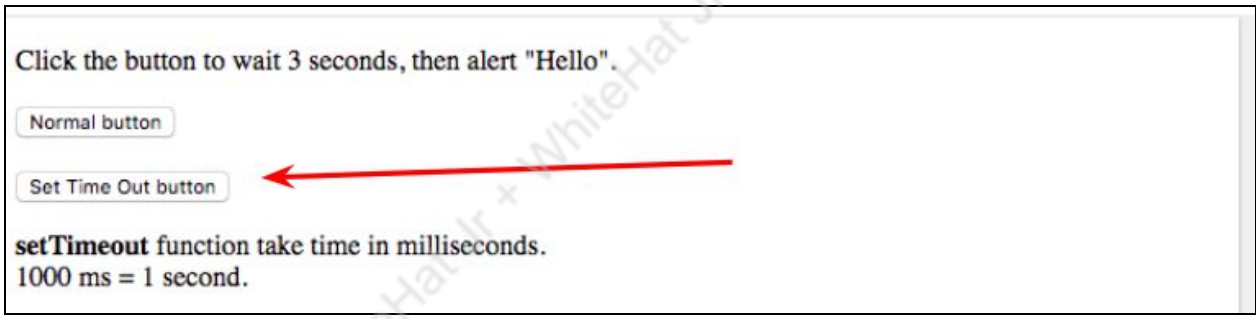
```
<!DOCTYPE html>
<html>
<body>

<p>Click the button to wait 3 seconds, then alert "Hello".</p>

<button onclick="normalButton()">Normal button</button>

<br>
<br>

<button onclick="setTimeOutButton()">Set Time Out button</button>

<script>
function normalButton() {
        alert("normal Button");
}
```

Defining the button

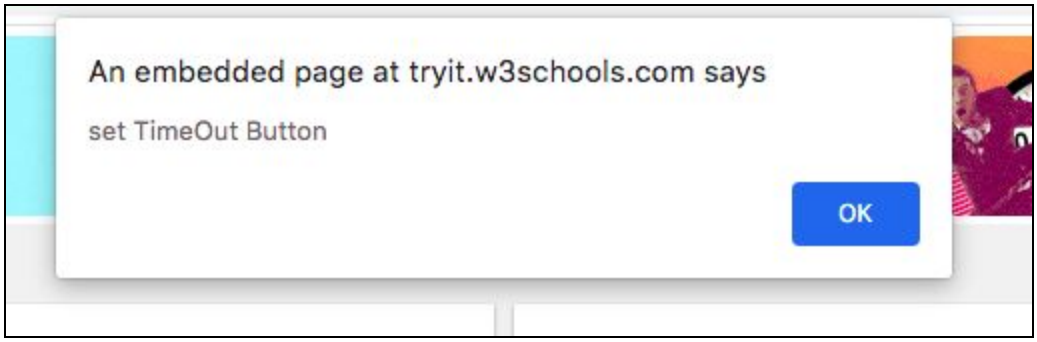
Defining the function

- So as you see the code it's very basic and normal code, which we already know how to define a button and give an onclick function to it. And then how to define that onclick function

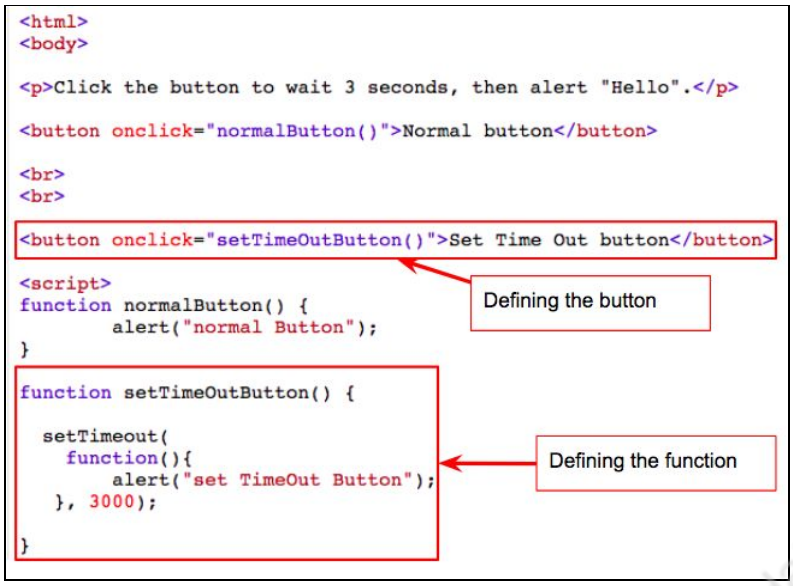
4. Now click the "Set TimeOut Button":



5. Now after 3 seconds an alert box will come:



Now see the code for "setTimeOutButton"



We already know how to write code for a HTML button and give it an onclick function.

Now see the **setTimeOutButton** function:

1. Define the function:

```
function setTimeOutButton() {
```

2. Now call the **setTimeout** function.

```
function setTimeOutButton() {
  setTimeout();
```

3. Now define an empty function inside the **setTimeout** function.

```
function setTimeOutButton() {
  setTimeout( function(){} );
```

4. Now give the delayed time you want to set, it accepts time in milliseconds, and 1 second is equal to 1000 milliseconds, so **if you want to delay code for:**
   - **1 seconds** taken you will write **1000 milliseconds**
   - **2 seconds** taken you will write **1000 milliseconds**
   - **3 seconds** taken you will write **3000 milliseconds**
   - **4 seconds** taken you will write **4000 milliseconds**

- and so on **.....**
    - Since we want to give a delay of 3 seconds, write 3000 after we define a function like this:

```
function setTimeOutButton() {

  setTimeout( function(){} , 3000);
```

5. Now write the code which we want to be delayed, inside the empty function:

```
function setTimeOutButton() {

  setTimeout( function(){

      alert("set TimeOut Button");

  } , 3000);
```

- So we have written a code for alerting, so as a result this alert box will be delayed.

Now implement this set timeout function in our code.

As you saw in the output at the start of today's class, after we say "**take my selfie**" the system says "**taking selfie in 5 seconds**" and the webcam starts and after 5 seconds the selfie is taken.

Now call the **take_snapshot()** function [which is a function to take a selfie] inside a **set Timeout function** because we want the process of taking a selfie to be delayed by 5 seconds.
And put this **set Timeout function** inside the **speak()** function.

We are putting this **set Timeout function** inside the **speak()** function because the trigger for starting a webcam is also there in **speak()** function. Like this:

```
function speak(){
    var synth = window.speechSynthesis;

    speak_data = "Taking you Selfie in 5 seconds";

    var utterThis = new SpeechSynthesisUtterance(speak_data);

    synth.speak(utterThis);
    Webcam.attach(camera);

    setTimeout(function()
    {
        take_snapshot();
    }, 5000);
}
```

Explaining the above code:

1. Define **set Timeout function** inside **speak()** function, like this:

```
function speak(){
    var synth = window.speechSynthesis;

    speak_data = "Taking you Selfie in 5 seconds";

    var utterThis = new SpeechSynthesisUtterance(speak_data);

    synth.speak(utterThis);
    setTimeout();

    Webcam.attach(camera);
}
```

2. Then define an empty function inside the **set Timeout.**

```
function speak(){
    var synth = window.speechSynthesis;

    speak_data = "Taking you Selfie in 5 seconds";

    var utterThis = new SpeechSynthesisUtterance(speak_data);

    synth.speak(utterThis);
    setTimeout( function() {} );

    Webcam.attach(camera);
}
```

3. Now give the delayed time you want to set. Here we are setting to 5 seconds you can set whatever you want.

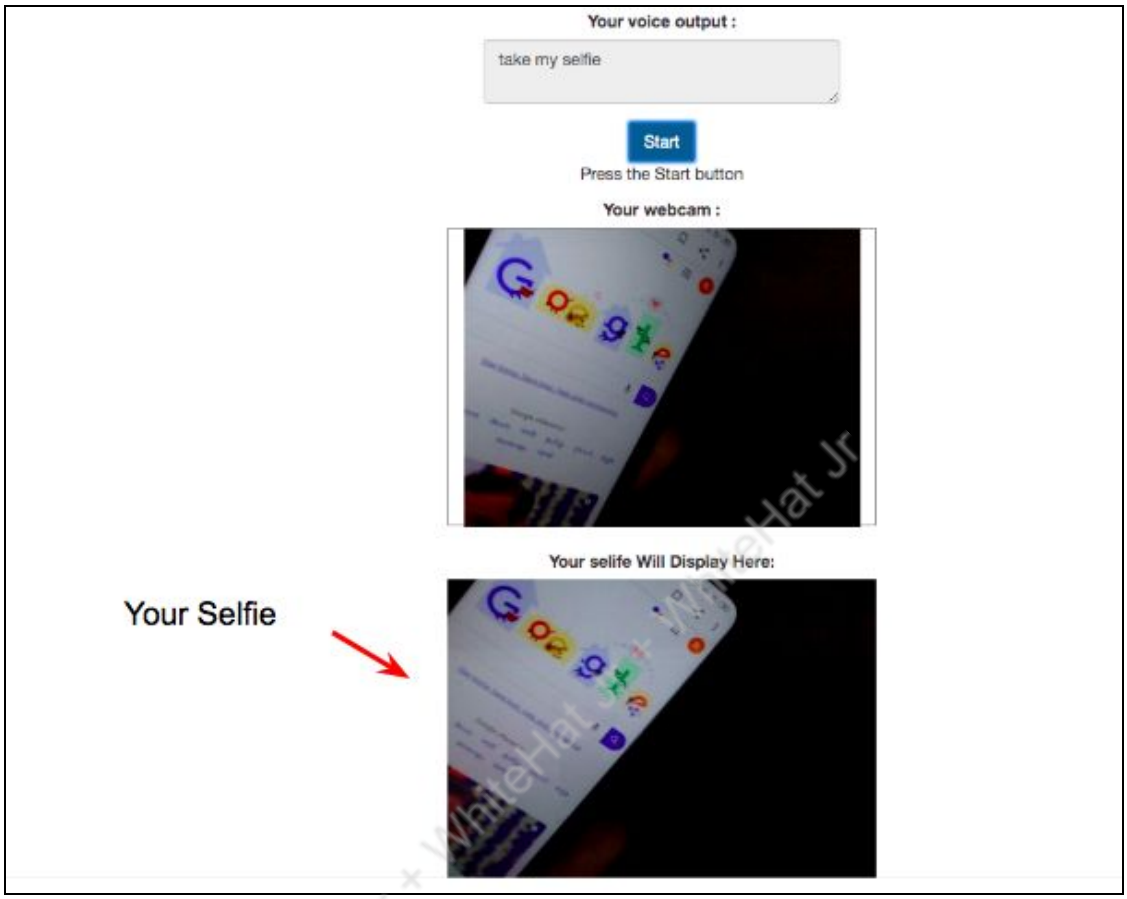
```
function speak(){
    var synth = window.speechSynthesis;

    speak_data = "Taking you Selfie in 5 seconds";

    var utterThis = new SpeechSynthesisUtterance(speak_data);

    synth.speak(utterThis);
    setTimeout( function() {} , 5000 );

    Webcam.attach(camera);
}
```

4. Now call **take_snapshot()** function inside the **set Timeout function** because we want taking selfies should get delayed by 5 seconds.

```
function speak(){
    var synth = window.speechSynthesis;

    speak_data = "Taking you Selfie in 5 seconds";

    var utterThis = new SpeechSynthesisUtterance(speak_data);

    synth.speak(utterThis);
    Webcam.attach(camera);

    setTimeout(function()
    {
        take_snapshot();
    }, 5000);

}
```


5. Now if you will run the code, and say anything, nothing will happen except the speech getting converted into text. And when the you will say "**take my selfie**" at that time, the system will say "**Taking selfie in 5 seconds**", and open the webcam live

view, and a selfie will be taken and shown below the webcam live view, like this:



Part 5 -
Now write code for saving the selfie which has been taken.
Write the code for this after the **take_snapshot()** function like, this -

```
function take_snapshot()
{
    Webcam.snap(function(data_uri) {
        document.getElementById("result").innerHTML = '<img id="selfie_image" src="'+data_uri+'">';
    });
}
function save()
{
  link = document.getElementById("link");
  image = document.getElementById("selfie_image").src ;
  link.href = image;
  link.click();
}
```

Zoom in look for save function:

```
function save()
{
  link = document.getElementById("link");
  image = document.getElementById("selfie_image").src ;
  link.href = image;
  link.click();
}
```

Explaining the above code:

1. First define the function `function save() { }`
2. Now store the anchor tag(which we created in **voice_selfie_app.html** for the purpose of downloading the selfie) inside a variable. The purpose of storing the element inside a variable is, we can easily refer to the anchor tag.

```
link = document.getElementById("link");
```

- The **src** holds the image link in **img** tag. So **src** is the important part of **img** tag.
- Same way **href** holds the link in the **anchor** tag. So **link** is the important part of the **anchor** tag.
- So take the image link from **src** of **img tag**, update the **href** of the **anchor** tag with this image link. And **click** the **anchor** tag.

3. Now get the image from the img tag where we have stored the selfie, and store it inside a variable.

```
image = document.getElementById("selfie_image").src ;
```

- `image =` is the variable in which we will store the image.
- `document.getElementById("selfie_image")` This we already know referring to the img tag using it's id.
- The way we write **document.getElementBy("any id").value**
  - This gets the value from the selected element.
  - The same way write **document.getElementBy("any id").src**
  - This gets the src from the selected element.

4. Now update the anchor tag **href**, with this image variable (this has the link of the image).

```
link.href = image;
```

- `link` - this is the variable that holds the anchor tag
- `.href` - we want to update **href**, that's why href
- We want to update **href** with the image link, and `image;` variable holds the image link, that's y - `= image;`

5. So the following code will automatically click the anchor tag:

```
link.click();
```

- **The purpose of writing a code for automatically clicking the anchor tag is we want the image should get downloaded automatically, and we don't have to click any button.**

6. Now call the **save()** function after the **take_snapshot()** function, because as soon as a selfie is taken we want to save it.

```
function speak(){
    var synth = window.speechSynthesis;

    speak_data = "Taking you Selfie in 5 seconds";

    var utterThis = new SpeechSynthesisUtterance(speak_data);

    synth.speak(utterThis);
    Webcam.attach(camera);

    setTimeout(function()
    {
        take_snapshot();
        save();
    }, 5000);

}
```


What's NEXT?

In the next class we will learn to make a 3D model of chandrayaan 2.