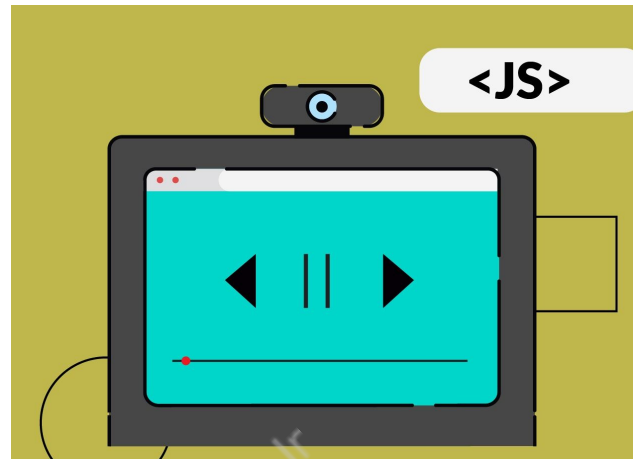


PRETRAINED NET MODELS



What is our GOAL for this MODULE?

We learned how to manipulate the volume and the speed of the song by the movements of our body parts.

What did we ACHIEVE in the class TODAY?

- We continued the JS code and added a few more functionalities.

Which CONCEPTS/ CODING did we cover today?

- Added code for drawing a red circle on x and y coordinates of leftWrist.
- Added code for doing calculations on y coordinate of leftWrist and passed the calculated value to setVolume() function.
- Added code for fetching score of leftWrist.
- Added an "if" statement to check that if the leftWrist is in front of the webcam, if yes then only draw a red circle and change the volume of the song.

How did we DO the activities?

1. First add the code for drawing a circle on both x and y coordinates of leftWrist.

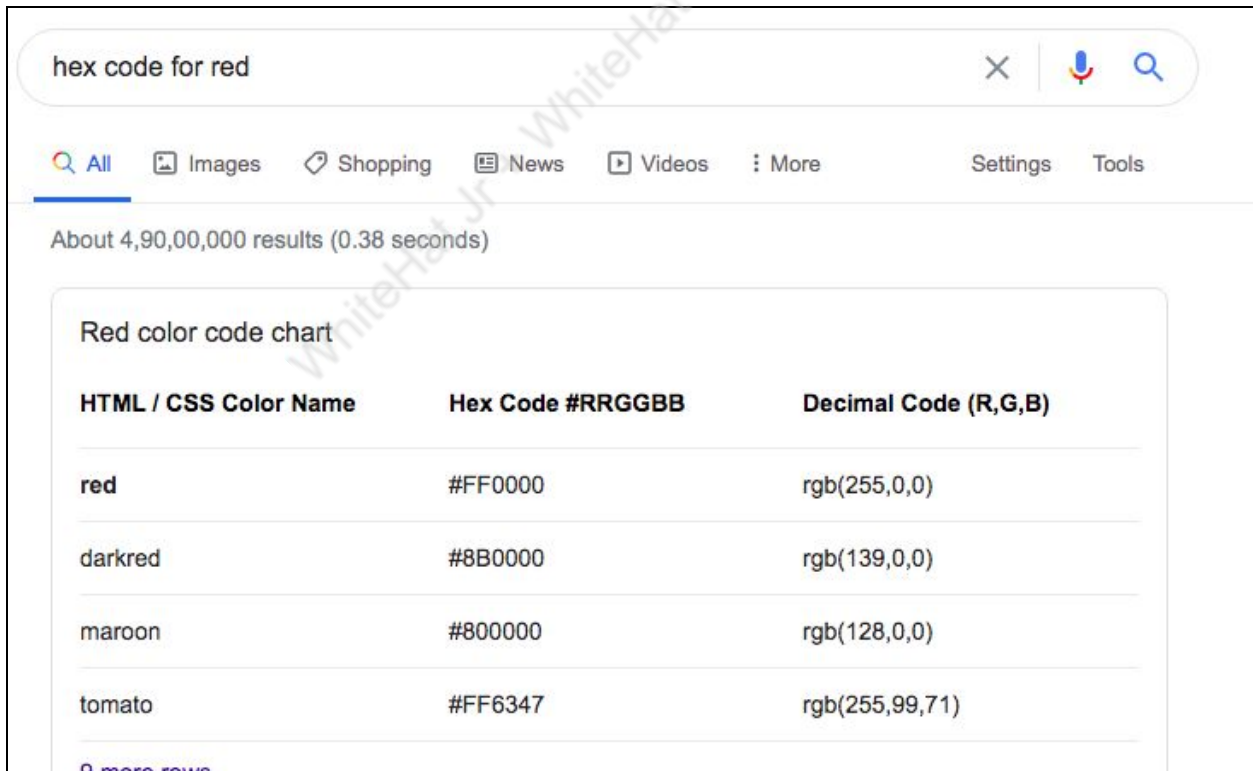
Setting the color for the circle.

2. Inside **fill()** function you can pass either, RGB code, HEX code, RGBA code. So we have used a HEX code for setting the color for the circle.

```
function draw() {
  image(video, 0, 0, 600, 500);

  fill("#FF0000");
}
```

- **"#FF0000"** - this is a HEX code of red color. It is not compulsory to use this color, you can use any color as per your choice.
- To get the HEX code of any color just google - "HEX code for **color name**" for eg:



hex code for red

About 4,90,00,000 results (0.38 seconds)

Red color code chart

HTML / CSS Color Name	Hex Code #RRGGBB	Decimal Code (R,G,B)
red	#FF0000	rgb(255,0,0)
darkred	#8B0000	rgb(139,0,0)
maroon	#800000	rgb(128,0,0)
tomato	#FF6347	rgb(255,99,71)

[9 more rows](#)

Now set the border color to the circle.

3. Inside **stroke()** function you can pass either, RGB code, HEX code, RGBA code. So we have used a HEX code for setting border color for a circle.

```
function draw() {  
  image(video, 0, 0, 600, 500);  
  
  fill("#FF0000");  
  stroke("#FF0000");  
}
```

- **"#FF0000"** - this is a HEX code of red color. it is not compulsory to use this color, you can use any color as per your choice.

Now write a code for drawing a circle.

circle() is a p5.js function used to draw a circle.

Syntax of **circle()**:

circle(x coordinate, y coordinate, radius);

4. So put both **x and y coordinates of the leftWrist**, in place of **x coordinate, y coordinate** of the **circle()**, which will result in moving the circle as per the moment of the leftWrist. We want a small circle, thereby giving a compact figure to the circle, give a small value to the **radius** for the circle.

```
function draw() {  
  image(video, 0, 0, 600, 500);  
  
  fill("#FF0000");  
  stroke("#FF0000");  
  
  circle(leftWristX, leftWristY, 20);  
}
```

- This will result in drawing a mini red circle on leftWrist.

NOTE - DO A TEST BY CLICKING ON GO LIVE BUTTON OF VISUAL STUDIO. THIS WILL RESULT IN RUNNING THE FILE ON THE LIVE SERVER



OF VISUAL

STUDIO.

Because we are using a sound file, and p5.js just doesn't allow us to run any sound file from a local system, therefore it needs to be run on a server.

Before we start the math part let's consider y coordinate of leftWrist as 400.345678765 and use this number as an example for doing the math.

5. Now convert the value of the leftWrist variable into a number and store it inside a variable. We will be doing this because if the leftWrist value is in a string type and if we do math using the string value of leftWrist variable, then we won't get a proper output, so to be on a safer side convert this variable into a number.

For converting the leftWrist variable into a number use JS predefined function **number()**.

Syntax of **number()** -

number(variable) - this will convert a string type variable into of the number type variable.

```
function draw() {
  image(video, 0, 0, 600, 500);

  fill("#FF0000");
  stroke("#FF0000");

  circle(leftWristX, leftWristY, 20);
  InNumberleftWristY = Number(leftWristY);
}
```

- We considered using this - **400.345678765** number as an example, so after applying the **Number()** function to the leftWrist variable we get a value **400.345678765** of type number and store this value in the **InNumberleftWristY** variable.
6. We know that the y coordinate of leftWrist has lots of decimals:

```
▶ leftWrist: {x: 263.52472737518667, y: 400.345678765}
```

- So when we will do some calculation on y coordinate of leftWrist we will get a result which will have a lot of decimals.
 - We don't want a lot of decimals after calculation. So we need to remove all the decimals from the y coordinate of leftWrist.
7. Put **InNumberleftWristY** variable inside **floor()** function, and assign all this to a new variable:

```
function draw() {  
  image(video, 0, 0, 600, 500);  
  
  fill("#FF0000");  
  stroke("#FF0000");  
  
  circle(leftWristX, leftWristY, 20);  
  InNumberleftWristY = Number(leftWristY);  
  remove_decimals = floor(InNumberleftWristY);  
}
```

- As we considered to use this - **400.345678765** number as an example, so after applying **floor()** to this number, the value will be **400**, and store this value in the **remove_decimals** variable.
8. Now divide the **remove_decimals** variable by 500, because **setVolume()** function takes values between 0 and 1, so dividing y coordinates of leftWrist by 500 will help us to get a value between 0 and 1. After dividing the **remove_decimals** variable by 500 store it inside a new variable.

Examples:

- If the y coordinate of the leftWrist is **100**, so after dividing it by 500 we will get a value **0.2**, so we can pass 0.2 to **setvolume()** function. And as a result a low volume will be set for the song.
- If the y coordinate of the leftWrist is **200**, so after dividing it by 500 we will get a value **0.4**, so we can pass 0.4 to **setvolume()** function. And as a result a little low volume will be set for the song.
- If the y coordinate of the leftWrist is **300**, so after dividing it by 500 we will get a value **0.6**, so we can pass 0.6 to **setvolume()** function. And as a result a little high volume will be set for the song.
- If the y coordinate of the leftWrist is **400**, so after dividing it by 500 we will get a value **0.8**, so we can pass 0.8 to **setvolume()** function. And as a result a low volume will be set for the song.

- If the y coordinate of the leftWrist is **500**, so after dividing it by 500 we will get a value **1**, so we can pass 1 to **setvolume()** function. And as a result a full volume will be set for the song.

```
function draw() {  
  image(video, 0, 0, 600, 500);  
  
  fill("#FF0000");  
  stroke("#FF0000");  
  
  circle(leftWristX, leftWristY, 20);  
  InNumberleftWristY = Number(leftWristY);  
  remove_decimals = floor(InNumberleftWristY);  
  volume = remove_decimals/500;  
}
```

- As we considered to use this - **400.345678765** number as an example. And after applying **floor()** we got a value as **400** and we had stored this value in the **remove_decimals** variable. Now divide the **remove_decimals** variable with 500, and this will get a value as **0.8** and store this value in a **volume** variable.
9. Remember in class no.126 we had defined a h3 tag and gave an id as "**volume**". And the purpose of this h3 tag was to hold the changing volume. So update this h3 tag using it's id "**volume**" with **volume** variable (as this variable is holding the changing sound).

```
function draw() {  
  image(video, 0, 0, 600, 500);  
  
  fill("#FF0000");  
  stroke("#FF0000");  
  
  circle(leftWristX, leftWristY, 20);  
  InNumberleftWristY = Number(leftWristY);  
  remove_decimals = floor(InNumberleftWristY);  
  volume = remove_decimals/500;  
  document.getElementById("volume").innerHTML = "Volume = " + volume;  
}
```


10. Now pass this **volume** variable (as this variable is holding the changing sound) to **setVolume()** function. This will result in changing the volume of the song as per the movement of leftWrist.

```
function draw() {  
  image(video, 0, 0, 600, 500);  
  
  fill("#FF0000");  
  stroke("#FF0000");  
  
  circle(leftWristX, leftWristY, 20);  
  InNumberleftWristY = Number(leftWristY);  
  remove_decimals = floor(InNumberleftWristY);  
  volume = remove_decimals/500;  
  document.getElementById("volume").innerHTML = "Volume = " + volume;  
  song.setVolume(volume);  
}
```

Now add code for checking if leftWrist is there in front of the webcam or not.

Not only does the posenet return the x and y coordinates of 17 body parts but also does return a score to each body part, respectively.

Score meaning the confidence that a body part has been detected OR a body part is in front of the webcam. So whenever a score is greater than **0.2**, it means the body part has been detected OR a body part is in front of the webcam.

1. So let's create a variable to store the score of leftWrist at the beginning of the **main.js** file, and let's set its value as 0.

```
song = "";  
scoreLeftWrist = 0;  
  
rightWristX = 0;  
rightWristY = 0;  
  
leftWristX = 0;  
leftWristY = 0;
```

- Run <https://mahdihat791.github.io/Ai-DJ/> and open console screen:

```
PoseNet Is Initialized      main.js:17
▶ [{-}]                    main.js:24
```

We need to read this object and fetch the score of the leftWrist, while reading the object also write the code:

- First click on the arrow to expand:

```
PoseNet Is Initialized      main.js:17
▶ [{-}]                    main.js:24
```

- We want to read the objects of **results** so first write

results

- Then click on the arrow next to **0** to expand:

```
▼ [{-}] 1                  main.js:24
▶ 0: {pose: {_-}, skeleton: Array(0)}
  length: 1
  ▶ __proto__: Array(0)
```

- We have clicked on 0 index which is inside the "results" object, so code will be

results[0]

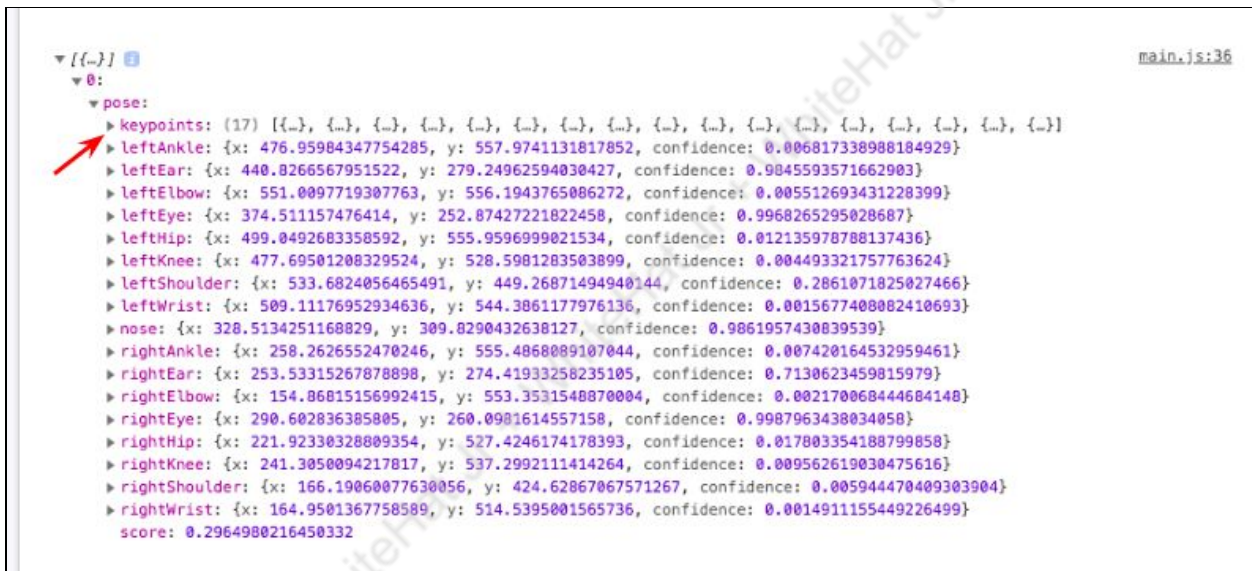
- Then click on the arrow next to **pose** to expand:

```
▼ [{-}] 1                  main.js:24
▼ 0:
  ▶ pose: {score: 0.25857010390866303, keypoints: Array(1_
  ▶ skeleton: []
  ▶ __proto__: Object
  length: 1
  ▶ __proto__: Array(0)
```


- Then inside 0 index we have clicked on pose object, so code will be:

```
results[0].pose
```

6. Then inside the **pose** object there are the two important parts **keypoints:** 17 body parts with x and y coordinates and score of each body part. This time expand **keypoints:** as it has the same thing which is 17 body parts with x and y coordinates. But also it contains the score for each body part. So click the arrow new to **keypoints:** to expand:



```

  ▼ [{"-"}]
    ▼ 0:
      ▼ pose:
        ▶ keypoints: (17) [{"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}]
        ▶ leftAnkle: {x: 476.95984347754285, y: 557.9741131817852, confidence: 0.006817338988184929}
        ▶ leftEar: {x: 440.8266567951522, y: 279.24962594030427, confidence: 0.9845593571662903}
        ▶ leftElbow: {x: 551.0097719307763, y: 556.1943765086272, confidence: 0.005512693431228399}
        ▶ leftEye: {x: 374.511157476414, y: 252.87427221822458, confidence: 0.9968265295028687}
        ▶ leftHip: {x: 499.0492683358592, y: 555.9596999021534, confidence: 0.012135978788137436}
        ▶ leftKnee: {x: 477.69501208329524, y: 528.5981283503899, confidence: 0.004493321757763624}
        ▶ leftShoulder: {x: 533.6824056465491, y: 449.26871494940144, confidence: 0.2861071825027466}
        ▶ leftWrist: {x: 509.11176952934636, y: 544.3861177976136, confidence: 0.0015677408082410693}
        ▶ nose: {x: 328.5134251168829, y: 309.8290432638127, confidence: 0.9861957430839539}
        ▶ rightAnkle: {x: 258.2626552470246, y: 555.4868089107044, confidence: 0.007420164532959461}
        ▶ rightEar: {x: 253.53315267878898, y: 274.41933258235105, confidence: 0.7130623459815979}
        ▶ rightElbow: {x: 154.06815156992415, y: 553.3531540870004, confidence: 0.002170068444604148}
        ▶ rightEye: {x: 290.602836385805, y: 260.0901614557158, confidence: 0.9987963438034058}
        ▶ rightHip: {x: 221.92330328809354, y: 527.4246174178393, confidence: 0.017803354188799858}
        ▶ rightKnee: {x: 241.3050094217817, y: 537.2992111414264, confidence: 0.009562619038475616}
        ▶ rightShoulder: {x: 166.19060077630056, y: 424.62867067571267, confidence: 0.005944470409303904}
        ▶ rightWrist: {x: 164.9501367758589, y: 514.5395001565736, confidence: 0.0014911155449226499}
        score: 0.2964980216450332
  
```

- If we want to get the score of the leftWrist, so inside results -> inside 0 index ->

inside pose -> inside keypoints. So write - **results[0].pose.keypoints**

Inside **keypoints:** there is:

- Index number of the body parts
- Score for the body parts
- Name of the body parts
- And position of the body parts - which is nothing but x and y coordinates of the body parts



The screenshot shows a JSON object with a 'pose' key containing an array of 'keypoints'. Annotations include:

- Score of the body parts**: Points to the 'score' field in each keypoint object.
- Name of the body parts**: Points to the 'part' field in each keypoint object.
- position of the body parts which is nothing but the x and y coordinates**: Points to the 'position' field in each keypoint object.
- Index number of the body parts**: Points to the index of the keypoint array (0 to 16).

Index	Score	Part	Position
0	0.9861957430839539	nose	{...}
1	0.9968265295028687	leftEye	{...}
2	0.9987963438034058	rightEye	{...}
3	0.9845593571662903	leftEar	{...}
4	0.7130623459815979	rightEar	{...}
5	0.2861071825027466	leftShoulder	{...}
6	0.005944470409303904	rightShoulder	{...}
7	0.005512693431228399	leftElbow	{...}
8	0.002170068444684140	rightElbow	{...}
9	0.0015677408082410693	leftWrist	{...}
10	0.0014911155449226499	rightWrist	{...}
11	0.012135978788137436	leftHip	{...}
12	0.017803354188799838	rightHip	{...}
13	0.004493321757763624	leftKnee	{...}
14	0.009562619030475616	rightKnee	{...}
15	0.006817338988184929	leftAnkle	{...}
16	0.007420164532959461	rightAnkle	{...}

- To get the score of any body parts we need to refer to its index. And this index number will always remain constant. Meaning:
 - The index number of the nose is 0 so to refer to the nose score we can always use 0 as an index number.
 - The index number of the rightKnee is 16 so to refer to the rightKnee score we can always use 16 as an index number.

- Now we want to get the score of leftWrist. The index of leftWrist is 9, so **inside results -> inside 0 index -> inside pose -> inside keypoints -> inside 9**. So the code

will be - `results[0].pose.keypoints[9]`

- Now we have the index number of leftWrist, so we can get the score inside of leftWrist, so **inside results -> inside 0 index -> inside pose -> inside keypoints -> inside 9 -> there is score**. So the code will be -

`results[0].pose.keypoints[9].score`

9. We have fetched the code for getting the score of the leftWrist, so update the **scoreLeftWrist** variable with this code, inside the **gotPoses()** function.

```
function gotPoses(results)
{
  if(results.length > 0)
  {
    console.log(results);
    scoreLeftWrist = results[0].pose.keypoints[9].score;

    rightWristX = results[0].pose.rightWrist.x;
    rightWristY = results[0].pose.rightWrist.y;
    console.log("rightWristX = " + rightWristX + " rightWristY = " + rightWristY);

    leftWristX = results[0].pose.leftWrist.x;
    leftWristY = results[0].pose.leftWrist.y;
    console.log("leftWristX = " + leftWristX + " leftWristY = " + leftWristY);
  }
}
```

10. Then console this variable.

```
function gotPoses(results)
{
  if(results.length > 0)
  {
    console.log(results);
    scoreLeftWrist = results[0].pose.keypoints[9].score;
    console.log("scoreLeftWrist = " + scoreLeftWrist);

    rightWristX = results[0].pose.rightWrist.x;
    rightWristY = results[0].pose.rightWrist.y;
    console.log("rightWristX = " + rightWristX + " rightWristY = " + rightWristY);

    leftWristX = results[0].pose.leftWrist.x;
    leftWristY = results[0].pose.leftWrist.y;
    console.log("leftWristX = " + leftWristX + " leftWristY = " + leftWristY);
  }
}
```

- We have the score of leftWrist inside the **scoreLeftWrist** variable.

Now we can use the **scoreLeftWrist** for checking that the leftWrist has been detected or is it in front of the webcam, if yes then only draw a red circle on x and y coordinates of leftWrist and do the calculation and change the volume of the song.

So for this we can add an “**if condition**” to check that the **scoreLeftWrist** is greater than **0.2** and put the code of the drawing circle and change the volume of the song inside this “**if condition**”.

```
function draw() {  
  image(video, 0, 0, 600, 500);  
  
  fill("#FF0000");  
  stroke("#FF0000");  
  
  if(scoreLeftWrist > 0.2)  
  {  
    circle(leftWristX, leftWristY, 20);  
    InNumberleftWristY = Number(leftWristY);  
    remove_decimals = floor(InNumberleftWristY);  
    volume = remove_decimals/500;  
    document.getElementById("volume").innerHTML = "Volume = " + volume;  
    song.setVolume(volume);  
  }  
}
```

What's NEXT?

It's almost time for the Capstone class. And by now, you are well-versed with this fun and challenging project.

In the capstone class, we will explore how to use our wrist to control the sound and speed of the song in our AI DJ App.