

- En la última clase teníamos esta gran cantidad de código:

```
song = "";

function preload()
{
  song = loadSound("music.mp3");
}

function setup() {
  canvas = createCanvas(600, 500);
  canvas.center();

  video = createCapture(VIDEO);
  video.hide();
}

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

function play()
{
  song.play();
}
```

- **Agregar código para configurar el volumen de la canción**  
Se usa **setVolume()** función (función predefinida) de p5.js que se usaría para configurar el volumen de la canción.  
Sintaxis de **setVolume(): variableName.setVolume(volume value):**  
**variableName** - Esta variable contendrá el archivo de audio.  
**setVolume** - este es el nombre de función predefinido de p5.js  
**volume value** - significa a qué volumen quieres reproducir la canción  
esto toma un valor entre 0.0 (silencio) y 1.0 (volumen completo):
  - Recurso **variableName.setVolume(0.1)** - Muy bajo
  - Recurso **variableName.setVolume(0.3)** - poco bajo
  - Recurso **variableName.setVolume(0.5)** - Medio
  - Recurso **variableName.setVolume(0.7)** - poco alto
  - Recurso **variableName.setVolume(0.9)** - Alto
  - Recurso **variableName.setVolume(1)** - Volumen completo

```
function play()
{
  song.play();
  song.setVolume(1);
}
```

- **Agregar código para establecer la velocidad de la canción.**

Para esto usamos la función `rate()`, que es una función predefinida de p5.js para controlar la velocidad. Sintaxis de `rate(): variableName.rate(playback rate):`

**variableName** - Esta variable contendrá el archivo de audio.

**rate** - Este es el nombre de función predefinido de p5.js

**playback rate** - significa a qué velocidad quieres reproducir la canción.

Establece la velocidad de reproducción. 1.0 es normal, .5 es la mitad de la velocidad, 2.0 es el doble de rápido.

- Recurso `variableName.rate(0.5)` - Muy lento
- Recurso `variableName.rate(1)` - Normal
- Recurso `variableName.rate(1.5)` - poco rápido
- Recurso `variableName.rate(2)` - El doble de rápido
- Recurso `variableName.rate(2.5)` - Muy rápido

Entonces, este código establecerá la velocidad como normal, y en la próxima clase aprenderemos cómo cambiar la velocidad según el movimiento de la muñeca de la mano derecha.

Este código se agregará dentro de una función `play()`.

```
function play()
{
  song.play();
  song.setVolume(1);
  song.rate(1);
}
```

- **Agregar código para inicializar el modelo Posenet**

```
function setup() {
  canvas = createCanvas(600, 500);
  canvas.center();

  video = createCapture(VIDEO);
  video.hide();

  poseNet = ml5.poseNet(video, modelLoaded);
}
```

- Agregando código para la función modelLoaded()

```
function setup() {  
  canvas = createCanvas(600, 500);  
  canvas.center();  
  
  video = createCapture(VIDEO);  
  video.hide();  
  
  poseNet = ml5.poseNet(video, modelLoaded);  
}  
  
function modelLoaded() {  
  console.log('PoseNet Is Initialized');  
}
```

Salida

PoseNet Is Initialized

[main.js:17](#)

- Agregar código para ejecutar el modelo Posenet

```
function setup() {  
  canvas = createCanvas(600, 500);  
  canvas.center();  
  
  video = createCapture(VIDEO);  
  video.hide();  
  
  poseNet = ml5.poseNet(video, modelLoaded);  
  poseNet.on('pose', gotPoses);  
}  
  
function modelLoaded() {  
  console.log('PoseNet Is Initialized');  
}
```

- Agregando el código de la función gotPoses()

```
function gotPoses(results)
{
    if(results.length > 0)
    {
        console.log(results);
    }
}
```

- Defina 2 variables para mantener las coordenadas x e y de la muñeca izquierda e inicialice con un valor de 0, defina estas variables al principio en el archivo main.js

```
song = "";
leftWristX = 0;
leftWristY = 0;

function preload()
{
    song = loadSound("music.mp3");
}
```

- Código para obtener la coordenada x de la muñeca izquierda y actualizar la variable izquierdaMuñecaX

```
function gotPoses(results)
{
    if(results.length > 0)
    {
        console.log(results);
        leftWristX = results[0].pose.leftWrist.x;
    }
}
```

- Código para obtener la coordenada y de la muñeca izquierda y actualizar la variable muñeca izquierda

```
function gotPoses(results)
{
  if(results.length > 0)
  {
    console.log(results);
    leftWristX = results[0].pose.leftWrist.x;
    leftWristY = results[0].pose.leftWrist.y;
  }
}
```

- Ahora consola estas dos variables

```
function gotPoses(results)
{
  if(results.length > 0)
  {
    console.log(results);
    leftWristX = results[0].pose.leftWrist.x;
    leftWristY = results[0].pose.leftWrist.y;
    console.log("leftWristX = " + leftWristX + " leftWristY = " + leftWristY);
  }
}
```

Salida

```

▶ [{-}]
scoreRightWrist = 0.002373021561652422 scoreLeftWrist = 0.0001994967315113172
rightWristX = 242.10033751370614 rightWristY = 440.54112350731566
leftWristX = 445.2759479128594 leftWristY = 430.8342643276757
▶ [{-}]
scoreRightWrist = 0.0003662311064545065 scoreLeftWrist = 0.00004254684972693212
rightWristX = 239.09887812058594 rightWristY = 472.52314100953333
leftWristX = 424.60807577211256 leftWristY = 492.545044659174
▶ [{-}]
scoreRightWrist = 0.0009220915962941945 scoreLeftWrist = 0.0002346446708543226
rightWristX = 259.59715871085893 rightWristY = 454.11855002360494
leftWristX = 450.46574583295256 leftWristY = 452.0261363722892

```

- Defina 2 variables para mantener las coordenadas x e y de la muñeca izquierda e inicialice con un valor de 0, defina estas variables al principio en el archivo main.js

```

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

function preload()
{
    song = loadSound("music.mp3");
}

```

- Código para obtener la coordenada x de la muñeca izquierda y actualizar la variable izquierdaMuñecaX

```

function gotPoses(results)
{
    if(results.length > 0)
    {
        console.log(results);
        leftWristX = results[0].pose.leftWrist.x;
        leftWristY = results[0].pose.leftWrist.y;
        console.log("leftWristX = " + leftWristX + " leftWristY = " + leftWristY);
        rightWristX = results[0].pose.rightWrist.x;
    }
}

```

- Código para obtener la coordenada y de la muñeca izquierda y actualizar la variable muñeca izquierda

```

function gotPoses(results)
{
    if(results.length > 0)
    {
        console.log(results);
        leftWristX = results[0].pose.leftWrist.x;
        leftWristY = results[0].pose.leftWrist.y;
        console.log("leftWristX = " + leftWristX + " leftWristY = " + leftWristY);

        rightWristX = results[0].pose.rightWrist.x;
        rightWristY = results[0].pose.rightWrist.y;
    }
}

```

- Ahora console estas dos variables

```

function gotPoses(results)
{
  if(results.length > 0)
  {
    console.log(results);
    leftWristX = results[0].pose.leftWrist.x;
    leftWristY = results[0].pose.leftWrist.y;
    console.log("leftWristX = " + leftWristX + " leftWristY = " + leftWristY);

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

```

## Salida

```

▶ [{...}]
scoreRightWrist = 0.002373021561652422 scoreLeftWrist = 0.0001994967315113172
rightWristX = 242.10033751370614 rightWristY = 440.54112350731566
leftWristX = 445.2759479128594 leftWristY = 430.8342643276757

▶ [{...}]
scoreRightWrist = 0.0003662311064545065 scoreLeftWrist = 0.00004254684972693212
rightWristX = 239.09887812058594 rightWristY = 472.52314100953333
leftWristX = 424.60807577211256 leftWristY = 492.545044659174

▶ [{...}]
scoreRightWrist = 0.0009220915962941945 scoreLeftWrist = 0.0002346446708543226
rightWristX = 259.59715871085893 rightWristY = 454.11855002360494
leftWristX = 450.46574583295256 leftWristY = 452.0261363722892

```