

::Create a Music Player App::

Design and develop a music player application for users to listen to their favorite music tracks.

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Simple Music Player</title>
```

```
<!-- Load FontAwesome icons -->
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.13.0/css/all.min.css">
```

```
<!-- CSS style -->
```

```
<style>
```

```
body {
```

```
background-color: lightgreen;
```

```
/* Smoothly transition the background color */
```

```
transition: background-color .5s;
```

```
}
```

```
.player {
```

```
height: 95vh;
```

```
display: flex;
```

```
align-items: center;
```

```
flex-direction: column;
```

```
justify-content: center;
```

```
}
```

```
.details {  
  display: flex;  
  align-items: center;  
  flex-direction: column;  
  justify-content: center;  
  margin-top: 25px;  
}
```

```
.track-art {  
  margin: 25px;  
  height: 250px;  
  width: 250px;  
  background-image: url("https://images.pexels.com/photos/262034/pexels-photo-262034.jpeg?auto=compress&cs=tinysrgb&dpr=3&h=750&w=1260");  
  background-size: cover;  
  border-radius: 15%;  
}
```

```
.now-playing {  
  font-size: 1rem;  
}
```

```
.track-name {  
  font-size: 3rem;  
}
```

```
.track-artist {  
  font-size: 1.5rem;  
}
```

```
.buttons {
```

```
display: flex;
flex-direction: row;
align-items: center;
}
```

```
.playpause-track,
.prev-track,
.next-track {
padding: 25px;
opacity: 0.8;
```

```
/* Smoothly transition the opacity */
transition: opacity .2s;
}
```

```
.playpause-track:hover,
.prev-track:hover,
.next-track:hover {
opacity: 1.0;
}
```

```
.slider_container {
width: 75%;
max-width: 400px;
display: flex;
justify-content: center;
align-items: center;
}
```

```
/* Modify the appearance of the slider */
.seek_slider,
```

```
.volume_slider {  
-webkit-appearance: none;  
-moz-appearance: none;  
appearance: none;  
height: 5px;  
background: black;  
opacity: 0.7;  
-webkit-transition: .2s;  
transition: opacity .2s;  
}
```

/* Modify the appearance of the slider thumb */

```
.seek_slider::-webkit-slider-thumb,  
.volume_slider::-webkit-slider-thumb {  
-webkit-appearance: none;  
-moz-appearance: none;  
appearance: none;  
width: 15px;  
height: 15px;  
background: white;  
cursor: pointer;  
border-radius: 50%;  
}
```

```
.seek_slider:hover,  
.volume_slider:hover {  
opacity: 1.0;  
}
```

```
.seek_slider {  
width: 60%;
```

```
}
```

```
.volume_slider {  
width: 30%;  
}
```

```
.current-time,  
.total-duration {  
padding: 10px;  
}
```

```
i.fa-volume-down,  
i.fa-volume-up {  
padding: 10px;  
}
```

```
i.fa-play-circle,  
i.fa-pause-circle,  
i.fa-step-forward,  
i.fa-step-backward {  
cursor: pointer;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="player">
```

```
<div class="details">
```

```
<div class="now-playing">PLAYING x OF y</div>
```

```
<div class="track-art"></div>
```

```
<div class="track-name">Track Name</div>
```

```
<div class="track-artist">Track Artist</div>
```

```

</div>

<div class="buttons">

<div class="prev-track" onclick="prevTrack()"><i class="fa fa-step-backward fa-2x"></i></div>

<div class="playpause-track" onclick="playpauseTrack()"><i class="fa fa-play-circle fa-5x"></i></div>

<div class="next-track" onclick="nextTrack()"><i class="fa fa-step-forward fa-2x"></i></div>

</div>

<div class="slider_container">

<div class="current-time">00:00</div>

<input type="range" min="1" max="100" value="0" class="seek_slider" onchange="seekTo()">

<div class="total-duration">00:00</div>

</div>

<div class="slider_container">

<i class="fa fa-volume-down"></i>

<input type="range" min="1" max="100" value="99" class="volume_slider"
onchange="setVolume()">

<i class="fa fa-volume-up"></i>

</div>

</div>

<!-- Main script for the player -->

<script>

let now_playing = document.querySelector(".now-playing");
let track_art = document.querySelector(".track-art");
let track_name = document.querySelector(".track-name");
let track_artist = document.querySelector(".track-artist");

let playpause_btn = document.querySelector(".playpause-track");
let next_btn = document.querySelector(".next-track");
let prev_btn = document.querySelector(".prev-track");

let seek_slider = document.querySelector(".seek_slider");

```

```

let volume_slider = document.querySelector(".volume_slider");
let curr_time = document.querySelector(".current-time");
let total_duration = document.querySelector(".total-duration");

let track_index = 0;
let isPlaying = false;
let updateTimer;

// Create new audio element
let curr_track = document.createElement('audio');

// Define the tracks that have to be played
let track_list = [
{
name: "Night Owl",
artist: "Broke For Free",
image: "https://images.pexels.com/photos/2264753/pexels-photo-2264753.jpeg?auto=compress&cs=tinysrgb&dpr=3&h=250&w=250",
path: "https://files.freemusicarchive.org/storage-freemusicarchive-org/music/WFMU/Broke_For_Free/Directionless_EP/Broke_For_Free_-_01_-_Night_Owl.mp3"
},
{
name: "Enthusiast",
artist: "Tours",
image: "https://images.pexels.com/photos/3100835/pexels-photo-3100835.jpeg?auto=compress&cs=tinysrgb&dpr=3&h=250&w=250",
path: "https://files.freemusicarchive.org/storage-freemusicarchive-org/music/no_curator/Tours/Enthusiast/Tours_-_01_-_Enthusiast.mp3"
},
{
name: "Shipping Lanes",
artist: "Chad Crouch",

```

```
image: "https://images.pexels.com/photos/1717969/pexels-photo-1717969.jpeg?auto=compress&cs=tinysrgb&dpr=3&h=250&w=250",
path: "https://files.freemusicarchive.org/storage-freemusicarchive-org/music/ccCommunity/Chad_Crouch/Arps/Chad_Crouch_-_Shipping_Lanes.mp3",
},
];
```

```
function loadTrack(track_index) {
clearInterval(updateTimer);
resetValues();

// Load a new track
curr_track.src = track_list[track_index].path;
curr_track.load();

// Update details of the track
track_art.style.backgroundImage = "url(" + track_list[track_index].image + ")";
track_name.textContent = track_list[track_index].name;
track_artist.textContent = track_list[track_index].artist;
now_playing.textContent = "PLAYING " + (track_index + 1) + " OF " + track_list.length;

// Set an interval of 1000 milliseconds for updating the seek slider
updateTimer = setInterval(seekUpdate, 1000);

// Move to the next track if the current one finishes playing
curr_track.addEventListener("ended", nextTrack);

// Apply a random background color
random_bg_color();
}

function random_bg_color() {
```



```
// Get a random number between 64 to 256 (for getting lighter colors)
```

```
let red = Math.floor(Math.random() * 256) + 64;
```

```
let green = Math.floor(Math.random() * 256) + 64;
```

```
let blue = Math.floor(Math.random() * 256) + 64;
```

```
// Construct a color with the given values
```

```
let bgColor = "rgb(" + red + "," + green + "," + blue + ")";
```

```
// Set the background to that color
```

```
document.body.style.background = bgColor;
```

```
}
```

```
// Reset Values
```

```
function resetValues() {
```

```
curr_time.textContent = "00:00";
```

```
total_duration.textContent = "00:00";
```

```
seek_slider.value = 0;
```

```
}
```

```
function playpauseTrack() {
```

```
if (!isPlaying) playTrack();
```

```
else pauseTrack();
```

```
}
```

```
function playTrack() {
```

```
curr_track.play();
```

```
isPlaying = true;
```

```
// Replace icon with the pause icon
```

```
playpause_btn.innerHTML = '<i class="fa fa-pause-circle fa-5x"></i>';
```

```
}
```

```
function pauseTrack() {
```

```
    curr_track.pause();
```

```
    isPlaying = false;
```

```
    // Replace icon with the play icon
```

```
    playpause_btn.innerHTML = '<i class="fa fa-play-circle fa-5x"></i>';
```

```
}
```

```
function nextTrack() {
```

```
    if (track_index < track_list.length - 1)
```

```
        track_index += 1;
```

```
    else track_index = 0;
```

```
    loadTrack(track_index);
```

```
    playTrack();
```

```
}
```

```
function prevTrack() {
```

```
    if (track_index > 0)
```

```
        track_index -= 1;
```

```
    else track_index = track_list.length;
```

```
    loadTrack(track_index);
```

```
    playTrack();
```

```
}
```

```
function seekTo() {
```

```
    seekto = curr_track.duration * (seek_slider.value / 100);
```

```
    curr_track.currentTime = seekto;
```

```
}
```

```

function setVolume() {
  curr_track.volume = volume_slider.value / 100;
}

function seekUpdate() {
  let seekPosition = 0;

  // Check if the current track duration is a legible number
  if (!isNaN(curr_track.duration)) {
    seekPosition = curr_track.currentTime * (100 / curr_track.duration);
    seek_slider.value = seekPosition;

    // Calculate the time left and the total duration
    let currentMinutes = Math.floor(curr_track.currentTime / 60);
    let currentSeconds = Math.floor(curr_track.currentTime - currentMinutes * 60);
    let durationMinutes = Math.floor(curr_track.duration / 60);
    let durationSeconds = Math.floor(curr_track.duration - durationMinutes * 60);

    // Adding a zero to the single digit time values
    if (currentSeconds < 10) { currentSeconds = "0" + currentSeconds; }
    if (durationSeconds < 10) { durationSeconds = "0" + durationSeconds; }
    if (currentMinutes < 10) { currentMinutes = "0" + currentMinutes; }
    if (durationMinutes < 10) { durationMinutes = "0" + durationMinutes; }

    curr_time.textContent = currentMinutes + ":" + currentSeconds;
    total_duration.textContent = durationMinutes + ":" + durationSeconds;
  }
}

// Load the first track in the tracklist
loadTrack(track_index);

```

</script>

</body>

</html>