JAVASCRIPT WEB API'S

1) WEB AUDIO API

The Web Audio API is a JavaScript API that provides powerful tools for creating and manipulating audio in web applications.

javascript

```javascript
// Create a new AudioContext object
const audioContext = new AudioContext();

// Load an audio file using an AudioBufferSourceNode
const audioFile = 'path/to/audiofile.mp3';
const audioBufferSource = audioContext.createBufferSource();
fetch(audioFile)
  .then(response => response.arrayBuffer())
  .then(buffer => audioContext.decodeAudioData(buffer))
  .then(audioBuffer => {
    audioBufferSource.buffer = audioBuffer;
    audioBufferSource.connect(audioContext.destination);
    audioBufferSource.start(0);
  })
  .catch(error => console.error('Error loading audio file:', error));
```

1) WEB AUDIO API

In this example, we create a new AudioContext object and load an audio file using the **fetch()** function and the **AudioContext.decodeAudioData()** method. Once the audio file has been decoded into an **AudioBuffer**, we create an AudioBufferSourceNode, set its **buffer** property to the decoded audio data, and connect it to the destination of the AudioContext. Finally, we start playing the audio file by calling the **start()** method of the **AudioBufferSourceNode**.

2) JAVASCRIPT CLIPBOARD API

The JavaScript Clipboard API allows developers to programmatically interact with the clipboard (i.e. copy/paste functionality) of a user's device. It provides a way to read, write, and manipulate clipboard content using JavaScript code.

2) JAVASCRIPT CLIPBOARD API

javascript

```javascript
const copyToClipboard = (text) => {
  navigator.clipboard.writeText(text)
    .then(() => {
      console.log("Text copied to clipboard");
    })
    .catch((err) => {
      console.error("Error copying text to clipboard: ", err);
    });
};
```

3)JAVASCRIPT VIBRATION API

The Vibration API is a JavaScript API that allows developers to trigger a device's vibration motor (if available) to provide haptic feedback to users. The API is available on most modern browsers on mobile devices.

javascript

```javascript
if ('vibrate' in navigator) {
  // Vibrate for 1000 milliseconds (1 second)
  navigator.vibrate(1000);
} else {
  console.warn('Vibration not supported');
}
```

3)JAVASCRIPT VIBRATION API

In this example, we first check if the navigator object contains a vibrate property, which indicates that the device supports the Vibration API. If it does, we call the **navigator.vibrate()** method and pass in the number of milliseconds to vibrate for. In this case, we vibrate for 1000 milliseconds (1 second).

4) JAVASCRIPT ANIMATION API

The Animation API is a JavaScript API that allows developers to create animations and transitions using JavaScript code.

The API provides a way to create complex animations with fine-grained control over timing and animation curves.

4) JAVASCRIPT ANIMATION API

```
const element = document.querySelector('#my-element');

// Create a new animation object
const animation = element.animate(
  [
    { transform: 'translateX(0px)' },
    { transform: 'translateX(100px)' }
  ], {
    duration: 1000, // 1 second
    easing: 'ease-out', // Animation curve
    fill: 'forwards' // Stay at the end position after the animation is finished
  });

// Start the animation
animation.play();
```

5) MEDIA CAPTURE API

The Media Capture API is a JavaScript API that allows developers to access media input devices, such as cameras and microphones, and capture media streams from those devices. The API is available on most modern browsers on desktop and mobile devices.

5) MEDIA CAPTURE API

Here is an example.

```
const captureButton = document.querySelector('#capture-button');
const photoPreview = document.querySelector('#photo-preview');

captureButton.addEventListener('click', async () => {
  try {
    // Request permission to access the camera
    const stream = await navigator.mediaDevices.getUserMedia({ video: true });

    // Create a new video element to preview the stream
    const videoElement = document.createElement('video');
    videoElement.srcObject = stream;
    videoElement.autoplay = true;
    document.body.appendChild(videoElement);

    // Wait for the video stream to start playing
    await videoElement.play();

    // Create a canvas element to capture a photo
    const canvas = document.createElement('canvas');
    canvas.width = videoElement.videoWidth;
    canvas.height = videoElement.videoHeight;

    // Draw the video frame onto the canvas
    const context = canvas.getContext('2d');
    context.drawImage(videoElement, 0, 0, canvas.width, canvas.height);

    // Convert the canvas to a data URL and set it as the photo preview
    const dataUrl = canvas.toDataURL();
    photoPreview.src = dataUrl;

    // Stop the video stream
    stream.getTracks().forEach(track => track.stop());
    videoElement.remove();
  } catch (error) {
    console.error('Error capturing photo:', error);
  }
});
```

6) WEB SHARE API

The Web Share API is a JavaScript API that allows developers to add a "Share" button to their website or web application, which allows users to share content directly from the webpage to their social media accounts or messaging apps. The Web Share API is available on many modern browsers, including Chrome, Firefox, Safari, and Edge.

6) WEB SHARE API

javascript

```javascript
const shareButton = document.querySelector("#share-button");

shareButton.addEventListener("click", async () => {
  try {
    await navigator.share({
      title: "My awesome website",
      text: "Check out this awesome website I found!",
      url: "https://example.com",
    });
  } catch (error) {
    console.error("Error sharing:", error);
  }
});
```

6) WEB SHARE API

we select a button with the ID share-button and add a click event listener to it. When the button is clicked, we call the navigator.share() method, which opens a native sharing dialog provided by the user's operating system. The navigator.share() method accepts an object with three properties: title, text, and url. These properties are used to populate the sharing dialog with information about the shared content.