PiP on the web

I love listening to music in the background or having a video play while I'm doing something else, something like what iTunes does when you minimize the app.

There is an API that allows you to do somethng similar. The Picture in Picture API is currently under incubation in the Web Incubation Community Group and an (incomplete) implementation available in Chrome since version 70 (behind the the `chrome://flags/#enable-surfaces-for-videos` flag) and Opera since version 57 (behind the the `opera://flags/#enable-surfaces-for-videos` flag).

The HTML document looks like any other video. We use the `playsinline` attribute to make sure it won't go to fullscreen in iOS devices.

We also add a button to trigger the picture in picture process.

```html
<video id="video"
    controls
    playsinline
    src="video/short.mp4"
    poster="video/artwork-512.png"></video>
<button id="togglePipButton">Toggle Picture-in-Picture</button>
```

Since the code is promise based so we can either use raw promises or async/await code. In most circumstances I'd go with raw promises since it gives me slightly better browser support but, since the feature is brand new we don't have to worry about browser support anyways... so async/await it is. :)

The code is broken down into three events and an auxiliary function. The first function is a click event handler in the button.

We disable the toggle button while we wait for the actions to happen.

in the try block we test if the video is already in the picture in picture window. If it is not then we add it to the PiP window and if it is (meaning the video is already playing in the PiP window) we remove it.

In the catch block we log an error to console.

In the finally block we re-enable the PiP button. This will happen regardless of how we got to the finally block.

```
let pipWindow;

togglePipButton.addEventListener('click', async function(event) {
  console.log('Toggling Picture-in-Picture...');
  togglePipButton.dis'Toggling Picture-in-Picture...'deo !== document.pic
      await video.requestPictureInPicture();
    else await document.exitPictureInPicture();
  } catch (error) {
    console.log(`I'm sorry David, I can't do that: ${error}`);
  } finally {
    togglePipButton.disabled = false;
  }
});
```

The next two events are picture-in-picture related; Once for when the video enters the picture in picture window and the other for when the video is removed from the PiP window.

The event sets `onPipWindowResize` as the resize event handler.

```
video.addEventListener('enterpictureinpicture', function(event) {
  console.log('Video Playing in Picture-in-Picture');

  pip'Video Playing in Picture-in-Picture';
  console.log(`The new window size is
    ${pipWindow.width}x${pipWindow.height}`);

  pipWindow.addEventListener('resize', onPipWindowResize);
});
'resize'
```

The `leavepictureinpicture` notifies when the video is removed from the PiP window and retakes the original position.

The event removes `onPipWindowResize` as the resize event handler since we no longer need it.

```
video.addEventListener('leavepictureinpicture', function(event) {
  console.log('Exiting Picture-in-Picture');

  pipWindow.removeEventListener('resize', onPipWindowResize);
});
```

`onPipWindowResize` is the function that will get called when the PiP window is resized. In this example we just log the new window size to the console.

```
function onPipWindowResize(event) {
  console.log(
    `New window size:
    ${pipWindow.width}x${pipWindow.height}`,
  );
}
```

The final portion of the script is to enable the button.

```
if ('pictureInPictureEnabled' in document) {
  setPipButton();
  video.addEventListener('loadedmetadata', setPipButton);
  video.addEventListener('emptied', setPipButton);
} else {
  togglePipButton.hidden = true;
}

function setPipButton() {
  togglePipButton.disabled =
    video.readyState === 0 ||
    !document.pictureInPictureEnabled ||
    video.disablePictureInPicture;
}
```

Links and Resources

- Picture in Picture Explainer
- Chrome Example
- Chrome Example Code
- Watch video using Picture-in-Picture