Today, we are going to be making a drum kit that can live on a website or locally. The idea is quite simple, you have tiles showing the different items that make up a drum kit. Each tile is assigned to a letter on the keyboard so when you press the letter, you get the sound. You can also click on the tile for the sound. The html file is pretty simple and loos like this.

<insert html>

Each button has a class of drum so in order to get the number we use query selector all to get them and then find the length. After doing that, we add a click event listener for each of them and when the click happens we call the drumSound function.

var numberOfDrumButtons = document.querySelectorAll(".drum").length;

for (i = 0; i < numberOfDrumButtons; i++) {

  document.querySelectorAll(".drum")[i].addEventListener("click", drumSound);

}

In the <code>drumSound()</code> function, we get the inner html of the button that got clicked. The inner html are the different button letters. Based on the letter we get back, we call the make sound function and the button animation function while passing it in as a parameter.

function drumSound() {

  // "this" selects the particular button that gets clicked.

// code below detects te button click.

  var buttonInnerHTML = this.innerHTML;

  makeSound(buttonInnerHTML);

  buttonAnimation(buttonInnerHTML);

}

Our <code>makeSound()</code> function just makes use of a switch and addresses all the various cases. This can also be done with if…else statements. The different cases correspond to the different inner html of the buttons.

function makeSound(key) {

  switch (key) {

    case "w":

      var tom1 = new Audio("sounds/tom-1.mp3");

      tom1.play();

      break;

    case "a":

      var tom2 = new Audio("sounds/tom-2.mp3");

      tom2.play();

      break;

    case "s":

      var tom3 = new Audio("sounds/tom-3.mp3");

      tom3.play();

      break;

    case "d":

      var tom4 = new Audio("sounds/tom-4.mp3");

      tom4.play();

      break;

    case "j":

      var snare = new Audio("sounds/snare.mp3");

      snare.play();

      break;

    case "k":

      var crash = new Audio("sounds/crash.mp3");

      crash.play();

      break;

    case "l":

      var kick = new Audio("sounds/kick-bass.mp3");

      kick.play();

      break;

    default:

      console.log(key);

  }

}

For each case, we create a new audio and them we call the play method on them. Our <code>buttonAnimation()</code> function below just gets the button that was pressed by using query selector and passing in the class as specified in the html. After it gets the button, it then add a class of pressed to it and after a while, it removes it. The pressed class is just to add a small effect so we know that the drum was played.

function buttonAnimation(currentKey) {

  var activeButton = document.querySelector("." + currentKey);

  activeButton.classList.add("pressed");

  setTimeout(function(){

    activeButton.classList.remove("pressed");

  }, 100);

}

What we have right now, just takes care of when we click it, because we all these would happen remember when we click on a button. In order for us to be able to press the keys that correspond to the buttons, we are going to add a keydown event listener to our document. This would help us get the keys we press on our keyboard.<br />

When we press a key, we call the <code>keyboardSound()</code> function which would simply just call the <code>makeSound()</code> and <code>buttonAnimation()</code> functions while passing in the <code>event.key</code> which equals to the letter pressed by the user.

document.addEventListener("keydown", keyboardSound);

function keyboardSound (event) {

  makeSound(event.key);

  buttonAnimation(event.key);

}

<br />

Cheers 🥂