

# Uitleg JavaScript Drum Kit

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>JS Drum Kit</title>
  <link rel="stylesheet" href="style.css">
  <link rel="icon" href="https://fav.farm/✔" />
  <style>
    body {
      background-image: url("img/default.jpg");
      background-repeat: no-repeat;
      background-size: cover;
    }
  </style>
</head>
<body>

  <audio data-key="65" src="sounds/clap.wav"></audio>
  <audio data-key="83" src="sounds/hihat.wav"></audio>
  <audio data-key="68" src="sounds/kick.wav"></audio>
  <audio data-key="70" src="sounds/openhat.wav"></audio>
  <audio data-key="71" src="sounds/boom.wav"></audio>
  <audio data-key="72" src="sounds/ride.wav"></audio>
  <audio data-key="74" src="sounds/snare.wav"></audio>
  <audio data-key="75" src="sounds/tom.wav"></audio>
  <audio data-key="76" src="sounds/tink.wav"></audio>

<script>
  function removeTransition(e) {
    if (e.propertyName !== 'transform') return;
    e.target.classList.remove('playing');
  }

  const mainBackgroundImage =
getComputedStyle(document.body).backgroundImage;
  let keyDown = false; // To keep track of key press state

  function playSound(e) {
    if (keyDown) return; // Exit if key is already pressed down
    const audio = document.querySelector(`audio[data-key="${e.keyCode}"]`);
    const key = document.querySelector(`div[data-key="${e.keyCode}"]`);
    if (!audio) return;

    keyDown = true; // Set key press state to true
    key.classList.add('playing');
```

```

audio.currentTime = 0;
audio.play();

// Change background image based on the key pressed
const imageName = audio.src.split('/').pop().split('.')[0];
const body = document.querySelector('body');
body.style.backgroundImage = `url("img/${imageName}.jpeg")`;

// Revert back to the main background image after a certain duration
setTimeout(() => {
  body.style.backgroundImage = mainBackgroundImage;
  keyDown = false; // Set key press state back to false
}, 100); // 100 milliseconds (0.1 second)
}

const keys = Array.from(document.querySelectorAll('.key'));
keys.forEach(key => {
  key.addEventListener('transitionend', removeTransition);
  key.addEventListener('keyup', () => {
    keyDown = false; // Set key press state to false on keyup event
  });
});
window.addEventListener('keydown', playSound);
</script>
</body>
</html>

```

Hieronder staat kort uitgelegd hoe je een drumkit kan bouwen:

1. Allereerst is er een div-container met de class "keys", die de toetsen van de drumkit bevat. Elke toets wordt gerepresenteerd door een div-element met de class "key". Binnen elke "key" wordt de letter van de toets (bv. A, S, D) weergegeven met de kbd-tag, en de naam van het geluid (bv. clap, hihat) wordt weergegeven met de span-tag.
2. Onder de div-container met de class "keys" bevinden zich audiocomponenten voor elk geluid. Elk audio-element is gekoppeld aan een specifieke toets met behulp van het attribuut data-key, dat de bijbehorende toetscode bevat. Bijvoorbeeld, de eerste audio-tag heeft data-key="65", wat overeenkomt met de letter 'A' (keyCode 65) van de toets.
3. Na de HTML-markup volgt de JavaScript-code. De script-tag omsluit de JavaScript-functionaliteit. De code maakt gebruik van enkele functies die gedefinieerd zijn in JavaScript.
4. De functie "removeTransition" wordt gedefinieerd. Deze functie wordt gebruikt om de visuele overgangseffecten van een toets te verwijderen nadat de overgang (transition) is voltooid. Het controleert of het eigenschapstype (propertyName) van de overgang 'transform' is en verwijdert vervolgens de klasse 'playing' van het betreffende element.

5. De functie "playSound" wordt gedefinieerd. Deze functie wordt uitgevoerd wanneer een toets wordt ingedrukt. Het haalt het audio-element op dat overeenkomt met de ingedrukte toetscode en het bijbehorende div-element. Als er geen audio-element is voor de toets, wordt de functie vroegtijdig verlaten. Anders wordt aan het div-element de klasse 'playing' toegevoegd, zodat er een visueel effect wordt getoond, en wordt het audio-element afgespeeld. De currentTime van het audio-element wordt op 0 gezet om ervoor te zorgen dat het geluid altijd vanaf het begin begint te spelen.
6. Vervolgens worden er een aantal eventlisteners toegevoegd. De "transitionend"-event wordt toegevoegd aan elk toets-element (.key). Wanneer de overgang (transition) van een toets is voltooid, wordt de functie "removeTransition" uitgevoerd om de visuele effecten te verwijderen.
7. De "keydown"-event wordt toegevoegd aan het venster (window). Wanneer er een toets wordt ingedrukt, wordt de functie "playSound" uitgevoerd om het bijbehorende geluid af te spelen en de visuele effecten te tonen.
8. Als extraatje heb ik ook afbeeldingen toegevoegd aan de drumkit. Voor elke geluidstoets heb ik een afbeelding met dezelfde naam toegevoegd (Alleen dan als '.jpeg' extensie i.p.v. '.wav') waardoor de naam van de afbeelding overeen komt met de naam van het geluid. In de JavaScript wordt de achtergrondafbeelding van de pagina gewijzigd op basis van de ingedrukte toets. Hierdoor wordt de drumkit niet alleen interactief met geluiden, maar ook met bijpassende afbeeldingen.