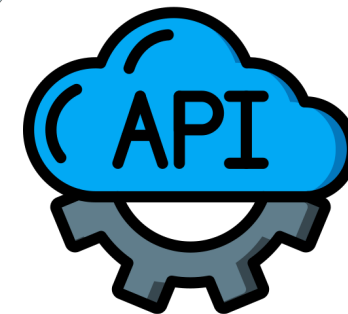
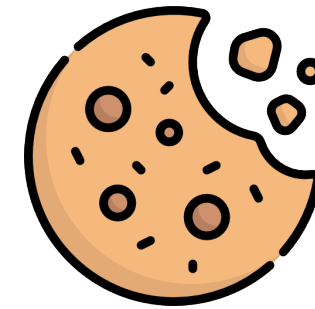


HTML, CSS & JS basis

Les 6:

DOM en eventhandling, & Cookies

JS



Onderwerpen les:

- Wat is DOM?
- Elementen vinden in de DOM
- Elementen aanpassen
- Wat zijn Events?
- Event handling
- Local storage, session storage & cookies
- Opdracht

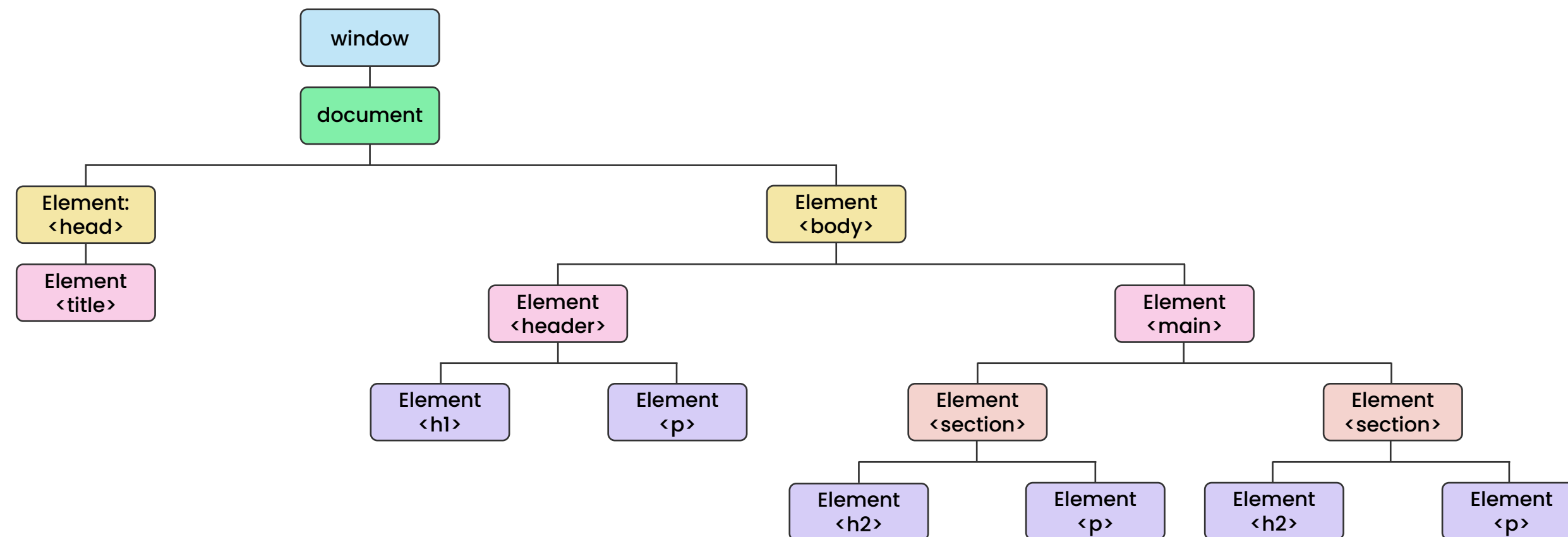
Leerdoelen

- ✓ Aan het einde van de dag kun je uitleggen wat de DOM is
- ✓ Aan het einde van de dag kan je DOM elementen selecteren en aanpassen
- ✓ Aan het einde van de dag kan je events afhandelen
- ✓ Aan het einde van de dag kun je cookies opslaan en ophalen

Document object model

Wat is DOM?

Wanneer een webpagina wordt geladen creëert de browser een Document Object Model van de pagina. De DOM is gestructureerd als een stamboom van objecten/dom elementen/nodes.



Document object model

Wat is DOM?

properties van dom elementen

Een dom element heeft een hele lijst aan properties (niet de CSS properties). Bijvoorbeeld een tagName, className, title, innerHTML, innerText en nog veel meer.

Bekijk de volledige lijst aan properties van een element via chrome dev tools.

1: selecteer een element

2: Tab -> Elements

3 Subtab onderaan -> Properties

| https://www.w3schools.com/js/js_htmlDOM.asp

Document object model

DOM manipulatie

selecteren elementen:

`getElementById('name_id')`

Wil je slechts 1 element selecteren uit de DOM? Dan kan je heel gemakkelijk het element selecteren aan de hand van zijn id.

```
let getsomething = document.getElementById('some-id');
```

`getElementsByClassName('name_class')`

Wil je 1 of meerdere elementen ophalen aan de hand van zijn class name? Deze methode zal een lijst van elementen opslaan.

```
let getsomestuff = document.getElementsByClassName('some-class');
```

Document object model

DOM manipulatie

selecteren elementen:

`getElementByTagName('div')`

Wil je meerdere elementen selecteren waarvan de Tag naam gelijk is? Gebruik dan `getElementByTagName`.

```
let getsomething = document.getElementsByTagName('li');
```

Document object model

DOM manipulatie

selecteren elementen:

`querySelector('div') & querySelectorAll('.div')`

De `querySelector()` methode geeft het eerst gevonden element die voldoet aan de query. Wil je alle resultaten die voldoen aan de query? Gebruik dan `querySelectorAll()`

Met deze twee methoden kun je geavanceerder zoeken dan met de voorgaande `getElement` methoden.

```
let getsomething = document.querySelectorAll("h1, h2, h3, h4, h5, h6");
```

alle headings

```
let getsomething = document.querySelectorAll("p, .classname, #idname");
```

zoek op tag, class, id

Document object model

DOM manipulatie

selecteren elementen:

`querySelector('div')` & `querySelectorAll('.div')`

```
let getsomething = document.querySelector(".myclass > p");
```

gebruik geavanceerde selectors zoals in CSS

```
let getsomething = document.querySelector("input[name=inputName]");
```

zoek aan de hand van een attribuut waarde

Document object model

DOM manipulatie

selecteren elementen:

parentElement

Wil je de parent van een bepaald element ophalen? gebruik dan .parentElement.

```
let getparent = document.querySelector(".myclass ").parentElement;
```

Document object model

DOM manipulatie

Elementen aanpassen

`.style.cssproperty`

De styling van een geselecteerd element kun je aanpassen met `.style.property = "value";`
Hiermee kun bijvoorbeeld de achtergrond kleur aanpassen of een element zijn display op none of juist block zetten en nog veel meer.

```
someElement.style.backgroundColor = "red";
```

verander element backgroundColor naar rood

Document object model

DOM manipulatie

Elementen aanpassen

`.innerHTML`

De `innerHTML` haalt de content op van een element (tekst, spaties en tags).

```
let getInnerHTML = someElement.innerHTML;
```

De `innerHTML` kunnen we ook veranderen of er iets aantoevoegen (+ of +=).

```
someElement.innerHTML = "hallo";
```

Document object model

DOM manipulatie

selecteren elementen:

- | https://www.w3schools.com/jsref/met_document_getelementbyid.asp
- | https://www.w3schools.com/jsref/met_document_getelementsbyclassname.asp
- | https://www.w3schools.com/jsref/met_document_getelementsbytagname.asp
- | https://www.w3schools.com/jsref/met_document_queryselector.asp
- | https://www.w3schools.com/jsref/met_document_queryselectorall.asp
- | https://www.w3schools.com/jsref/prop_node_parentelement.asp

Document object model

DOM manipulatie

Elementen aanpassen

.innerHTML, multiline tag en templated template literals

Met de backticks kun je een string op meerdere regels plaatsen. Aan de hand van tags kun je variabelen en andere javascript code gebruiken binnen de string. `${some javascript}`

```
let name = "Maurice";
let htmlinput = `
<h1>welkom op de website van ${name}</h1>
<p>paragraph lorem ipsum</p>
`;
someElement.innerHTML = htmlinput;
```

Document object model

DOM manipulatie

Elementen aanpassen

- | https://www.w3schools.com/js/js_html_dom_css.asp
- | https://www.w3schools.com/jsref/prop_html_innerhtml.asp
- | https://www.w3schools.com/js/js_string_templates.asp

Document object model

DOM manipulatie

Elementen toevoegen, verplaatsen en verwijderen

`.createElement()`

In javascript kun je nieuwe elementen creëren met `.createElement('tag');`

```
let newElement = document.createElement('div');
```

Maken nieuw element

`.appendChild()`

Het nieuw aangemaakte element kun je overal in de DOM plaatsen

```
selectedElement.appendChild(newElement);
```

Plaatsen element

Document object model

DOM manipulatie

Elementen toevoegen, verplaatsen en verwijderen

.remove()

Elementen verwijderen kan met remove()

```
selectedElement.remove();
```

Verwijderen element

- | https://www.w3schools.com/jsref/met_document_createelement.asp
- | https://www.w3schools.com/jsref/met_node_appendchild.asp
- | https://www.w3schools.com/jsref/met_element_remove.asp

Document object model

DOM manipulatie

classList

`.classList.add("mystyle")`

De styling van een element kun je ook aanpassen door een class toe te voegen of juist te verwijderen

```
selectedElement.classList.add("mystyle");
```

class toevoegen

```
selectedElement.classList.remove("mystyle");
```

Verwijderen element

Document object model

DOM manipulatie

classList

```
selectedElement.classList.toggle("mystyle");
```

toggle tussen classes (toevoegen of verwijderen)

- | https://www.w3schools.com/jsref/prop_element_classlist.asp
- | https://www.w3schools.com/jsref/met_domtokenlist_add.asp
- | https://www.w3schools.com/jsref/met_domtokenlist_remove.asp
- | https://www.w3schools.com/jsref/met_domtokenlist_toggle.asp

Events

Wat zijn events?

HTML DOM events

HTML DOM events zijn “dingen” die gebeuren bij HTML elementen. Met JavaScript kunnen wij reageren op deze events. Voorbeeld events: De pagina is klaar met laden, er wordt geklikt op een element, een input waarde is veranderd.

Er zijn ontzettend veel events, deze kun je verdelen in groepen zoals: KeyboardEvents, MouseEvents, FormEvents en meer. Deze events behoren allemaal tot het Event Object.

| https://www.w3schools.com/js/js_events.asp

Events

Event handling

toevoegen event in HTML

Events kunnen we toevoegen als attribuut aan onze HTML elementen.

```
<button onclick="myFunction()">click me</div>
```

```
<button onclick="alert('hello')">click me</div>
```

Events

Event handling

toevoegen event in JavaScript

In voorgaande hoofdstukken hebben we geleerd hoe een element kan worden opgehaald in JavaScript. We kunnen van elk opgehaald element een event afhandelen met functies.

```
selectedElement.addEventListener("click", function() {  
    alert("Hello World");  
});
```

Events

Event handling

toevoegen event in JavaScript

Natuurlijk zijn er ook situaties waar je voor meerdere elementen dezelfde actie wil uitvoeren bij een bepaald event.

```
moreElements.forEach(function(item) {  
    item.addEventListener("click", function() {  
        alert("Hello World");  
    });  
});
```

Events

Event handling

Populaire events

onclick	Gebruiker klikt op een element
onchange	Een HTML element is veranderd (bv: input value)
onmouseover	Gebruiker hoovert over een element
onmouseout	Gebruiker hover weg van het element
onkeydown	Gebruikt drukt op het keyboard
onload	De browser is klaar met het laden van de pagina

Storage

Wat is Storage?

Cookies vs local storage vs session storage

Cookies zijn bedoeld om gelezen te worden door een server, local storage en session storage zijn bedoeld voor het lezen door de browser. De local storage verloopt niet en session storage wordt leeggemaakt zodra de sessie afgelopen is.

Bekijk cookies en local- session s

inspecteren > application > cookies/local storage/ session storage

- | https://www.w3schools.com/jsref/prop_doc_cookie.asp
- | https://www.w3schools.com/jsref/prop_win_localstorage.asp
- | https://www.w3schools.com/jsref/prop_win_sessionstorage.asp

Storage

Cookies

Aanmaken cookies

```
function formFunction() {  
  let username = document.getElementById('username').value;  
  document.cookie = `username = ${username}`;  
}
```

```
function formFunction() {  
  let username = document.getElementById('username').value;  
  document.cookie = `username = ${username}`;  
}
```

Storage

Cookies

Ophalen cookies

```
function getCookie(cname) {  
    let name = cname + "=";  
    let ca = document.cookie.split(';');  
    for (let i = 0; i < ca.length; i++) {  
        let c = ca[i];  
        while (c.charAt(0) == ' ') {  
            c = c.substring(1);  
        }  
        if (c.indexOf(name) == 0) {  
            return c.substring(name.length, c.length);  
        }  
    }  
    return "";  
}
```

Opdracht

Drumpad

3 drumsets

Maak een drumpad waar 3 verschillende drumpad sets in zitten. (3 sets met verschillende geluiden/instrumenten).

click en keydown events

Klikt de gebruiker op een drumpad item dan speelt het bijbehorende geluid af. Klinkt iemand op het toetsenbord en correspondeert de key met 1 van de drumpad items? Dan speelt het bijbehorende geluid af.

Gebruikers naam

Vraag de gebruiker om zijn naam via een prompt. Plaats deze naam in de pagina. Plaats deze naam ook in een cookie (of local storage). Bij een volgend bezoek wordt gecontroleerd of een naam is opgeslagen als cookie. Is dit het geval? Dan wordt er niet opnieuw gevraagd om de naam. De gebruiker kan de naam wijzigen door te klikken op change name, dit zal de prompt opnieuw aanroepen.

Opdracht

Drumpad

Geluiden

Je mag zelf geluiden van instrumenten of iets anders downloaden. Op mijn github bij deze les staat een zipbestand met drum geluiden.

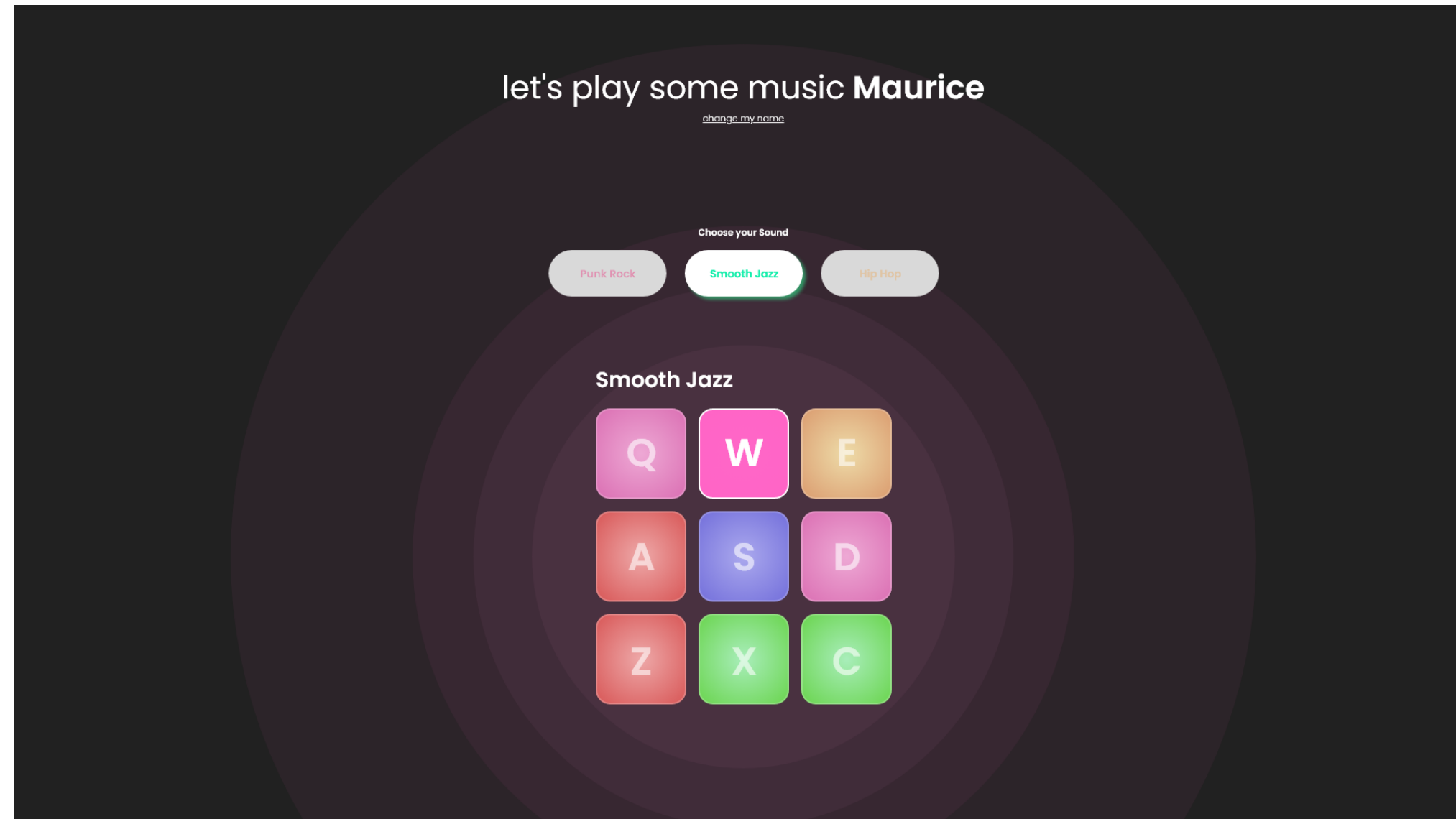
Extra: animaties

Zodra een geluid wordt afgespeeld kan de knop oplichten en misschien op de achtergrond de kleur van de knop met een leuk effect tevoorschijn komen.

Opdracht

Drumpad

Design voorbeeld



Opdracht

Drumpad

Tips

Geluiden als array aan objecten

Sla de geluiden op als een array aan objecten die geladen worden op het scherm

```
var allSounds = [  
  {  
    naam:"boom",  
    soundLink: new Audio('sounds/boom.wav'),  
    color: "#0C87FA",  
    icon: "img/base.svg"  
  },  
  {  
    naam:"clap",  
    soundLink: new Audio('sounds/clap.wav'),  
    color: "#6CC223",  
    icon: "img/clap.svg"  
  },  
  // animal sounds  
  {  
    naam:"chicken",  
    soundLink: new Audio('sounds/chicken.wav'),  
    color: "#FEEC66",  
    icon: "img/chicken.svg"  
  },  
  {  
    naam:"cat",  
    soundLink: new Audio('sounds/cat.wav'),  
    color: "#FEEC66",  
    icon: "img/cat.svg"  
  }  
];
```

Opdracht

Drumpad

Tips

Geluid afspelen

Met `naamVariable.play()` kun je audio afspelen.

```
let audio = new Audio('audio_file.mp3');  
audio.play();
```