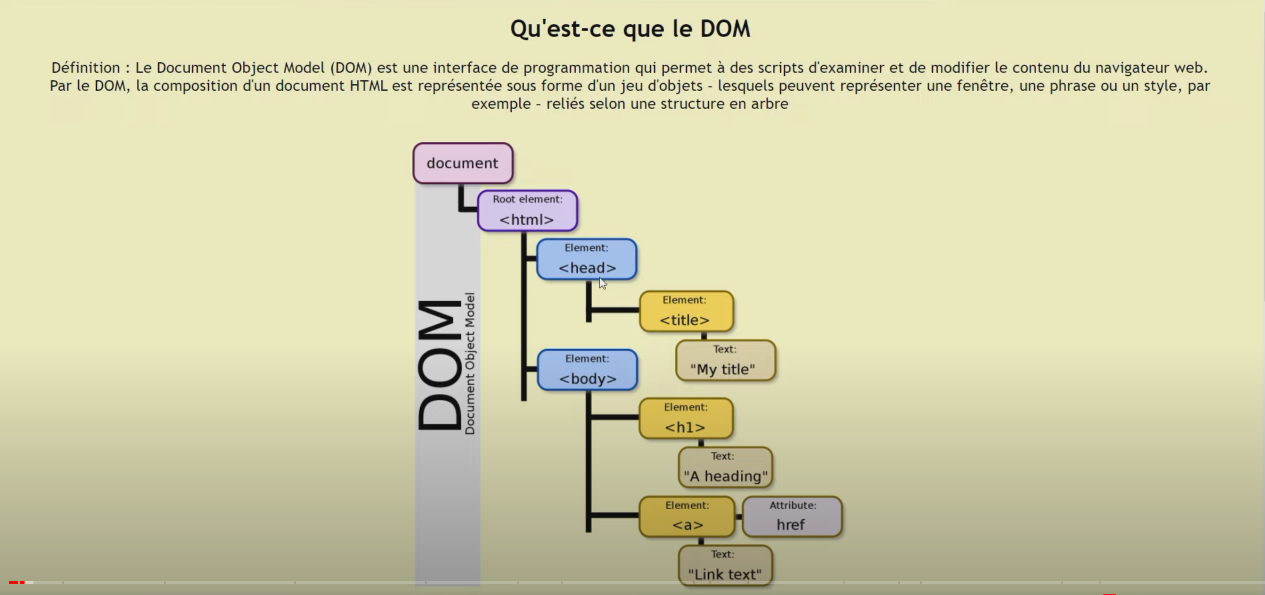
Le DOM JAVASCRIPT (vidéo Youtube)

<https://www.youtube.com/watch?v=6q-zt0aQ74U&t=5392s>



**Bonnes pratiques :**

* On peut utiliser la propriété visibility au lieu de la propriété display pour cacher puis montrer un élément.
* Ne pas hésiter à travailler dans la console qui agit comme le fichier JS pour faire des tests (ex : les Query Selectors)
* Une « méthode » ( comme querySelector) est une fonction déjà codée
* Lorsqu’on crée une variable, penser à toujours faire un console log pour voir si elle est bien selectionnée

Click events

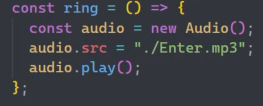
* addEventListener prend deux paramètres : (« évènement », (fonction, evenement appelé e ou vide) => { } ) ;
* On peut faire des transitions en secondes et en ease (par exemple)
* Pour regrouper tous les éléments de style en JS, le mieux est d’ajouter une classe dans le JS et de définir ses propriétés dans le CSS
* Toggle : si la propriété n’existe pas, il l’ajoute, si elle existe il l’enlève
* On peut utiliser la propriété CSS transform : translate sur un axe X ou Y avec une valeur (nombre)
* Priorités : style injecté > div > id > class > balise HTML

Mouse events

* Pour créer un mousemove (cercle qui suit le curseur), on peut créer une div, lui donner des propriétés de bordure en CSS (en position absolue), puis on se positionne au dessus du dom (document), dans la fenêtre cette fois : window.
* On peut récupérer le paramètre e de l’évènement et voir les données de cet évènement en console sous forme d’un objet (on peut y accéder avec un point après le e en console log)
* Pour positionner un curseur (ex : un rond) : on lui injecte un top et un left en px (on peut aller chercher les données en pageX et pageY dans les données de la console avec une concaténation
* Répéter une ligne : Shift + Alt + Flèche du bas
* Pour centrer le curseur sur le centre du rond : transform-translate : (ex : -50%, -50%)
* Z-index peut permettre de faire passer un élement devant ou derrière un autre
* Mousedown : souris vers le bas ; Mouseup : souris vers le haut ; Mouseenter : quand le pointeur entre dans une autre zone , Mouseout : quand le pointeur sort d’une zone ; Mouseover : comme un Hover CSS mais en JS.

Key press event

* Un évènement sur une touche se fait généralement sur le document
* Pour savoir sur quelle touche on a appuyé : dans l’évènement, on va chercher l’évènement e avec un console log
* Pour entrer du texte dans la page, on pointe la balise voulue, puis on lui injecte un textContent qui vaut e.key.
* On peut donc dans cet évènement donner du style ou autre avec des conditions
* On peut ajouter un son en JS avec l’écriture suivante, puis en appelant la fonction :



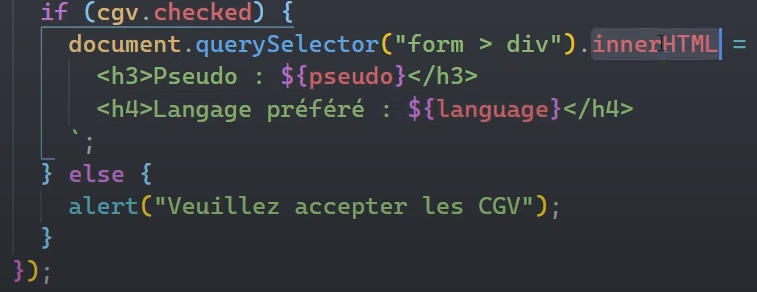
* Avec key appelé dans ring et une concaténation dans la constante audio.src = ( key + « .mp3 »), on peut appeler n’importe quel son avec le nom d’une lettre qui passe.

Scroll events

* Même comportement que les autres évènements comme mouse ou click

Form events ( à revoir )

* L’id de l’input doit être le même que le for du label
* Pour aller chercher un type d’input, on utilise la syntaxe : inputName = document .querySelector(‘input[type=’text’]’)
* L’input peut être un évènement comme le click, on peut en tirer les données depuis le console log avec un évènement, comme la value par exemple
* On peut stocker une valeur d’un évènement dans une variable pour l’utiliser ensuite (en allant chercher dans la console les propriétés contenues dans value par exemple)
* La méthode .preventDefault( ) permet dans un évènement de se prémunir contre le comportement par défaut du navigateur (ex : changement de page au Submit).
* On peut utiliser du innerHTML pour concaténer (voir la doc pour l’écriture)



Load event

* Permet de logger la page et de constater que la page a bien été entièrement chargée
* S’utilise avec window et non document

Les for Each

* On peut d’abord aller chercher les éléments voulus avec getElementsByClassName et les stocker dans une constante ou une variable.
* Le paramètre en entrée de boucle for each (par exemple boxes : ce sera box)
* GetElementsByClassName est moins pratique que le QuerySelectorAll

Projet Calculatrice

* PRENDRE DES NOTES ET FAIRE LE PROJET EN TEMPS REEL
* VIDEO 1 sur le CSS de From Scratch sur Youtube : Voir pour utiliser le grid