<https://codepen.io/nicolaspatschkowski/pen/dyorGbO>

let numberOfSeasons = 6;

let numberOfEpisodes = 12;

je crée 2 variables

let paragraph = document.querySelector('#info');

paragraph.innerText = `${numberOfSeasons} seasons, ${numberOfEpisodes} episodes per season`

je crée une variable paragraphe où j’importe la div info

je reprend la div info, j’ajoute innertext pour afficher du texte et des valeurs de variables

<https://codepen.io/nicolaspatschkowski/pen/jOPJWJO>

let numberOfSeasons = 6;

let numberOfEpisodes = 12;

let episodeTime = 45;

let commercialTime = 5;

j’ai 4 variables

let totalShowTime = (episodeTime + commercialTime) \* numberOfSeasons \* numberOfEpisodes;

je fait une operation de calcul avec ces variables pour en créer une autre

let paragraph = document.querySelector('#info');

paragraph.innerText = `${numberOfSeasons} seasons, ${numberOfEpisodes} episodes per season

Total viewing time: ${totalShowTime} minutes`

je reprend la div info, j’affiche du texte et des valeurs de variables que j’ai crée

<https://codepen.io/nicolaspatschkowski/pen/XWbGXjb>

const hoursPerDay = 24;

const minutesPerHour = 60;

const secondsPerMinute = 60;

je crée des constants de donnée

const dayInput = document.querySelector('#day-input');

const calculateButton = document.querySelector('#calculate-button');

const hours = document.querySelector('#hours');

const minutes = document.querySelector('#minutes');

const seconds = document.querySelector('#seconds');

j’importe les div correspondantes du html pour créer des constantes

calculateButton.addEventListener('click', () => {

let days = dayInput.value;

let calcHours = days \* hoursPerDay;

let calcMinutes = calcHours \* minutesPerHour;

let calcSeconds = calcMinutes \* secondsPerMinute;

quand je clique sur le bouton  « calculer », cela crée des variables à partir du nombre saisi pour faire un calcul de total heure, minutes, secondes

hours.innerText = `${calcHours} hours`;

minutes.innerText = `${calcMinutes} minutes`;

seconds.innerText = `${calcSeconds} seconds`;

});

Je reprends les const hours, minutes et secondes que j’ai crée et j’affiche du texte et les valeurs de variables que j’ai calculé

<https://codepen.io/nicolaspatschkowski/pen/YzXgyOG>

let episodeTitle = 'The First Battle';

let episodeDuration = 45;

let hasBeenWatched = false;

on a des variables dont un bolean

document.querySelector('#episode-info').innerText = `Episode: ${episodeTitle}

Duration: ${episodeDuration} min

${hasBeenWatched ? 'Already watched' : 'Not yet watched'}`

On récupère directement la div episode-info sans créer de variable (pas besoin) et on affiche du texte et des variables. Pour la variable false on pose la question et on met les deux réponses

<https://codepen.io/nicolaspatschkowski/pen/WNvmRRZ>

let episode = {

title: 'Dark Beginnings',

duration: 45,

hasBeenWatched: false

};

On crée un objet contenant des infos sur le film

let episodeTitle = episode.title;

let episodeDuration = episode.duration;

let episodeHasBeenWatched = episode.hasBeenWatched;

On crée des variables qui reprennent une seule info du film

document.querySelector('#episode-info').innerText = `Episode: ${episodeTitle}

Duration: ${episodeDuration} min

${episodeHasBeenWatched ? 'Already watched' : 'Not yet watched'}`

On affiche directement la partie épisode info de html avec les infos des variables crées

<https://codepen.io/nicolaspatschkowski/pen/qBdvRVW>

class Episode {

constructor(title, duration, hasBeenWatched) {

this.title = title;

this.duration = duration;

this.hasBeenWatched = hasBeenWatched;

}

}

On crée un modèle de film

let firstEpisode = new Episode('Dark Beginnings', 45, true);

let secondEpisode = new Episode('The Mystery Continues', 45, false);

let thirdEpisode = new Episode('The Unexpected Climax', 60, false);

A partir de ce modèle on crée des films avec les données correspondantes, plus court

document.querySelector('#first-episode-info').innerText = `Episode: ${firstEpisode.title}

Duration: ${firstEpisode.duration} min

${firstEpisode.hasBeenWatched ? 'Already watched' : 'Not yet watched'}`;

On affiche les données de chaque film à partir des modèles crées

document.querySelector('#second-episode-info').innerText = `Episode: ${secondEpisode.title}

Duration: ${secondEpisode.duration} min

${secondEpisode.hasBeenWatched ? 'Already watched' : 'Not yet watched'}`;

On affiche les données de chaque film à partir des modèles crées

document.querySelector('#third-episode-info').innerText = `Episode: ${thirdEpisode.title}

Duration: ${thirdEpisode.duration} min

${thirdEpisode.hasBeenWatched ? 'Already watched' : 'Not yet watched'}`;

On affiche les données de chaque film à partir des modèles crées

<https://codepen.io/nicolaspatschkowski/pen/NWqJdBy>

class Episode {

constructor(title, duration, hasBeenWatched) {

this.title = title;

this.duration = duration;

this.hasBeenWatched = hasBeenWatched;

}

}

On crée un modèle d’épisode d’un seul film

let firstEpisode = new Episode('Dark Beginnings', 45, true);

let secondEpisode = new Episode('The Mystery Continues', 45, false);

let thirdEpisode = new Episode('An Unexpected Climax', 60, false);

A partir de ce modèle on crée des épisodes avec les données correspondantes, plus court

let episodes = [firstEpisode, secondEpisode, thirdEpisode];

On crée un tableau avec ces épisodes et leurs données

const body = document.querySelector('body');

On récupère tout le body

for(let i = 0; i < 12; i++) {

let newDiv = document.createElement('div');

newDiv.classList.add('series-frame');

let newTitle = document.createElement('h2');

newTitle.innerText = 'The Story of Tau';

let newParagraph = document.createElement('p');

newParagraph.innerText = `${episodes[i].title}

${episodes[i].duration} minutes

${episodes[i].hasBeenWatched ? 'Already been watched' : 'Not yet watched'}`;

(comprend pas cette partie)

newDiv.append(newTitle);

newDiv.append(newParagraph);

body.append(newDiv);

(comprend pas cette partie)

}

On crée une boucle qui va analyser chaque ligne du tableau jusqu’à 12 lignes et créer en conséquence une div qui va rajouter des cadres pour chaque épisode du tableau (ici 3 i), ou chaque cadre aura le titre de la série et un paragraphe contenant du texte et les données de la class (titre, durée, regardée ou non)

<https://codepen.io/nicolaspatschkowski/pen/QWbodVr>

class Episode {

constructor(title, duration, hasBeenWatched) {

this.title = title;

this.duration = duration;

this.hasBeenWatched = hasBeenWatched;

}

}

On crée un modèle d’épisode d’un seul film

let firstEpisode = new Episode('Dark Beginnings', 45, true);

let secondEpisode = new Episode('The Mystery Continues', 45, false);

let thirdEpisode = new Episode('An Unexpected Climax', 60, false);

A partir de ce modèle on crée des épisodes avec les données correspondantes, plus court

let episodes = [];

episodes.push(firstEpisode, secondEpisode, thirdEpisode, thirdEpisode);

On rajoute à la fin un deuxième thirdEpisode

episodes.pop();

let numberOfEpisodes = episodes.length;

On demande le nombre de lignes dans le tableau

const body = document.querySelector('body');

On récupère tout le body

document.querySelector('#episodes').innerText = numberOfEpisodes;

On récupère le paragraphe “nombre episodes : et on lui ajoute comme texte le nombre de ligne du tableau

for(let episode of episodes) {

let newDiv = document.createElement('div');

newDiv.classList.add('series-frame');

let newTitle = document.createElement('h2');

newTitle.innerText = 'The Story of Tau';

let newParagraph = document.createElement('p');

newParagraph.innerText = `${episode.title}

${episode.duration} minutes

${episode.hasBeenWatched ? 'Already been watched' : 'Not yet watched'}`;

(comprend pas cette partie)

newDiv.append(newTitle);

newDiv.append(newParagraph);

body.append(newDiv);

(comprend pas cette partie)

}

On crée une boucle qui va analyser chaque ligne du tableau jusqu’à 12 lignes et créer en conséquence une div qui va rajouter des cadres pour chaque épisode du tableau (ici 3 i), ou chaque cadre aura le titre de la série et un paragraphe contenant du texte et les données de la class (titre, durée, regardée ou non)

<https://codepen.io/quentin-laurent/pen/ExNdMQb>

const espaceMessage = document.getElementById("message");

On récupère le message de réponse (vide)

const bouton = document.getElementById("bouton");

On récupère le bouton (vide)

const ageInput = document.getElementById("age");

On récupère le champ de saisie de l’âge (non rempli)

let age;

let ageMajorite = 18;

On défini la variable âge qu'on utilisera et la variable définissant l'âge de la majorité

function valider(){

espaceMessage.innerHTML = "Vous êtes autorisé à entrer";

}

Cette fonction affichera le message de validation

function refuser(){

alert("Cette espace est interdit aux personnes mineurs");

}

Cette fonction affichera un message d'erreur

function onConfirm(){

Cette fonction va servir à vérifier l’âge

age = parseInt(ageInput.value);

On récupère la saisie de l'âge et on transforme le texte en nombre entier

if(isNaN(age)){

alert("Ceci n'est pas un nombre");

return;

}

Si la saisie n'est pas un nombre, on affiche un message d'erreur

if(age < ageMajorite){

refuser();

}else{

valider()

}

Selon l’âge on applique la condition avec le message correspondant

ageInput.value = "";

}

On vide le champ de saisie après chaque click que sur le bouton commandé

bouton.addEventListener('click', onConfirm);

On écoute l'action de click sur le bouton et on appelle la fonction onConfirm

<https://codepen.io/quentin-laurent/pen/eYBQoMr>

const espaceMessage = document.getElementById("message");

On récupère le message de réponse (vide)

const bouton = document.getElementById("bouton");

On récupère le bouton (vide)

const ageInput = document.getElementById("age");

On récupère le champ de saisie de l’âge (non rempli)

const parentalInput = document.getElementById("parental");

On pointe sur l'élément de la checkbox de contrôle parental

let age;

let isControlParentalActive;

let ageMajorite = 18;

On défini la variable age qu'on utilisera et un variable définissant l'age de la majorité

function valider(){

espaceMessage.innerHTML = "Vous êtes autorisé à entrer";

}

Cette fonction affichera le message de validation

function refuser(){

alert("Cette espace est interdit aux personnes mineurs");

}

Cette fonction affichera un message d'erreur

function onConfirm(){

Cette fonction va servir à vérifier l’âge

age = parseInt(ageInput.value);

On récupère la saisie de l'age et on transforme le texte en nombre entier

if(isNaN(age)){

alert("Ceci n'est pas un nombre");

return;

}

Si la saisie n'est pas un nombre, on affiche un message d'erreur

isControlParentalActive = parentalInput.checked;

On récupère la valeur du checkbox de contrôle parentale. "true" si coché ou "false" si décoché

if(age < ageMajorite && isControlParentalActive){

refuser();

}else{

valider()

}

Selon l’âge on applique la condition avec le message correspondant

ageInput.value = "";

}

On vide le champ de saisie après chaque click que sur le bouton commandé

bouton.addEventListener('click', onConfirm);

On écoute l'action de click sur le bouton et on appelle la fonction onConfirm

<https://codepen.io/nicolaspatschkowski/pen/dyorvOv>

class Episode {

constructor(title, duration, hasBeenWatched) {

this.title = title;

this.duration = duration;

this.hasBeenWatched = hasBeenWatched;

}

}

On crée un modèle d’épisode d’un seul film

let episodes = [

new Episode('Dark Beginnings', 45, true),

new Episode('The Mystery Continues', 45, false),

new Episode('An Unexpected Climax', 60, false)

];

A partir de ce modèle on crée des épisodes avec les données correspondantes, plus court SOUS FORME D’UN TABLEAU

for (let episode of episodes) {

episode.hasBeenWatched = false;

}

On parcourt ce tableau pour changer les valeurs de hasbeenwatched en false

const body = document.querySelector('body');

for(let episode of episodes) {

let newDiv = document.createElement('div');

newDiv.classList.add('series-frame');

let newTitle = document.createElement('h2');

newTitle.innerText = 'The Story of Tau';

let newParagraph = document.createElement('p');

newParagraph.innerText = `${episode.title}

${episode.duration} minutes

${episode.hasBeenWatched ? 'Already been watched' : 'Not yet watched'}`;

(comprend pas cette partie)

newDiv.append(newTitle);

newDiv.append(newParagraph);

body.append(newDiv);

(comprend pas cette partie)

}

On crée une boucle qui va analyser chaque ligne du tableau et créer en conséquence une div qui va rajouter des cadres pour chaque épisode du tableau , ou chaque cadre aura le titre de la série et un paragraphe contenant du texte et les données de la class (titre, durée, regardée ou non)

<https://codepen.io/nicolaspatschkowski/pen/MWwxpvg>

const calculateAverageRating = (ratings) => {

if(ratings.length === 0) {

return 0;

}

on souhaite faire la moyenne des valeurs présentes dans le tableau   ratings  . Mais dans le cas où le tableau est vide, on souhaite retourner la valeur 0. La propriété   length  de   ratings  permet donc de connaître la longueur du tableau et de faire alors une condition en conséquence.

let sum = 0;

for (let rating of ratings) {

sum += rating;

}

Ce code a pour objectif de faire la somme de tous les éléments du tableau. Pour cela, on crée une variable   sum  avec comme valeur initiale 0. Ensuite on parcourt le tableau   ratings  et pour chaque élément on additionne la valeur avec la valeur de   sum  et le résultat est stocké dans   sum  . Une fois l’ensemble de ce code exécuté,   sum  contient donc la somme de toutes les valeurs du tableau.

return sum / ratings.length;

Nous disposons maintenant de la variable   sum  qui contient la somme de tous les éléments du tableau, et de l’expression   ratings.length  qui donne le nombre d'éléments du tableau. Avec ces éléments, nous retournons le calcul de la moyenne.

}

const tauRatings = [5, 4, 5, 5, 1, 2];

const colinRatings = [5, 5, 5, 4, 5];

notes des épisodes

const tauAverage = calculateAverageRating(tauRatings);

const colinAverage = calculateAverageRating(colinRatings);

if (tauAverage && colinAverage) {

document.querySelector('#tau-score').innerText = tauAverage.toFixed(1) + ' stars';

document.querySelector('#colin-score').innerText = colinAverage.toFixed(1) + ' stars';

document.querySelector('#clara-score').innerText = `${calculateAverageRating([]) === 0 ? 'No ratings' : calculateAverageRating([]) + ' stars'}`

}