

Starter app

<https://github.com/Orodan/EC-starter-app>

EX 1 - VARIABLE

- Créer une variable contenant une chaîne de caractères
- Y affecter un chiffre
- Créer une deuxième variable contenant un chiffre
- Additionner les deux chiffres
- Afficher le résultat dans la console

EX 2 - IF

- Créer une variable vide
- Utiliser un 'if' pour afficher dans la console 'ma variable est positive' uniquement si la variable a une valeur supérieure à 0
- Modifier la définition de votre variable pour l'initialiser avec une valeur positive
- Rajouter un else pour afficher dans la console 'ma variable est négative' uniquement si la variable a une valeur inférieure à 0
- Modifier la définition de votre variable pour l'initialiser avec une valeur négative
- Rajouter un else pour afficher dans la console 'ma variable est égale à 0' uniquement si la variable a une valeur égale à 0
- Modifier la définition de votre variable pour l'initialiser avec une valeur à 0

EX 3 - TABLEAUX

- Créer un tableau contenant 'a', 'b', 'c'
- Parcourez ce tableau et afficher un à un ses éléments
- Rajouter 'd' à la fin du tableau
- Parcourez ce tableau et afficher un à un ses éléments
- Retirer le premier élément du tableau
- Parcourez ce tableau et afficher un à un ses éléments
- Retirer l'élément à l'index 1 du tableau

- Parcourez ce tableau et affichez un à un ses éléments

EX 4 - FONCTIONS

- Reprenez le code de l'exercice précédent.
- Définissez une fonction 'display' qui prends en paramètre un tableau, le parcourt et affiche un à un ses éléments.
- Remplacer les boucles for par l'appel à la fonction 'display'

EX 5 - OBJETS

- Écrivez le code, une ligne pour chaque action :
- Créer un objet vide user.
- Ajoutez la propriété name avec la valeur John.
- Ajoutez la propriété surname avec la valeur Smith.
- Changer la valeur de name pour Pete.
- Supprimez la propriété name de l'objet.

EX 6 : Somme des propriétés de l'objet

Nous avons un objet stockant les salaires de notre équipe : let salaries = { John: 100, Ann: 160, Pete: 130 }

- Écrivez le code pour additionner tous les salaires et les enregistrer dans la variable sum. Devrait être égale à 390 dans l'exemple ci-dessus.
- Si salaries est vide, le résultat doit être 0.

Ex 7 : Multipliez les valeurs de propriétés numériques par 2

- Créez une fonction multiplyNumeric(obj) qui multiplie toutes les valeurs de propriétés numériques de obj par 2.

Par exemple : // before the call let menu = { width: 200, height: 300, title: "My menu" };

multiplyNumeric(menu);

// after the call menu = { width: 400, height: 600, title: "My menu" };

- Veuillez noter que multiplyNumeric n'a pas besoin de retourner quoi que ce soit. Il devrait modifier l'objet en place.

Ex 8 : Vérifier le vide

- Ecrivez la fonction isEmpty(obj) qui renvoie true si l'objet n'a pas de propriétés, sinon false. Devrait fonctionner comme ça : let schedule = {};

```
isEmpty(schedule) // true
```

```
schedule["8:30"] = "get up";
```

```
isEmpty(schedule) // false
```

Ex 9 : Créer une calculatrice

Créer une class Calculator avec 3 méthodes :

- sum(a, b) renvoie la somme de ces propriétés.
- multiply(a, b) renvoie le produit de la multiplication de ces propriétés.
- getLastValue() renvoie le dernier résultat produit

Par exemple : let calculator = new Calculator();

```
calculator.sum(1, 2) // 3 calculator.multiply(2, 3) // 6 calculator.getLastValue() // 6
```

Ex 10 : Créer un accumulateur

Créer une classe Accumulator avec une propriété 'value' et une méthode 'read'.

- La propriété value sera initialisée par la valeur du premier argument du constructeur.
- Par défaut cette valeur sera à 0 si aucun argument n'est passé au constructeur.
- La méthode read() devrait utiliser 'prompt' pour lire un nouveau numéro et l'ajouter à value.

En d'autres termes, la propriété value est la somme de toutes les valeurs entrées par l'utilisateur avec la valeur initiale startingValue.

```
let accumulator = new Accumulator(1); // valeur initiale 1
```

```
accumulator.read(); // ajoute la valeur entrée par l'utilisateur accumulator.read(); // ajoute la valeur  
entrée par l'utilisateur
```

```
console.log(accumulator.value); // affiche la somme de ces valeurs
```

Ex 11 : Récupérer le premier élément

Créer un paragraphe dans votre page

```
<p>Ceci est mon premier paragraphe</p>
```

Afficher dans la console le contenu du premier paragraphe

Ex 12 : Récupérer un élément précis

Créer deux paragraphes dans la page

```
<p>Ceci est mon premier paragraphe</p>
<p>Ceci est mon second paragraphe</p>
```

Afficher dans la console le contenu du second paragraphe

Exo 13 : Cacher au clique

Créer une div avec l'id 'text', une hauteur et largeur de 50px et un fond rouge sur votre page. Créer une bouton 'Cacher' sur votre page.

Ajouter du javascript pour cacher la div avec l'id 'text' au clique sur le bouton.

Ex 14

Créer un paragraphe

```
<p>Je change de couleur quand la souris me survol</p>
```

Changer la couleur de fond du paragraphe au survol de la souris Remettre la couleur de fond initiale lorsque la souris quitte le paragraphe Faire de même avec la couleur du texte du paragraphe

Ex 15 : calculateur de prix

Créer un champ prix de type 'number', avec le label 'prix'. Créer un champ quantité de type 'number' avec le label 'quantité'. Créer une paragraphe avec l'id 'total' et aucun contenu.

Ajouter du javascript pour:

- récupérer les valeurs de prix et quantité renseignées par l'utilisateur
- multiplier les deux valeurs récupérées pour obtenir le total
- afficher la valeur du total dans le paragraphe portant l'id 'total'

Exo 16 : todos

Créer une bouton contenant le texte 'Ajouter un todo' Créer une div portant l'id 'todos', avec une couleur de fond de la couleur de votre choix Créer une fonction addTodo permettant de rajouter une nouvelle

`<div class=« todo »>`Ceci est un todo au sein de la div portant l'id todos. Déclencher cette fonction à chaque click sur le bouton 'Ajouter un todo'

Si vous en êtes arrivé là, vous avez terminé tous les exercices prévues pour la partie JS de ce cours. Félicitations ! N'hésitez pas à rajouter des fonctionnalités à votre app 'todos'. Voici quelques idées:

- Pouvoir préciser le texte du todo au moment de son ajout - avec un formulaire par exemple
- Pouvoir supprimer un todo
- Pouvoir éditer un todo