

EVAL ALGO JS

Exo 1

Demandez un mot à l'utilisateur.

Afficher le nombre de lettres du mot ainsi que chaque lettre.

Exo 2

Demandez une phrase à l'utilisateur

Afficher le nombre de mots dans cette phrase. Les mots sont séparés par des espaces ou une apostrophe.

Exo 3

Demandez le score de 4 candidats à une élection. Vous êtes le premier candidat.

Si le total des 4 candidats n'est pas égal à 100, indiquez l'erreur dans un alert et recommencez la demande de score.

Si le total est égal à 100 :

Vous êtes élu si votre score est supérieur ou égal à 50.

Vous êtes battu si votre score est inférieur à 12.5 ou que l'un des autres candidats recueille plus de 50.

Dans les autres cas vous êtes en ballottage favorable si vous avez le meilleur score des 4 candidats sinon vous êtes en ballottage défavorable.

Afficher le résultat.

Exo 4

Demander le montant des achats de l'utilisateur (un entier positif)

Demander la somme qu'il donne (un entier positif)

Recommencer en prévenant dans un alert si la somme donnée est inférieure au montant des achats.

Sinon :

Indiquez si le compte est juste

Sinon

Indiquer combien vous lui rendez en billet de 10€, billet de 5€ et pièces de 2[€] et 1€.

Indiquer uniquement le rendu nécessaire (ex : pas de -> 0 billet de 10€).

Réaliser les tests nécessaires pour vérifier votre code.

Exo 5

Soit ces 6 variables :

```
const a = 7;  
const b = 3.14574;  
const c = 9.9999999;  
let d;  
let e = null;  
let f = "John Doe";
```

Créer une fonction qui prend en paramètre une variable et qui retourne soit :

- Que cette variable n'est pas un nombre
- Que cette variable est un entier
- Que cette variable est un décimal (que vous allez arrondir à 2 chiffres après la virgule)

Tester cette fonction avec les 6 variables. Le résultat doit être le suivant :

```
7 est un entier  
3.15 est un décimal arrondi  
10 est un décimal arrondi  
undefined n'est pas un nombre  
null n'est pas un nombre  
John Doe n'est pas un nombre
```

Exo 6 :

Créer une fonction qui prend en paramètre un nombre de secondes et retourner la durée du film en heures, minutes et secondes. Ne pas afficher lorsque l'une des valeurs est égale à 0.

Exo 7 :

Demander à l'utilisateur, la saisie de 5 chiffres à la fois. Lui indiquer à la fin le plus grand.

Exo 8 :

A partir de l'objet ci-dessous.

Vous devez modifier le 3eme acteur.

Ajouter 5 likes au film avec une boucle, afficher la note SensCritique et ne plus avoir vu le film

```
const movie = {
  title: 'Fight Club',
  actors: ['Brad Pitt', 'Edward Norton', 'Helena Bonham Carter'],
  director: 'David Fincher',
  year: 1999,
  duration: 139,
  rating: {
    imdb: 8.8,
    sensCritique: 8.2,
    allocine: 7.2
  },
  numberLike: 16503,
  like: function() {
    movie.numberLike++;
  },
  unlike: function() {
    movie.numberLike--;
  },
  seen: true
}
```

Exo 9 :

A partir de ce tableau d'objets, créer une fonction qui calcule la moyenne pondérée en utilisant reduce.

```
const students = [
  { name: "Alice", notes: [20, 15, 9.5], coeff: [3, 3, 12] },
  { name: "Bob", notes: [10, 7, 15], coeff: [3, 7, 5] },
  { name: "Charlie", notes: [12, 13, 14], coeff: [2, 4, 10] },
];
```

Exo 10 :

A partir de ce tableau, avec l'utilisation de filter, ne récupérer que les personnes qui aime marcher, puis afficher pour ces dernières un message de bienvenue du type 'Hello nom_de_la_personne' !

```
const persons = [
  { nom: 'Alice', age: 25, aimeMarcher: true },
  { nom: 'Bob', age: 30, aimeMarcher: false },
  { nom: 'Charlie', age: 35, aimeMarcher: true },
  { nom: 'Diana', age: 40, aimeMarcher: false },
  { nom: 'Edward', age: 20, aimeMarcher: true }
];
```