



1. Travaux dirigés :

Quiz de révision et prérequis :

- 1) A l'intérieur d'une page HTML ou on met Java script
 - a. <js>
 - b. <script>
 - c. <javascript>
- 2) Quelle est la vraie syntaxe pour changer le contenu de l'élément suivant
test
 - a. document.getElementById('span1').innerHTML='salut monde'
 - b. document.getElement('span1').innerHTML='salut monde'
 - c. document.getElementByName('span1').innerHTML='salut monde'
 - d. document.getElementById('#span1').innerHTML='salut monde'
- 3) Quelle est l'emplacement de Java Script
 - a. <head>
 - b. <body>
 - c. Les deux
- 4) Quelle la bonne syntaxe pour faire reference à un script externe script1.js
 - a. <script name='script1.js'>
 - b. <script href='script1.js'>
 - c. <script src='script1.js'>
- 5) Un script externe doit contenir le tag <script>
 - a. Oui
 - b. Non
- 6) Comment créer une fonction en javascript
 - a. function=function1()
 - b. function function1()
 - c. function=>function1
- 7) Comment écrire la condition if
 - a. if x==7 then
 - b. if x=7
 - c. if(x==7)
- 8) Comment écrire la condition x n'est pas égal à 7
 - a. if(x<>7)
 - b. if x= !7
 - c. if x<>7
 - d. if (x !=7)
- 9) La bonne syntaxe de la boucle while
 - a. while k=1 to 20
 - b. while (k<=20)
 - c. while(k<=20 ;k++)



10) La bonne syntaxe de for

- a. for j=1 to 10
- b. for(j<=10 ;j++)
- c. for(j=0 ;j<=10 ;j++)
- d. for(j=0 ;j<=10)

11) Quelle est syntaxe correcte pour créer un Array

- a. var fruits= 'pomme','orange','banane'
- b. var fruits= ['pomme','orange','banane']
- c. var fruits=(1 : 'pomme',2 : 'orange',3 : 'banane')

12) Quel évènement est déclenché si on clique sur un élément HTML

- a. onclick
- b. onmousedown
- c. onmouseover
- d. onchange

13) Est-ce que javascript est sensible à la casse

- a. oui
- b. non

14) Soit

```
let numbers=[3,7,2,5,9,4,10]  
console.log(numbers.filter(n=>n>=5))
```

Ce code affiche :

- a. 5
- b. Erreur
- c. 7,5,9,10
- d. [7,5,9,10]

15) Soit

```
let numbers=[3,7,2,5,9,4,10]  
console.log(numbers.find(n=>n>=5))
```

Ce code affiche :

- a. 5
- b. 7
- c. [7,5,9,10]
- d. 7,5,9,10
- e. undefined

16) Soit

```
let numbers=[3,7,2,5,9,4,10]  
console.log(numbers.find(n=>n==6))
```

Ce code affiche :

- a. 5
- b. 7
- c. [7,5,9,10]
- d. Undefined



17) Soit

```
let numbers=[3,7,2,5,9,4,10]
console.log(numbers.map(n=>n*2))
```

Ce code affiche :

- a. 6
- b. 6,14,4,10,18,8,20
- c. [6,14,4,10,18,8,20]
- d. erreur

18) Soit

```
let numbers=[3,7,2,5,9,4,10]
console.log(numbers.reduce(function(total,n){return total+n},0))
```

Ce code affiche :

- a. 30
- b. 40
- c. Erreur
- d. undefined

19) soit le code suivant :

```
function salutation(nom, presentation){ presentation(nom) }
function hello(nom){
  console.log(`hello ${nom.toUpperCase()}`)
}
function salut(nom){
  console.log(`salut ${nom.toUpperCase()}`)
}
salutation("rami",hello)
salutation("fahmi",salut)
```

Ce code affiche :

- a. Rami hello
fahmi salut
- b. hello RAMI
salut FAHMI
- c. erreur
- d. undefined

20) soit le code suivant :

```
let cmpt={second:0}
const incremente={()=>{cmpt.second++;if(cmpt.second==60) cmpt.second=0;
console.log(cmpt.second)}}
const compteur=setInterval(incremente,1000)
```

- a. affiche des valeurs de 1 à 1000
- b. affiche des valeurs de 1 à 59 puis recommence de 0 à 59 plusieurs fois
- c. erreur
- d. undefined

Les aspects avancés de JavaScript

Exercice 1 : Manipulation des classes

Manipulation des classes et objets en javascript ES6

- Créer la classe Document contenant les attributs id et dateEdition
- Créer la classe Livre qui hérite de la classe Document contenant en plus les attributs titre et auteur.
- Créer deux instances de Document
- Créer trois instances de Livre
- Créer la méthode infoLivre qui retourne les informations du livre exemple :

```
Livre: id:10 titre:P00 Auteur: Rami date Edition:10/05/2020
```

Exercice 2 : Manipulation des classes

Créer une application appelée RestfulReading (Lecture reposante), qui permettra aux utilisateurs de suivre les livres qu'ils sont en train de lire ou qu'ils ont lus.

Votre tâche est de créer la classe Book (Livre) et de remplir la base de données de développement de trois ou quatre livres.

La classe Book doit contenir les champs suivants :

- title - string - le titre du livre
- author - string - l'auteur du livre
- description - string - une description du livre
- pages - number - le nombre total de pages
- currentPage - number - la page où se trouve l'utilisateur actuellement (entre 1 et pages)
- read - boolean - si l'utilisateur a lu ou non le livre (par défaut: false)

La classe Book doit aussi contenir la méthode instance suivante : readBook(page). Cette instance permet à l'utilisateur de dire à quelle page il se trouve actuellement

- Si l'argument page est inférieur à un ou supérieur au nombre total de pages du livre, readBook retourne dans la console 0
- Si l'argument page est supérieur ou égal à 1 et inférieur au nombre total de pages du livre, readBook modifie le champ currentPage de l'instance pour être égal à la valeur de l'argument passé, et retourne dans la console 1
- Si l'argument page est égal au nombre total de pages du livre, readBook modifie le champ currentPage de l'instance pour être égal à la valeur de l'argument passé, modifie le champ read de l'instance en true, et retourne dans la console 1

Quand vous aurez créé la classe Book, vous en créerez au moins trois instances valables et vous les mettrez dans le tableau books. Affichez un de vos livres dans la console.

Utilisez l'instance de classe readBook en lui donnant un argument inférieur au nombre de page de ce livre (cela doit afficher dans la console "En cours de lecture") et testez la

classe `readBook` en lui donnant un argument égale au nombre de pages, ce qui doit afficher "Lu".

Exercice 3 : Manipulation des Array

Soit le tableau suivant : `const input = [1, 2, 3, 4, 5]`

Utiliser la méthode **map** pour créer un tableau contenant le carré des éléments de `nums`

Résultat `[1, 4, 9, 16, 25]`

Exercice 4 :

Soit le tableau suivant : `const input = [1, -2, 3, 4, -5]`

Utiliser la méthode **filter** pour créer un tableau contenant les éléments positifs de `nums`

Résultat `[1, 3, 4]`

Utiliser la méthode **reduce** pour calculer la somme des éléments de `nums`

Résultat `-1`

Utiliser la méthode **filter** et **reduce** pour calculer la somme des éléments positifs de `nums`

Résultat `8`

Exercice 5 : Manipulation des Array

Soit

```
let livres=[{id :10,titre : 'POO',auteur : 'RAMI',prix :300},  
{id :11,titre : 'JS ES6',auteur : 'FAMI',prix :230},  
{id :12,titre : 'Algorithmes',auteur : 'KARIMI',prix :330},  
{id :13,titre : 'HTML& CSS',auteur : 'RAMI',prix :340}]
```

1. Créer la liste titres contenant tous les titres à partir de `Livres` utiliser `map`
2. Créer la liste titres contenant tous les livres de l'auteur 'RAMI' à partir de `Livres` utiliser `filter`
3. Chercher le livre dont le id égal à 12 utiliser `find`
4. Calculer le total des prix utiliser `reduce`
5. Créer un Array `mesLivres` copie de `livres`



Exercice 6 : Programmation fonctionnelle

On souhaite créer un compteur timer qui on peut l'incrémenter et décrémenter

heure:10
minute:23
second:13

Le bouton incrémenter permet d'incrémenter les seconds

Le bouton décrémenter permet de décrémenter les seconds

Le bouton arrêter permet de stopper le compteur

Travail à faire

1. En utilisant le code Html et javascript ci-dessous tester le l'incrémentation et l'arrêt
2. Donner le code pour le bouton décrémenter

Page index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
</head>
<body>
  <label>heure:</label><span id="spHeure"></span><br/>
  <label>minute:</label><span id="spMinute"></span><br/>
  <label>second:</label><span id="spSecond"></span><br/>
  <input type="button"
onclick="incrémenterTimer()" value="incrémenter"/>
  <input type="button" value="décrémenter"/>
  <input type="button" onclick="stopTimer()" value="arrêter"/>
  <script src="compteur.js" ></script>
</body>
</html>
```



Fichier compteur.js

```
spanHeure=document.getElementById("spHeure")
spanMinute=document.getElementById("spMinute")
spanSecond=document.getElementById("spSecond")
let initTime={heure:10,minute:23,second:5}
let currentTime=initTime
let timer=null
function incrementer(temps){
    time={...temps}
    return function(){
        time.second++
        if(time.second>=60){
            time.second=0
            time.minute++
            if(time.minute>=60){
                time.minute=0
                time.heure++
            }
        }
        affiche(time)
        currentTime=time
        return time
    }
}
function affiche(time){
    spanHeure.innerHTML=time.heure;
    spanMinute.innerHTML=time.minute;
    spanSecond.innerHTML=time.second;
}
function stopTimer() {
    if(timer){
        clearInterval( timer);
    }
    timer = null;
}
function incrementerTimer() {
    stopTimer()
    let next=incrementer(currentTime)
    timer=setInterval(next,1000);
}
```