

Les fonctions

Exercices

Notions de base

Exercice 1 – appel de fonctions

1 Voici une fonction qui affiche un message à l'utilisateur :

```
function bonsoir(){  
  console.log("bonsoir");  
}
```

Écrire l'instruction qui appelle cette fonction. Tester.

2 Voici une fonction qui affiche un message à l'utilisateur :

```
function bonjour(nom){  
  console.log('bonjour ' + nom);  
}
```

Écrire l'instruction qui affichera « bonjour Robin », en faisant appel à cette fonction.

3 Voici une fonction qui affiche la date d'aujourd'hui :

```
function date(){  
  let now = new Date() ;  
  console.log(now.toString());  
}
```

Écrire l'instruction qui appelle cette fonction. Tester.

4 Voici une fonction qui retourne le double d'un nombre :

```
function double(n){  
  return 2 * n ;  
}
```

Écrire l'instruction qui affiche à l'utilisateur le double de 256, en appelant cette fonction.

1.4 Voici une fonction qui retourne la multiplication de 2 nombres :

```
function multiplier(p1,p2){  
    return p1 * p2 ;  
}
```

Écrire l'instruction qui affiche à l'utilisateur le résultat de la multiplication de 5 par 6, en appelant cette fonction.

Voici une fonction qui converti des degrés Celsius en degrés Farenheit:

```
function conversion(valeur){  
    return valeur * 1.8 + 32 ;  
}
```

Écrire l'instruction qui affiche à l'utilisateur la conversion de 25° Celsius en Farenheit.

1.5 Voici une fonction qui convertit une couleur hexadécimale en code RGB :

```
function toRgb(hex) {  
    hex = hex.slice(1);  
    let r = Number.parseInt(hex.slice(0, 2), 16);  
    let g = Number.parseInt(hex.slice(2, 4), 16);  
    let b = Number.parseInt(hex.slice(4, 6), 16);  
    return [r,g,b];  
}
```

Quel type de données retourne cette fonction : Number, String ou Array ?

Écrire l'instruction qui affiche à l'utilisateur les composantes RGB de la couleur "#FF0000".

Syntaxe de fonction fléchée

Indiquer si ces définitions de fonctions sont syntaxiquement correctes

```
const subtract = (x, y) => { return x - y};
```

```
const add = (x, y) => x + y;
```

```
const doIt = () => console.log('test')
```

Exercice 2 – définition de fonctions

2.1 Écrire une fonction qui retourne la surface d'un cercle, en fonction de son rayon. Utiliser la constante prédéfinie Math.PI

(https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/Math).

2.2 Écrire une fonction qui retourne la surface d'un carré, en fonction de la longueur de son côté.

2.3 Écrire une fonction qui retourne la surface d'un rectangle, en fonction de la longueur de ses côtés.

2.4 Écrire une fonction qui prend en paramètre une chaîne de caractère et qui retourne la même chaîne en minuscule excepté la première lettre en majuscule.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/substring
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/toUpperCase

2.5 Écrire une fonction qui prend en paramètre deux chaînes de caractère et qui retourne une seule chaîne concaténée.

Exercices d'entraînement

Exercice 3 : passer un tableau en paramètre

Écrire une fonction qui prend en paramètre un tableau et qui retourne le nombre d'éléments du tableau.

Exercice 4 : extraire une fonction d'un algorithme

Voici un algorithme qui calcule et affiche le prix d'un nombre de photocopies en appliquant le tarif suivant : les 10 premières photocopies coûtent 0,50€/unité, les 20 suivantes coûtent 0,30€/unité, toutes les autres 0,20€/unité.

```
let t1 = 0.5;
let t2 = 0.3;
let t3 = 0.2;
let q1 = 10;
let q2 = 20;
let p;
let n;

p=0;
n = prompt(« Nombre de copies ? »);
if(n<q1){
  p=n*t1;
}
else if(n<q1+q2){
  p=t1*q1+t2*(n-q1);
}
else{
  p=t1*q1+t2*q2+t3*(n-q2-q1);
}
console.log('tarif :', p) ;
```

Transformer cet algorithme en fonction. Cette fonction prendra en paramètre le nombre de photocopies. La valeur de retour sera le prix.

Exercice 5 : Contrôler les paramètres d'une fonction

Voici une fonction qui prend en paramètre une couleur au format hexadécimal (de la forme #FF0000). Cette fonction retourne la couleur complémentaire.

```
function couleurComplementaire(hex) {
  hex = hex.slice(1);
  let r = ('00' + (255 - parseInt(hex.slice(0, 2), 16)).toString(16)).slice(-2);
  let g = ('00' + (255 - parseInt(hex.slice(2, 4), 16)).toString(16)).slice(-2);
  let b = ('00' + (255 - parseInt(hex.slice(4, 6), 16)).toString(16)).slice(-2);
  return '#' + r + g + b;
}
```

4.1 Écrire l'instruction qui affiche à l'utilisateur la couleur complémentaire des couleurs #FF0000, #26A69A et #D95965.

4.2 Écrire l'instruction qui demande à l'utilisateur de saisir une couleur (au format hexadécimal) et qui affiche à l'écran la couleur complémentaire.

4.3 Modifier la fonction de façon à s'assurer que la couleur soit correctement formatée. Si la couleur n'est pas bien formatée, la fonction devra retourner «false».

Le format est correct si :

- la chaîne comprend 7 caractères,
- le premier caractère est un #,
- les 6 autres caractères sont des chiffres (entre 0 et 9) ou des lettres (entre A et F).

4.4 Modifier également votre programme pour qu'il demande à l'utilisateur de renouveler la saisie tant que la couleur n'est pas bien formatée.

Exercice 6 : appeler une fonction dans un programme

Créer une fonction `getExtension()` . Cette fonction prend en paramètre un nom de fichier (ex : "chat.jpg") et elle retourne l'extension (ex : ".jpg").

Conseil : utiliser les méthodes `indexOf()` et `slice()`

Vérifier le code avec les valeurs suivantes : "screenshot.png", "icon.svg", "index.html", "attestation.pdf"

Exercice 7 : appeler une fonction dans un programme

A. Ecrire une fonction `isValid()` qui prend en paramètre une chaîne de caractère, qui retourne `true` si cette chaîne contient un « @ » et qui retourne `false` dans le cas contraire.

B. Ecrire un programme qui demande à l'utilisateur de saisir une adresse email, qui vérifie que l'adresse est valide puis l'enregistre dans un tableau appelé « mailingList » et qui affiche à l'utilisateur le message « Votre adresse est enregistrée ».

Exercice 8 : appeler une fonction dans un programme

A) Écrire une fonction :

- Cette fonction prend en paramètre un nombre.
- Cette fonction teste si le nombre est supérieur ou égal à zéro et retourne «true». Sinon, elle retourne «false».

B) Demander à l'utilisateur de saisir un nombre positif. Vérifier la saisie en utilisant la fonction.

C) Demander à l'utilisateur de saisir un nombre (positif ou négatif). Le programme affiche le nombre en console et invite l'utilisateur à saisir un nouveau nombre. La boucle s'arrête lorsque l'utilisateur saisi STOP. Dans ce cas, le programme affiche combien de nombres positifs on été saisis.

Exercice 9 : appeler des fonctions dans un programme

A) Écrire une fonction :

- Cette fonction prend en paramètre un tableau de notes d'un bulletin scolaire.
- Cette fonction retourne la moyenne des notes.

B) Écrire une fonction :

- Cette fonction prend en paramètre 1 nombre.
- Cette fonction teste si le nombre est supérieure ou égale à 10 et retourne « Succès ». Sinon, elle retourne «Echec ».

C) Demander à l'utilisateur de saisir plusieurs nombres et tester, grâce à ces 2 fonctions si la moyenne des notes est supérieure à zéro. Afficher le résultat.

Fonctions de rappel

Exercice 10 : Manipulation de données

Récupérer le fichier adresses.js à cette adresse : https://drive.google.com/file/d/1ISMTxLgBEGT_-0Ze1D4ozoNu0W0jwr5P/view?usp=sharing

1. Afficher toutes les adresses avec une boucle
2. Déterminer si l'adresse shayla5@yahoo.com est déjà dans le tableau
3. trier les adresses par ordre alphabétique
4. trouver les adresses invalides (arobase manquant)

Exercice 11 : Compte à rebours

Afficher un compte à rebours dans la console. Arrivé à 0, afficher « GO »