

TP n°4 : les fonctions**Exercice 1 – appel de fonctions**

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

```
function bonsoir(){  
    sortir("bonsoir");  
}
```

Écrire l'instruction qui appelle cette fonction. Tester.

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

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

Écrire l'instruction qui appelle cette fonction. Tester.

1.3 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.

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 ?

NB : la documentation sur la méthode slice() est consultable à l'adresse

https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/String/slice

et celle de parseInt() à l'adresse

https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/Number/parseInt

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

Vous venez de manipuler des fonctions :

- avec 0, 1 ou 2 paramètres : une fonction peut avoir plusieurs paramètres formels,
- qui ne retourne aucune donnée ou qui retourne une donnée : une fonction ne peut pas retourner plus d'une donnée.

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.

Exercice 3

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 = entrerNB();
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);
}
sortir(p) ;

```

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

Exercice 4 :

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

```

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 (au format RGB).

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 "erreur". Tester.

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.

Rappel : 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).