



EXERCICES JAVASCRIPT

PARTIE 1 : BASE

ÉCRITURE D'UNE FONCTION SIMPLE

Vous utiliserez les fonctions `alert`(qui affiche une chaîne de caractères) et `prompt` (qui invite à la saisie d'une donnée), dont voici les syntaxes :

Modifier le code html ci-dessous, pour que le `click` sur le bouton lance une fonction.

Cette fonction demande la saisie d'une largeur, d'une longueur et affiche la surface du rectangle correspondant.

- Remplacez ce calcul par celui du périmètre (double de la somme de la longueur et de la largeur).

```
<html>
<head>
  <title>Écriture d'une fonction simple</title>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <script type="text/javascript">
    // <!--

    // -->
  </script>
</head>
<body>

<h1>Écriture d'une fonction simple</h1>

<p>Cliquez sur le bouton pour lancer la fonction&nbsp;: <button onclick="">Lancer la
fonction</button></p>

</body>
</html>
```

ÉCRITURE D'UNE FONCTION RENVOYANT UNE VALEUR

Appel à une fonction retournant une valeur.

1. Créez une fonction qui :

- demande la saisie d'un rayon ;
- retourne la surface du cercle de ce rayon

2. Affichez le résultat de l'appel à cette fonction en cliquant sur le bouton.

```
<html>

<head>
  <title>Écriture d'une fonction renvoyant une valeur</title>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <script type="text/javascript">
    // <!--

    // -->
  </script>
</head>
<body>

  <h1>Écriture d'une fonction complexe</h1>

  <p>Cliquez sur le bouton pour lancer la fonction&nbsp;  ; <button onclick="">Lancer la
  fonction</button></p>

</body>
</html>
```

ÉCRITURE D'UNE FONCTION COMPLEXE AVEC ARGUMENTS

Appel à une fonction retournant une valeur

1. Créez deux variables globales a et b, initialisées respectivement à 3 et à 2.
2. Créez une fonction multiplie prenant un argument x prenant comme valeur par défaut 8, et renvoyant le résultat de la multiplication de x par 3
3. Créez une fonction affiche, appelée au clic sur le bouton, qui affiche dans des boîtes d'alerte successivement le résultat de multiplie appliquée a, à b, puis sans aucun paramètre (en exécutant donc la fonction avec la valeur de x par défaut)

```
<html lang="fr">

<head>
  <title>Écriture d'une fonction avec arguments</title>
  <meta charset="UTF-8">
  <script>

    </script>
</head>
<body>

  <h1>Écriture d'une fonction avec arguments</h1>

  <p>Cliquez sur le bouton pour lancer la fonction&nbsp;: <button onclick="">Lancer la
fonction</button></p>

</body>
</html>
```

UTILISATION D'UN TABLEAU

Appel à une fonction retournant une valeur

1. Créez un tableau nommé tab dont le premier élément est -2, le deuxième 1 et le troisième 4
2. Créez une fonction additionne prenant un argument x et renvoyant le résultat de l'addition de x à 2
3. Créez une fonction affiche, appelée au clic sur le bouton, qui affiche dans des boîtes d'alerte successivement le résultat de additionne appliqué au premier élément, puis au dernier élément du tableau (en utilisant la propriété length).

```
<html>

<head>
  <title>Utilisation d'un tableau</title>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <script type="text/javascript">
    // <!--

    // -->
  </script>
</head>
<body>

  <h1>Utilisation d'un tableau</h1>

  <p>Cliquez sur le bouton pour lancer la fonction ; <button onclick="">Lancer la
  fonction</button></p>

</body>
</html>
```

MÉTHODES DES TABLEAUX

1. Définissez le tableau fibo initialisé à [1,1,2,3,5,8] et affichez-le dans la console
2. Ajoutez 13 à la fin du tableau et 0 au début, et affichez-le dans la console
3. Remplacez chaque élément du tableau par le reste de sa division par 4, stockez le résultat dans la variable fibo2 et affichez-le dans la console
4. Triez le tableau par ordre croissant des nombres et affichez-le dans la console

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8" />
<title>Méthodes des tableaux</title>
<script>
function go(){

}
</script>
</head>
<body>

<button onclick="go()">Lancement</button>

</body>
</html>
```

BOUCLES FOR

1. Au click sur le premier bouton, lancez la fonction `boucle()` qui crée un tableau de 3 éléments et utilise une boucle `for` pour le remplir de sorte que l'élément `i` du tableau contienne i^2 .
2. Affichez le tableau dans une boîte d'alerte.
3. Au click sur le second bouton, lancez la fonction `boucle2()` qui demande à l'utilisateur via un prompt de saisir la longueur souhaitée du tableau, puis crée ce tableau et utilise une boucle `for` pour le remplir de sorte que l'élément `i` du tableau contienne i^2 .
4. Affichez le tableau dans une boîte d'alerte.

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Exercice sur les boucles for</title>
    <meta charset="UTF-8"/>
    <script>

    </script>
  </head>
  <body>
    <h1>Boucles for</h1>

    <button onclick="boucle()">Cliquez-moi!</button>
    <button onclick="boucle2()">Cliquez-moi aussi!</button>
  </body>
</html>
```

UTILISATION D'UN TEST LOGIQUE

1. Appel à une fonction retournant une valeur
2. Créez un tableau nommé tab dont le premier élément est -2, le deuxième 1 et le troisième 4
3. Créez une fonction soustrait prenant un argument x et renvoyant le résultat de la soustraction de x-2 si x est positif ou nul, la chaîne de caractères "Nombre négatif!" sinon.
4. Créez une fonction affiche, appelée au clic sur le bouton, qui affiche dans des boîtes d'alerte successivement le résultat de soustrait appliqué au premier élément, puis au dernier élément du tableau (en utilisant la propriété length).

```
<html>

<head>
  <title>Utilisation d'un test logique</title>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <script type="text/javascript">
    // <!--

    // -->
  </script>
</head>
<body>

  <h1>Utilisation d'un test logique</h1>

  <p>Cliquez sur le bouton pour lancer la fonction&nbsp;;: <button onclick="">Lancer la
  fonction</button></p>

</body>
</html>
```


UTILISATION DE SWITCH ET DE L'OBJET DATE

Au click sur le bouton, lancer la fonction jourDeLaSemaine().

- Cette fonction détermine le jour de la semaine et affiche selon le cas dimanche, lundi, mardi... etc. jusqu'à samedi.

```
<html>

<head>
  <title>Utilisation de switch et de l'objet Date</title>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <script type="text/javascript">
    // <!--

    // -->
  </script>
</head>
<body>

  <h1>Utilisation de switch et de l'objet Date</h1>

  <p>Cliquez sur le bouton pour lancer la fonction&nbsp;;: <button onclick="">Lancer la
  fonction</button></p>

</body>
</html>
```

PARTIE 2 : CRÉER DES CLASSES

GÉOMÉTRIE

Notre code JavaScript va être utilisé dans un outil de dessin et on a besoin de classe pour représenter nos formes géométriques. On se concentrera sur la création de rectangle et de carré.

Le rectangle sera construit à l'aide de 2 informations (la largeur et la hauteur). Le carré lui sera construit qu'avec une seule dimension et sera l'extension du rectangle.

Voici le code qu'il vous faudra faire fonctionner :

```
const r = new Rectangle(10, 20);
console.log(r.perimeter) // 60
console.log(r.isValid) // true
const r2 = new Rectangle(-10, 20);
console.log(r2.isValid) // false
const c = new Square(10);
console.log(c.perimeter) // 40
console.log(r.isBiggerThan(c)) // true
```

BIBLIOTHÉCAIRE

Dans cet exercice on cherche à gérer une bibliothèque de livres. On commencera par créer une classe qui permettra de représenter un livre (on construira l'objet avec un titre et un nombre de page). On aura ensuite plusieurs propriétés / méthodes utiles

- page, renverra la page courante du livre (1 par défaut)
- nextPage(), permet de tourner la page et incrémentera la page courante
- close(), permet de fermer un livre (revenir à la 1ère page)

Ensuite on créera une class Library pour organiser nos livres.

- addBook(), permet de rajouter un livre à la bibliothèque
- addBooks([]), permet de rajouter plusieurs livres d'un coup (on lui passera un tableau)
- findBooksByLetter('s'), permet de lister tous les livres qui ont un titre qui commence par la lettre indiquée

Voilà le code qu'il vous faudra faire fonctionner :

```
const b = new Book('Seigneur des anneaux', 200);
console.log(b.page)
b.nextPage()
console.log(b.page)
b.close()
console.log(b.page)

const l = new Library()
l.addBook(b)
l.addBooks([
  new Book('Ready player one', 100),
  new Book('Oui-oui', 10),
  new Book('Sillage', 50),
])
console.log(l.findBooksByLetter('S'))
```