Module : JavaScript

Exercice n°7.2 - Fonctions

# Durée de l’exercice : 30minFonctions

# La prochaine étape consiste à vous entraîner à écrire votre fonction pour accomplir une tâche spécifique. Lisez la description et regardez le résultat de l'exemple pour déterminer ce qui devrait être implémenté dans la fonction. Écrivez chaque fonction dans un fichier distinct du dossier functions\_exercise et testez votre travail en exécutant le fichier.

# findArea.js

Écrivez une fonction "findArea" qui reçoit 2 paramètres (6,9) et calcule l'aire du rectangle.

# crazySum.js

Ecrivez une fonction "crazySum" qui reçoit deux entiers donnés.

Si les deux valeurs sont identiques, elle renvoie le triple de leur somme.

# average.js

// Ecrivez une fonction "average" qui accepte trois nombres comme arguments.

// La fonction doit retourner la moyenne des trois nombres.

console.log(average(3, 10, 8)) ; // 7

console.log(moyenne(10, 5, 12)) ; // 9

console.log(moyenne(6, 20, 40)) ; // 22

# half.js

// Écrivez une fonction `half` qui accepte un nombre comme argument.

// La fonction doit retourner la moitié du nombre.

console.log(half(8)) ; // 4

console.log(half(15)) ; // 7,5

console.log(half(90)) ; // 45

# is-long.js

// Écrivez une fonction `isLong` qui accepte une chaîne de caractères comme argument.

// La fonction doit retourner un booléen indiquant si la chaîne est plus longue que 5 caractères.

console.log(isLong("pie")) ; // false

console.log(isLong("kite")) ; // false

console.log(isLong("kitty")) ; // false

console.log(isLong("telescope")) ; // true

console.log(isLong("thermomètre")) ; // true

console.log(isLong("restaurant")) ; // true

# keep-it-quiet.js

// Ecrivez une fonction `keepItQuiet` qui accepte une chaîne de caractères en tant qu'argument.

// La fonction doit retourner les minuscules de la chaîne avec 3 points de suspension ajoutés à la fin.

console.log(keepItQuiet("HOORAY")) ; // 'hooray...' (en anglais)

console.log(keepItQuiet("Doggo")) ; // 'doggo...' (en anglais)

console.log(keepItQuiet("WHAT ?!?!")) ; // 'what ?!?!...' (quoi ?!!...)

# starts-with-r.js

// Ecrivez une fonction `startsWithR` qui accepte une chaîne de caractères comme argument.

// La fonction doit retourner un booléen indiquant si oui ou non la chaîne commence par 'r' ou 'R'.

console.log(startsWithR("roger that")) ; // vrai

console.log(startsWithR("Row, row, row your boat")) ; // true

console.log(startsWithR("slip")) ; // false

console.log(startsWithR("Taxicab")) ; // false

#longer.js

// Ecrivez une fonction `longer` qui accepte deux chaînes de caractères comme argument.

// La fonction doit retourner la chaîne la plus longue. Si les chaînes de caractères ont la même longueur, alors la première chaîne est retournée.

console.log(longer("drum", "piranha")) ; // 'piranha'.

console.log(longer("basket", "fork")) ; // 'basket' (panier)

console.log(longer("flanelle", "durable")) ; // "durable".

console.log(longer("disrupt", "ability")) ; // "disrupt".

console.log(longer("bird", "shoe")) ; // 'bird' (oiseau)

# ends-in-ly.js

// Écrivez une fonction `endsInLy` qui accepte une chaîne de caractères comme argument.

// La fonction renvoie un booléen indiquant si la chaîne se termine ou non par la sous-chaîne 'ly'.

console.log(endsInLy("pretty")) ; // false

console.log(endsInLy("instant")) ; // false

console.log(endsInLy("analytic")) ; // false

console.log(endsInLy("timidement")) ; // vrai

console.log(endsInLy("fly")) ; // true

console.log(endsInLy("gallantly")) ; // true

#string-size.js

// Ecrivez une fonction `stringSize` qui accepte une chaîne de caractères en tant qu’argument.

// La fonction doit retourner la chaîne 'small' si l'argument est plus court que 5 caractères, 'medium' si elle fait exactement 5 caractères, et 'large' si elle est supérieure à 5 caractères.

console.log(stringSize("cat")) ; // 'small' (petit)

console.log(stringSize("bell")) ; // 'small' (petit)

console.log(stringSize("ready")) ; // 'medium' (moyen)

console.log(stringSize("shirt")) ; // 'medium' (moyen)

console.log(stringSize("shallow")) ; // 'large' (grand)

console.log(stringSize("intelligence")) ; // 'large' (grand)

#divisible.js

// Écrivez une fonction `divisible(num1, num2)` qui accepte deux nombres comme arguments.

// La fonction doit retourner un booléen indiquant si oui ou non `num1` est divisible par `num2`.

console.log(divisible(12, 3)) ; // vrai

console.log(divisible(12, 5)) ; // false

console.log(divisible(60, 4)) ; // true

console.log(divisible(60, 11)) ; // false

console.log(divisible(21, 7)) ; // true

console.log(divisible(21, 6)) ; // false

#larger.js

// Ecrivez une fonction `larger` qui accepte deux nombres en tant qu’argument.

// La fonction doit retourner le plus grand nombre.

console.log(larger(256, 400)) ; // 400

console.log(larger(31, 4)) ; // 31

console.log(larger(-6, 7)) ; // 7

console.log(larger(11.3, 11.2)) ; // 11.3

console.log(larger(-10, -3)) ; // -3

# contains.js

// Écrivez une fonction `contains(str1, str2)` qui accepte deux chaînes de caractères comme argument.

// La fonction doit renvoyer un booléen indiquant si oui ou non `str2` est contenue dans `str1`.

// La fonction doit ignorer toute différence de capitalisation.

console.log(contains("caterpillar", "pill")) ; // true

console.log(contains("lion's share", "on")) ; // true

console.log(contains("SORRY", "or")) ; // true

console.log(contains("tangent", "gem")) ; // false

console.log(contains("clock", "ok")) ; // false

#average-of-four.js

// Écrivez une fonction `moyenneDeQuatre(num1, num2, num3, num4)` qui accepte quatre nombres comme arguments.

// La fonction doit renvoyer la moyenne des quatre nombres.

console.log(averageOfFour(10, 4, 12, 3)) ; // 7,25

console.log(averageOfFour(-20, 50, 4, 21)) ; // 13,75

console.log(averageOfFour(10, 4, 12, 3)) ; // 7.25

console.log(averageOfFour(5, 5, 3, 7)) ; // 5