

## campus.owlab 2017 || Function – param – return I

1/ Écrire une fonction **one**.

Dans le corps de one, une alert() affiche le message de votre choix.

2 / Écrire une fonction **hello**.

Dans le corps de hello, afficher la chaîne «hello» dans la console.

Ne pas utiliser **return** pour cette fonction.

Quelle est la valeur de console.log(hello()) et pourquoi ?

3 / Écrire une fonction **dummyReturn**, prenant un paramètre **p**.

**Retourner** **p**, non modifié.

Tester avec l'instruction console.log(dummyReturn())

Tester avec une affectation, var x = dummyReturn(), log(x)

4 / Écrire une fonction **foo**.

foo ne prend pas de **paramètre**.

**Retourner** la chaîne «bar» quand elle est exécutée.

Tester avec l'instruction console.log(foo())

Tester avec une affectation, var x = foo()

5 / Écrire une fonction **helloWho**, prenant un paramètre **nom**.

**Retourner** la chaîne «hello» concaténée avec **nom**.

Tester avec l'instruction console.log(helloWho())

Tester avec une affectation, var x = helloWho()

6 / Écrire une fonction **isNumber**, prenant un paramètre **n**.

**Retourner** **true** si **n** est un nombre, **false** sinon.

Tester avec l'instruction console.log(isNumber());

Tester avec une affectation, var x = isNumber()

7 / Écrire une fonction **contraire**, prenant un paramètre un booléen **b**.

Si **b** n'est pas un booléen, **retourner** une nouvelle **Erreur**.

Sinon **retourner** le **booléen** contraire de **b**.

Tester avec l'instruction console.log(contraire());

Tester avec une affectation, var x = contraire()

8 / Écrire une fonction `isEmpty`, prenant un paramètre `p`.

`isEmpty` retourne `true` si:

Si `p` est un tableau et que sa longueur est nulle.

Si `p` est une string de taille 0.

Si `p` est un objet sans aucune propriété.

Sinon `isEmpty` retourne `false`.

Tester avec l'instruction `console.log(isEmpty());`

Affecter avec NOT `isEmpty`, `var x = !isEmpty()`

9/ Créer une fonction `addCSSClass`, prenant 2 paramètres, `s` et `css`.

`s` est un sélecteur CSS (String)

`css` est une classe CSS (String)

Sélectionner le (ou les) objet(s) HTML trouvés avec `s`.

Ajouter à chaque élément la classe `css`, sans écraser les autres classes.

Retourner un tableau contenant le ou les éléments modifiés.

Tester avec l'instruction `console.log(addCSSClass());`

10/ Combiner `addCSSClass` et `isEmpty` pour déterminer si la classe CSS a bien été ajoutée à des objets HTML.

11/ Rédiger la documentation des fonctions 1 à 9 en consultant le site [usejsdoc.org](http://usejsdoc.org).  
Générer un fichier HTML avec l'utilitaire de ligne de commande fourni par jsdoc.