Module : JavaScript

Exercice n°9 - Objects

# Durée de l’exercice : 30minObjects

Alors que les tableaux utilisent des nombres pour indexer leurs valeurs, les objets utilisent des chaînes de caractères pour indexer leurs valeurs. Ces chaînes de caractères sont appelées clés.

{clé : valeur}

**Objectif de l'exercice** : Les objets sont une construction que les programmeurs utilisent pour représenter des entités et les propriétés qu'elles possèdent.

**Enoncé** : Suivez les étapes ci-dessous pour réaliser votre exercice. Prédisez d'abord le résultat puis exécutez le code dans le terminal et comparez votre prédiction.

# créer un dossier objects\_exercise dans ce dossier créer les fichiers ci-dessous, taper le code, prédisez le résultat et comparez avec le résultat imprimé après avoir exécuté le code.

# object-1.js

let movie = {

titre : "Fight Club",

year : 1999,

genre : ["drame", "thriller"],

avec : ["Brad Pitt", "Edward Norton"],

} ;

console.log(movie.year) ;

console.log(movie["title"]) ;

console.log(movie.genre) ;

console.log(movie.genre[0]) ;

console.log(movie.genre[1]) ;

console.log(movie.duration) ;

console.log(movie["starring"][1]) ;

console.log(movie.starring.length) ;

# objet-2.js

let restaurant = {

nom : "Bob's Burgers",

location : "123 Ocean Avenue",

propriétaires : ["Bob Belcher", "Linda Belcher"],

création : 2011,

menu : ["burgers", "frites", "milk-shakes"],

} ;

console.log("propriétaires" dans le restaurant) ;

console.log("employés" dans restaurant) ;

let someKey = "menu" ;

console.log(someKey dans restaurant) ;

console.log(restaurant.menu) ;

console.log(restaurant["menu"]) ;

console.log(restaurant[someKey]) ;

console.log(restaurant.someKey) ;

console.log(restaurant.menu.includes("fries")) ;

#objet-3.js

let dog = {

nom : "Manny",

âge : 5,

race : "carlin",

couleur : "fawn",

aliments préférés : ["bacon"],

} ;

console.log(chien.âge) ;

console.log(chien.race) ;

console.log(chien.favoriteFoods) ;

chien.âge++ ;

dog.breed = dog.breed.toUpperCase() ;

dog.favoriteFoods.push("saucisse") ;

console.log(chien.âge) ;

console.log(chien.race) ;

console.log(chien.favoriteFoods) ;

for (let property in dog) {

console.log(propriété + " est " + chien[propriété]) ;

}

#objet-4.js

let recipe = {

nom : "Crêpes à l'ancienne",

difficulté : "facile",

savoureux : true,

ingrédients : ["œufs", "lait", "beurre", "farine", "sucre"],

} ;

console.log(recette.nom) ;

console.log(recette["nom"]) ;

console.log(recette.ingrédients.longueur) ;

console.log(recette.calories) ;

let someVariable = "difficulty" ;

console.log(recette[someVariable]) ;

console.log(recette.quelqueVariable) ;

for (let i = 0 ; i < recipe.ingredients.length ; i++) {

console.log(recette.ingrédients[i]) ;

}

Maintenant il est temps d'écrire un code(Practice ! Practice ! Practice !)

Enregistrez les fichiers ci-dessous dans le dossier objects\_exercise et écrivez le code exécuté pour voir le résultat. Essayez de comprendre le contrôle du programme.

# max-object-value.js

// Ecrivez une fonction `maxObjectValue` qui accepte un objet comme un objet comme argument.

// Ecrivez une fonction qui renvoie un tableau contenant la paire clé-valeur de la plus grande valeur pour la plus grande valeur de l'objet. Vous pouvez supposer que toutes les valeurs sont des nombres et qu'il n'y a pas d'égalité.

console.log(maxObjectValue({ a : 5, b : 2, c : 6, d : 7, e : 4 })) ; // ['d', 7]

console.log(maxObjectValue({ litchi : 11, ramboutan : 13, papaye : 9 })) ; // ['ramboutan', 13]

# key-pair.js

// Écrivez une fonction `keyPair(obj1, obj2, key)` qui accepte deux objets et une chaîne de caractères comme arguments.

// La fonction doit retourner un tableau contenant les valeurs de la clé donnée dans `obj1` et `obj2`.

// Voir les exemples.

let cat1 = { nom : "jinkee", race : "calico" } ;

let cat2 = { name : "garfield", breed : "red tabby" } ;

console.log(keyPair(cat1, cat2, "breed")) ; // [ 'calico', 'redtabby' ]

console.log(keyPair(cat1, cat2, "name")) ; // [ 'jinkee', 'garfield' ]

let sport1 = { nom : "volleyball", équipe : true } ;

let sport2 = { nom : "golf", équipe : false } ;

console.log(keyPair(sport1, sport2, "team")) ; // [ true, false ]

# element-quantities.js

// Créez une fonction `elementQuantities` qui accepte un objet comme argument. L'objet contient des éléments comme clés et des quantités comme valeurs.

// La fonction doit retourner un tableau contenant les éléments qui apparaissant avec les quantités correctes.

// Voir les exemples.

let quantities1 = { cat : 3, oiseau : 1, chien : 2 } ;

console.log(elementQuantities(quantities1)) ; // ['cat', 'cat', 'chat', 'oiseau', 'chien', 'chien']

let quantities2 = { blue : 3, marron : 1 } ;

console.log(elementQuantities(quantities2)) ; // ['blue', 'blue', 'blue', 'brown']

# object-add.js

// Écrivez une fonction `objectAdd` qui accepte deux objets comme arguments.

// La fonction doit retourner un nouvel objet contenant les mêmes clés que les objets arguments.

// Si une clé est trouvée dans les deux objets, sa valeur correspondante doit être la somme des valeurs des arguments de l'objet.

// Si une clé n'est trouvée que dans l'un des objets, la valeur correspondante doit être la même que celle de ce seul objet.

let obj1 = { x : 3, y : 10} ;

let obj2 = { y : 2, x : 1} ;

console.log(objectAdd(obj1, obj2)) ; // { x : 4, y : 12 }

let obj3 = { a : 3, b : 2, c : -1} ;

let obj4 = { b : 5, c : 1, e : 4} ;

console.log(objectAdd(obj3, obj4)) ; // { a : 3, b : 7, c : 0, e : 4

}

# greatest-population.js

// Ecrivez une fonction `greatestPopulation` qui accepte en argument un tableau d'objets pays comme argument.

// La fonction doit retourner le nom du pays qui a la plus grande population.

// Vous pouvez supposer que le tableau contient au moins un pays.

let countries1 = [

{ nom : "Cameroun", population : 27744989, PIB : 38.68 },

{ nom : "Belarus", population : 9477918, pib : 59.66 },

{nom : "Indonésie", population : 267026366, PIB : 1042 },

{nom : "Guyane", population : 750204 750204, PIB : 3.88 },

] ;

console.log(greatestPopulation(countries1)) ; // 'Indonesia' (Indonésie)

let countries1 = [

{ nom : "Nouvelle-Zélande", population : 4925477, pib : 204,9 },

{ nom : "Mozambique", population : 30098197, pib : 14.72 },

{nom : "Groenland", population : 57616, PIB : 2,71 },

{nom : "Kazakhstan", population : 19091949, PIB : 179.3 },

{nom : "Birmanie", population : 56590071, PIB : 71.21 },

] ;

console.log(greatestPopulation(countries2)) ; // 'Birmanie'.

*Ci-dessous la progression du travail ---------------*