

Testez vos compétences : Fonctions

L'objectif de ce test de compétences est d'évaluer si vous avez compris nos [articles Fonctions — blocs de code réutilisables](#) , [Créez votre propre fonction](#) et [Valeurs de retour des fonctions](#) .

Remarque : Vous pouvez essayer des solutions en téléchargeant le code et en le plaçant dans un éditeur en ligne tel que [CodePen](#) , [JSFiddle](#) ou [Glitch](#) . S'il y a une erreur, elle sera enregistrée dans le panneau de résultats de la page ou dans la console JavaScript du navigateur pour vous aider.

Si vous êtes bloqué, vous pouvez nous contacter via l'un de nos [canaux de communication](#) .

Manipulation du DOM : considérée comme utile

Certaines des questions ci-dessous nécessitent que vous écriviez du code de manipulation [DOM](#) pour les compléter, par exemple en créant de nouveaux éléments HTML, en définissant leur contenu textuel sur des valeurs de chaîne spécifiques et en les imbriquant dans des éléments existants de la page, le tout via JavaScript.

Nous ne l'avons pas encore enseigné explicitement dans le cours. mais vous aurez vu



point de départ est notre didacticiel [sur la manipulation de documents](#) .

Fonctions 1

Pour la première tâche, vous devez créer une fonction simple — `chooseName()` — qui imprime un nom aléatoire du tableau fourni (`names`) dans le paragraphe fourni (`para`), puis l'exécuter une fois.

Essayez de mettre à jour le code live ci-dessous pour recréer l'exemple terminé :

```
'Bert', 'Jada']
const para = document.createElement('p');

// Add your code here

function chooseName(){
  let name = Math.random(names);
}

para.textContent = chooseName();

// Don't edit the code below here!

section.innerHTML = ' ';
```

Reset

[Téléchargez le point de départ de cette tâche](#) pour travailler dans votre propre éditeur ou dans un éditeur en ligne.

Fonctions 2

Pour notre deuxième tâche liée aux fonctions, vous devez créer une fonction qui dessine un rectangle sur la variable de référence fournie `<canvas>` (variable de référence `canvas`, contexte disponible dans `ctx`), en fonction des cinq variables d'entrée fournies :

- `x` — la coordonnée x du rectangle.
- `y` — la coordonnée y du rectangle.
- `width` — la largeur du rectangle.
- `height` — la hauteur du rectangle.
- `color` — la couleur du rectangle.

Vous souhaitez effacer le canevas avant de dessiner, afin que lorsque le code est mis à jour dans le cas de la version live, vous n'obteniez pas beaucoup de rectangles dessinés les uns sur les autres.

Essayez de mettre à jour le code live ci-dessous pour recréer l'exemple terminé :

```
const canvas = document.querySelector('canvas');
const ctx = canvas.getContext('2d');

const x = 50;
const y = 60;
const width = 100;
const height = 75;
const color = 'blue';

// Add your code here
```

Reset

[Téléchargez le point de départ de cette tâche](#) pour travailler dans votre propre éditeur ou dans un éditeur en ligne.

Fonctions 3

Dans cette tâche, vous revenez au problème posé dans la tâche 1, dans le but de l'améliorer. Les trois améliorations que nous souhaitons que vous apportiez sont :

1. Refactorisez le code qui génère le nombre aléatoire dans une fonction distincte appelée `random()`, qui prend comme paramètres deux limites génériques entre lesquelles le nombre aléatoire doit se trouver et renvoie le résultat.
2. Mettez à jour la `chooseName()` fonction afin qu'elle utilise la fonction de nombre aléatoire, prenne le tableau parmi lequel choisir comme paramètre (le rendant plus flexible) et renvoie le résultat.
3. Imprimez le résultat renvoyé dans le paragraphe (`para`) `textContent`.

Essayez de mettre à jour le code live ci-dessous pour recréer l'exemple terminé :

```
const names = ['Chris', 'Li Kang', 'Anne', 'Francesca', 'Mustafa', 'Tina',  
  'Bert', 'Jada']  
const para = document.createElement('p');  
  
// Add your code here  
  
// Don't edit the code below here!  
  
section.innerHTML = ' ';  
section.appendChild(para);
```

Reset

[Téléchargez le point de départ de cette tâche](#) pour travailler dans votre propre éditeur ou dans un éditeur en ligne.

Fonctions 4

Dans cette tâche, nous avons un tableau de noms et nous utilisons `Array.filter()` pour obtenir un tableau contenant uniquement des noms de moins de 5 caractères. Le filtre reçoit actuellement une fonction nommée `isShort()` qui vérifie la longueur du nom, retournant `true` si le nom contient moins de 5 caractères, et `false` sinon.

Nous aimerions que vous transformiez cela en fonction de flèche. Voyez à quel point vous pouvez le rendre compact.

Anne,Tina,Bert,Jada

```
const names = ['Chris', 'Li Kang', 'Anne', 'Francesca', 'Mustafa', 'Tina',  
  'Bert', 'Jada'];  
const para = document.createElement('p');  
  
function isShort(name) {  
  return name.length < 5;  
}  
  
const shortNames = names.filter(isShort);  
para.textContent = shortNames;  
  
// Don't edit the code below here!  
  
section.innerHTML = ' ';
```

Reset

[Téléchargez le point de départ de cette tâche](#) pour travailler dans votre propre éditeur ou dans un éditeur en ligne.

Cette page a été modifiée pour la dernière fois le 21 novembre 2023 par [les contributeurs de MDN](#).