

Cette page a été traduite à partir de l'anglais par la communauté. Vous pouvez également contribuer en rejoignant la communauté francophone sur MDN Web Docs.

Galerie d'images

Maintenant que nous avons examiné les blocs fondamentaux de construction de JavaScript, nous allons tester vos connaissances sur les boucles, les fonctions, les conditions et les événements en vous aidant à créer un élément assez commun que vous verrez sur de nombreux sites web. Une galerie JavaScript animée.

Conditions préalables:	Avant de tenter cette évaluation, vous devriez avoir parcouru tous les articles de ce module.
Objectif:	Tester la compréhension des boucles, des fonctions, des conditions et des événements JavaScript.

Point de départ

Pour réaliser cette évaluation, vous devez récupérer le fichier [ZIP](#) et le décompresser quelque par sur votre ordinateur.

Vous pouvez également utiliser un site comme [JSBin](#) ou [Thimble](#) pour effectuer votre évaluation. Vous pouvez copier le code HTML,CSS et JavaScript dans l'un de ces éditeurs en ligne. Si l'éditeur en ligne que vous utilisez ne dispose pas de panneaux JavaScript/CSS séparés, n'hésitez pas à utiliser les éléments `<script>` / `<style>` dans la page HTML.

Note : Si vous êtes bloqué, demandez-nous de l'aide — voir la section [Évaluation ou aide supplémentaire](#) au bas de cette page.

Résumé du projet

Vous avez reçu des fichiers HTML, CSS, des images et quelques lignes de code JavaScript; vous devez écrire le code JavaScript nécessaire pour en faire un programme fonctionnel. Le corps HTML ressemble à ceci:

HTML

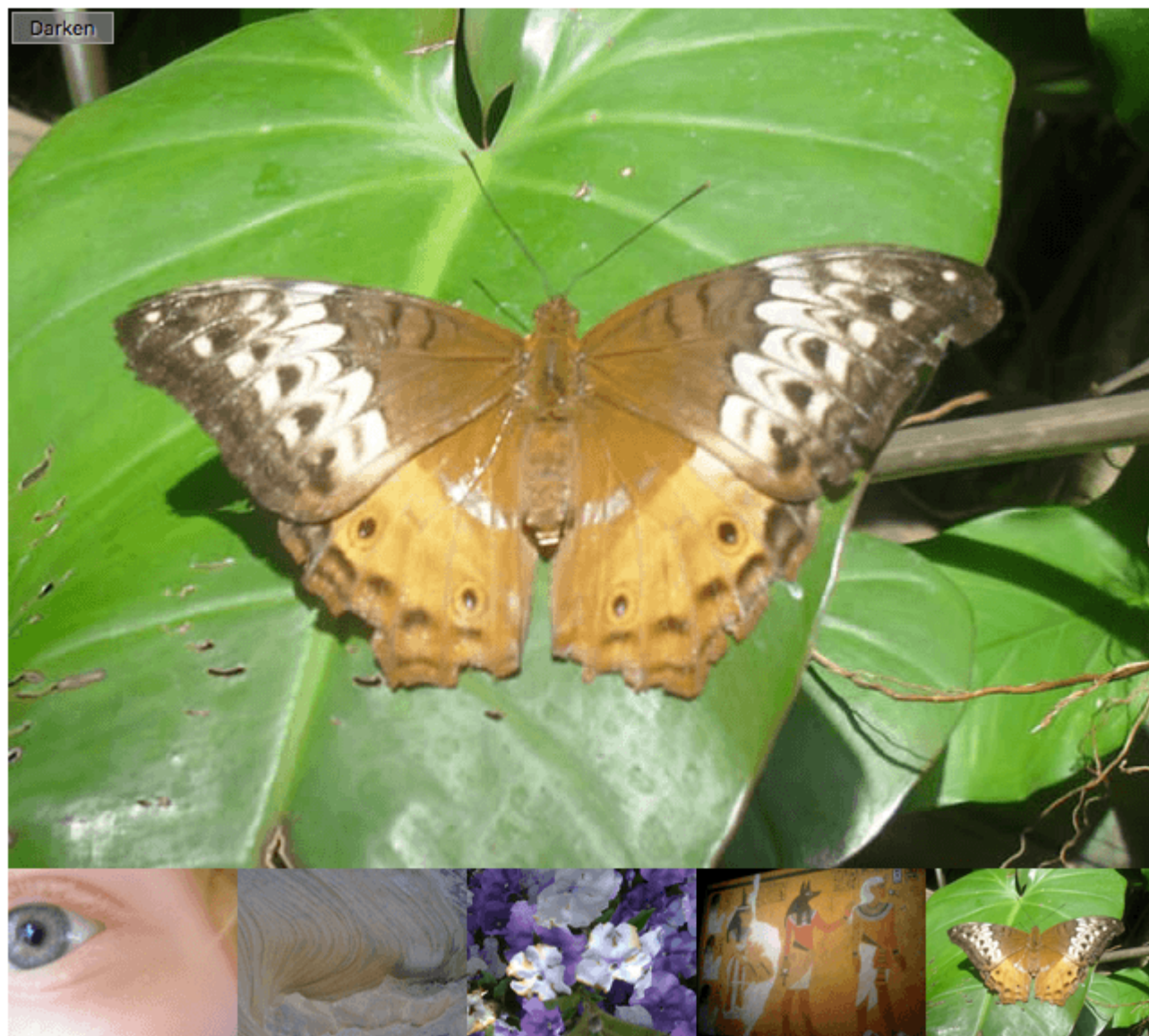
```
<h1>Image gallery example</h1>

<div class="full-img">
  
  <div class="overlay"></div>
  <button class="dark">Darken</button>
</div>

<div class="thumb-bar"></div>
```

L'exemple ressemble à ceci:

Image gallery example



Les parties les plus intéressantes du fichier CSS de l'exemple:

- Positionnez les trois éléments en absolu à l'intérieur du `full-img <div>`: le `` dans lequel l'image est affichée grandeur nature, un `<div>` vide dimensionné à la même taille que le `` et placé juste au-dessus (ceci est utilisé pour appliquer un effet assombrissant à l'image via une couleur d'arrière-plan semi-transparente), et un bouton `<button>` qui est utilisé pour contrôler l'effet d'assombrissement.
- Réglez la largeur des images à l'intérieur de `thumb-bar <div>` (appelées images miniatures) à 20% et faites un float à gauche pour qu'elles soient côte-à-côte sur une ligne.

Votre JavaScript doit:

- Itérer toutes les images dans une boucle, et pour chacune d'elle, insérer un élément `` à l'intérieur de `thumb-bar <div>` qui va incorporer cette image dans la page.
- Associer un gestionnaire d'événement `onclick` à chaque `` à l'intérieur de `thumb-bar <div>` pour que, lorsqu'elles sont cliquées, l'image correspondante soit affichée dans l'élément `displayed-img `.
- Associer un gestionnaire d'événement `onclick` au `<button>` pour que, lorsqu'il est cliqué, un effet assombrissant soit appliqué à l'image grandeur nature. Lorsqu'il est cliqué à nouveau, l'effet doit disparaître.

Pour vous donner une idée, regardez l'[exemple](#) (Ne regardez pas le code source!).

Les différentes étapes

Les sections suivantes décrivent ce que vous avez à faire.

Itération sur les images

Nous vous avons fourni des lignes qui stockent une référence à `thumb-bar <div>` dans une variable nommée `thumbBar`, créent un nouvel élément ``, définissent son attribut `src` à un emplacement de valeur `xxx`, et ajoutent ce nouvel élément `` dans `thumbBar`.

Vous avez besoin de:

1. Ajouter votre code en-dessous du commentaire *Looping through images* à l'intérieur d'une boucle qui itère sur les 5 images — vous n'avez besoin que de 5 itérations, chacune représentant une image.
2. Remplacez, pour chaque itération, la valeur `xxx` de l'emplacement par une chaîne de caractères qui correspond au chemin de l'image considérée. Il faut définir la valeur de l'attribut `src` dans chaque cas. Gardez à l'esprit que, à chaque fois, l'image est dans le répertoire des images et que son nom est `pic1.jpg`, `pic2.jpg`, etc.

Ajout d'un gestionnaire onclick à chaque miniature

À chaque itération, vous devez ajouter un gestionnaire `onclick` au `newImage` courant. Il doit:

1. Trouver la valeur de l'attribut `src` de l'image courante. Cela peut être fait avec la fonction [getAttribute\(\)](#) sur ``, en lui passant le paramètre `"src"` à chaque fois. Mais comment avoir l'image? Utiliser `newImage` ne marche pas du fait que la boucle est finie avant que le gestionnaire d'événement ne soit appelé; de cette manière, la valeur de `src` sera toujours celle du dernier ``. Pour résoudre cela, gardez à l'esprit que, pour chaque gestionnaire d'événement, c'est `` qui en est la cible. Pourquoi ne pas récupérer l'information de l'objet événement?
2. Exécuter une fonction, en lui passant en paramètre la fameuse valeur de `src`. Vous pouvez nommer la fonction à votre guise.
3. Cette fonction du gestionnaire d'événement doit définir la valeur de l'attribut `src` de `displayed-img ` à la valeur du `src` passé en paramètre. Nous vous avons fourni une ligne qui stocke une référence de l'`` concerné dans une variable nommée `displayedImg`. Notez que nous voulons une fonction nommée.

Écrire le gestionnaire du bouton d'assombrissement

Il ne reste que notre `<button>` d'assombrissement — nous vous avons fourni une ligne qui stocke une référence au `<button>` dans une variable appelée `btn`. Vous devez ajouter un gestionnaire `onclick` qui:

1. Vérifie la classe appliquée à `<button>` — à nouveau, vous pouvez utiliser `getAttribute()`.
2. Si le nom de classe est `"dark"`, changer la classe du `<button>` pour `"light"` (avec [setAttribute\(\)](#)), son contenu textuel par `"Lighten"`, et le [background-color](#) du voile d'assombrissement `<div>` par `"rgba(0,0,0,0.5)"`.
3. Si le nom de classe n'est pas `"dark"`, changer la classe du `<button>` pour `"dark"`, son contenu textuel par `"Darken"`, et le [background-color](#) du voile d'assombrissement `<div>` par `"rgba(0,0,0,0)"`.

Les lignes suivantes fournissent une base pour réaliser les changements décrits aux points 2 et 3.

JS

```
btn.setAttribute("class", xxx);  
btn.textContent = xxx;  
overlay.style.backgroundColor = xxx;
```



Évaluation ou aide supplémentaire

Si vous souhaitez que votre travail soit évalué, ou si vous êtes bloqué et que vous voulez demander de l'aide :

1. Mettez votre travail dans un éditeur partageable en ligne tel que [CodePen](#) , [jsFiddle](#) , ou [Glitch](#) .
2. Rédiger un sujet pour demander une évaluation et/ou une aide à le [forum Discourse du MDN](#) . Ajoutez la balise "learning" à votre message pour que nous puissions le trouver plus facilement. Votre message doit inclure :
 - Un titre descriptif tel que "Évaluation demandée pour la galerie d'images".
 - Des détails sur ce que vous souhaitez que nous fassions — par exemple ce que vous avez déjà essayé, si vous êtes bloqué et avez besoin d'aide.
 - Un lien vers l'exemple que vous souhaitez faire évaluer ou pour lequel vous avez besoin d'aide, dans un éditeur en ligne. C'est une bonne pratique à adopter — il est très difficile d'aider une personne ayant un problème de codage si on ne peut pas voir son code.
 - Un lien vers la page de la tâche ou de l'évaluation proprement dite, afin que nous puissions trouver la question pour laquelle vous souhaitez de l'aide.

Si vous suivez cette évaluation dans le cadre d'un cours organisé, vous devriez pouvoir donner votre travail à votre professeur ou mentor pour la notation. Si vous apprenez en autodidacte, vous pouvez obtenir le guide de notation simplement en le demandant [sur le fil de discussion de cet exercice](#) , ou sur [#mdn](#) du canal IRC de [Mozilla IRC](#) . Faites d'abord l'exercice, vous ne gagnerez rien à tricher!

This page was last modified on 3 août 2023 by [MDN contributors](#).