

Exécutez du Javascript dans une page HTML

HTML, JS

Langage Compétences : 2

Lorsqu'il faut exécuter du code sur une page HTML, on pense immédiatement à JavaScript. Mais quelles sont les méthodes pour le faire correctement. Voici un résumé des différentes manières d'y parvenir.

Publié lundi 4 octobre 2021, 09h04
Modifié lundi 26 août 2024, 10h04



By Olivier Paudex

Introduction

Exécuter du Javascript sur une page web peut se faire de différentes manières. Encore faut-il encore comprendre les avantages et inconvénients de chacune d'entre elles. Voici quelques astuces pour exécuter du code Javascript de différentes manières.

Les fichiers de base

Cette publication se base sur deux fichiers :

- **index.html**, qui contient le code HTML de la page à afficher

- **script.js**, qui contient le code Javascript qui va s'exécuter sur les objets de la page HTML

Pour que les exemples fonctionnent, les deux fichiers doivent être créés dans le même dossier.

Le fichier Javascript

Commençons par créer un fichier **"script.js"**, qui va servir de base aux exemples.

Ci-dessous, un exemple de code qui va afficher dans la console du navigateur l'élément DOM dont l'ID est **"info".** Il va ensuite le modifier, puis afficher son nouveau contenu.

```
const myElement = document.getElementById('info');
console.log (myElement.innerHTML);
myElement.innerHTML = "Ceci est le texte modifié";console.log (myElement.innerHTML);
```

Le fichier HTML

La deuxième étape consiste à créer le fichier **"index.html"** que l'on va modifier au fur et à mesure des exemples.

Ci-dessous, un exemple de code qui va afficher un titre, formaté en **"H2"**, puis un paragraphe, dont l'ID est **"info"**.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>My Web page</title>
  </head>
  <body>
    <div>
      <h2>Titre</h2>
      <p id="info">Ceci est un texte</p>
    </div>
  </body></html>
```

La méthode embarquée ou classique

La méthode la plus simple consiste à exécuter le code Javascript directement depuis le contenu HTML, en y ajoutant les balises **"<script>"** et **"</script>".**

Il est aussi possible d'appeler un fichier externe de cette manière.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>My Web page</title>
  </head>
  <body>
    <div>
      <h2>Titre</h2>
      <p id="info">Ceci est un texte</p>
    </div>
  </body>
  <script src="./script.js" type="text/javascript"></script></html>
```

Cependant, la position de la balise **"<script>"** est très importante. Pour exécuter les modifications sur l'objet DOM **"info"**, il faut obligatoirement écrire cette ligne après le **"body"**.

La méthode "defer"

Si l'on désire exécuter le code Javascript après que la page soit entièrement chargée, la manière la plus simple est encore d'utiliser le mot clé **"defer"**. Pour ceci, il faut ajouter la référence au fichier Javascript dans la section **"head"** du fichier HTML. Contrairement à la méthode classique, cette ligne est écrite dans l'entête de la page HTML, ce qui rend le code tout de suite beaucoup plus clair.

```
<script src="./script.js" type="text/javascript" defer></script>
```

La méthode “DOMContentLoaded”

Pour appliquer cette méthode, il faut ajouter la même référence que ci-dessus au code HTML, mais sans le **“defer”**.

```
<script src="./script.js" type="text/javascript"></script>
```

Puis, dans un deuxième temps, il faut modifier le code Javascript. Cette méthode va créer un événement qui se déclenche après le chargement de tous les objets DOM de la page web. En d’autres mots, il faut utiliser cette méthode seulement si l’on ne désire pas modifier les images et le code CSS.

```
document.addEventListener("DOMContentLoaded", function(){
  const myElement = document.getElementById('info');
  console.log (myElement.innerHTML);
  myElement.innerHTML = "Ceci est le texte modifié";
  console.log (myElement.innerHTML);});
```

La méthode “load”

A l’instar de la méthode DOM, cette méthode va créer un événement qui se déclenche après le chargement complet de la page web, images et feuille de styles y compris. La page étant entièrement chargée, on peut donc utiliser cette méthode en toute sécurité.

```
window.addEventListener("load", function(){
  const myElement = document.getElementById('info');
  console.log (myElement.innerHTML);
  myElement.innerHTML = "Ceci est le texte modifié";
  console.log (myElement.innerHTML);});
```

La méthode “onload”

La méthode **“onload”** est à appliquer au code HTML, sur le DOM **“body”**, par exemple. Il permet de charger une fonction bien définie.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Ma page web</title>
    <script src="./script.js" type="text/javascript"></script>
  </head>
  <body onload="info()">
    <div>
      <h2>Titre</h2>
      <p id="info">Ceci est un texte</p>
    </div>
  </body></html>
```

Le code Javascript devient alors une simple fonction dont le nom est **“info”**.

```
 function info(){
  const myElement = document.getElementById('info');
  console.log (myElement.innerHTML);
  myElement.innerHTML = "Ceci est le texte modifié";
  console.log (myElement.innerHTML);};
```

La librairie jQuery

La librairie jQuery facilite l'écriture du code. Si l'on utilise un CMS comme **"Wordpress"**, celle-ci est déjà incluse et prête à l'emploi. Mais si l'on utilise aucun CMS, comme dans les exemples de cette publication, alors, il faut télécharger la bibliothèque de fonctions et la référencer au fichier HTML.

- **Téléchargez la dernière version de jQuery.**
- Copiez le fichier dans le même dossier que le fichier HTML et Javascript.
- Référencez la librairie comme ci-dessous **(pour la version compressée et minifiée 3.6.0)**

```
<script src="jquery-3.6.0.min.js"></script>
<script src="./script.js" type="text/javascript"></script>
```

Une fois la référence ajoutée au code HTML, on peut commencer à écrire du code jQuery.

> **Attention à la syntaxe** : Ne pas oublier de fermer la parenthèse à la dernière ligne.

```
$(document).ready(function(){
  const myElement = document.getElementById('info');
  console.log (myElement.innerHTML);
  myElement.innerHTML = "Ceci est le texte modifié";
  console.log (myElement.innerHTML);});
```

Le mot de la fin

Voilà quelques unes des manières d'exécuter du code Javascript sur une page web,avec ou sans un CMS. La difficulté du choix survient au moment de penser à la performance du site. Choisir de charger toute la page avant d'exécuter le code Javascript est souvent la meilleure solution, mais aussi la moins performante. Quant à jQuery, ses méthodes sont devenus tellement populaire que la librairie est un standard appliqué par beaucoup de développeurs.