

La librairie Prism

CSS, JS, Wordpress

Web Compétences : 2

La rédaction de contenus informatique comprenant de nombreux exemples de code demande, pour être clair, une mise en évidence. Wordpress propose nativement une balise HTML qui permet de réaliser ceci de différentes manières. Voici une méthode, utilisée sur ce site, qui utilise une librairie et un plugin.

Publié mardi 15 septembre 2020, 19h14
Modifié lundi 26 août 2024, 10h05



By Olivier Paudex

Introduction

Avec le passage à l'éditeur **Gutenberg**, WordPress a créé un malaise dans la communauté des utilisateurs. Certains voyaient en Gutenberg, un nouvel outil, compliqué à utiliser, à paramétrer, et de ce fait, pas vraiment utile. De l'autre côté de la barrière, on lisait les commentaires approbateurs de ceux qui avaient totalement adopté le nouvel éditeur.

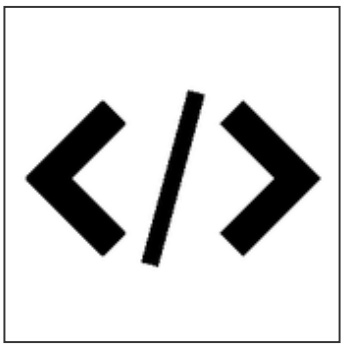
Avec Elementor comme constructeur de pages, certains se sont même poser la question si celui-ci avait encore une raison d'exister. La réponse ne tarda pas d'arriver et selon les termes d'Elementor : Il ne faut pas confondre **"constructeur de pages"** et **"éditeur de contenu"**. Même si certains diront qu'il est tout à fait envisageable de créer une page avec Gutenberg comme seul outil.

Le contenu de Fuyens.ch est entièrement créé avec Gutenberg, et toutes pages sont construite avec Elementor.

Bloc syntaxique de code

Ce site, comme beaucoup d'autres qui publient des sujets sur l'informatique, rehausse les explications par des exemples de code. Dans l'éditeur Gutenberg, il existe une balise HTML de type**"<code></code>"**. Même si cette balise permet de copier-coller un exemple de code, celui-ci reste écrit en noir sur fond blanc, sans surligneur pour la syntaxe. Ce qui ne donne pas vraiment l'effet attendu.

Il est possible d'améliorer ceci, en utilisant un plugin de mise en évidence de la syntaxe. Ce site utilise le plugin **"Code Syntax Block"**.



Le plugin Code Syntax Block

Le plugin s'intègre parfaitement à Gutenberg et offre la possibilité de sélectionner le langage de programmation, de numéroter les lignes de code, ainsi que de choisir un thème de couleurs.

La librairie Prism

Le plugin ci-dessus est construit sur la base de la librairie Prism. **Prism est un surligneur syntaxique léger et extensible**. Il est conçu en deux parties, la feuille de style CSS et les actions écrit en Javascript.

Il est possible de créer sa propre librairie en se rendant sur le site de Prism sous **https://prismjs.com/**. Prism propose aujourd'hui :

- 8 thèmes.
- Plus de 200 langages.
- Une trentaine de plugins.

Pour créer un thème, il suffit de sélectionner les options et de récupérer les deux fichiers**"css"** et **"js"**.

Vous pouvez aussi récupérer celui de ce site, en cliquant sur le lien**"pièce jointe"**, en haut de cette publication.



Les deux fichiers résultant de la personnalisation de la librairie Prism

Intégration avec le plugin Code Syntax Block

Il est possible d'utiliser la librairie Prism avec le plugin Code Syntax Block. Il suffit de copier le fichier**"css"** avec les autres, dans le dossier du même nom, à la racine du thème enfant, puis d'ajouter ces quelques lignes de code :

```
// Use local prism theme
function Prism_CSS_Path() {
  return '/css/prism.css';
}add_filter('mkaz_prism_css_path');
```

Ce filtre a été écrit par l'auteur du plugin, **Marcus Kazmierczak**. Toute la documentation se trouve sur son **Github**.

> La liaison de la feuille de style avec le plugin fixe le thème choisi pour toutes les fenêtres de code. La sélection du thème depuis l'éditeur de Gutenberg devient caduque.

Intégration des plugins de Prism

La librairie Prism se compose également d'un fichier **"js"** qui permet d'ajouter des fonctions, comme le fameux **"Copy"**, utilisé par ce site, et l'affichage du langage.

Comme l'auteur ne parle pas de l'intégration du fichier **"js"** dans sa documentation, il est toujours possible d'utiliser la méthode classique de mise en queue de code Javascript, pour intégrer ce fichier. Il suffit d'ajouter ces quelques lignes de code dans un fichier **"php"** et de copier le fichier **"prism.js"** dans un dossier **"js"** à la racine du thème enfant.

```
// Prism Highlighter library
if (!function_exists('PrismJS')) {
  function PrismJS() {
    // Only on single pages
    if (!is_admin() && (is_single())) {
      $jquery = array ('jquery');
      $version = '1.0';
      $in_footer = true;
      // Enqueue script
      wp_enqueue_script('PrismJS', get_stylesheet_directory_uri() . '/js/prism.js', $jquery, $
    }
  }
}add_action('wp_enqueue_scripts', 'PrismJS');
```

Il suffit de passer la souris sur la zone de code pour voir les nouveaux boutons apparaître. La fonction **"Copy"** est bien pratique.



Les plugins de la librairie Prism