# Utiliser le composeur de pages Elementor

En cours d'actualisation...

En attendant,vous pouvez consulter les anciennes versions des tutoriels

## Elementor, c'est quoi ?

### Un composeur de page "clé en main"

Le site de l'ORB a été conçu avec WordPress, et son plugin Elementor qui embarque son propre éditeur de contenus. Cet éditeur Elementor vient en complément de l'éditeur par défaut de Wordpress, et il apporte une interface et des widgets permettant aux utilisateurs de créer des rendus plus esthétiques, plus rapidement, et avec un temps de formation moindre (pas de code).

Elementor "code pour vous" un ensemble de widgets que vous pourrez importer dans vos pages, et que vous pourrez paramétrer dans une assez grande mesure. Pour les utilisateurs plus avancés, il permet d'ajouter des règles css et de personnaliser encore davantage les widgets mis à disposition.

### Éditeur WordPress ou composeur Elementor ?

**Elementor ne prend en charge que la gestion et l'organisation des contenus directement visibles par les visiteurs du site**. Il peut remplacer l'éditeur par défaut de Wordpress (Gutenberg) pour le contenu des pages en lui-même.

**Les "métadonnées" des publications** (auteurs, mots-clés, images mises en avant, référencement, dates de publications, catégories, permaliens etc) **restent gérées par WordPress de manière classique**. De même, les fonctionnalités permettant de publier, programmer, dépublier un article, gérer les médias ou créer des utilisateurs par exemple, restent totalement gérés par l'interface classique de WordPress (/wp-admin).

Pour faire simple :

| Tâche | Éditeur |
|---|---|
| Métadonnées des posts (dates de publications, référencement, classification, auteurs, images en avant etc) | WordPress |
| Contenu visible par l'utilisateur (titres, images, textes etc.) | Elementor recommandé (rendus plus esthétiques et simples à mettre en forme), Wordpress à défaut |

## Fonctionnement général

## Fonctionnement de base d'une page web

Une page web est essentiellement conçue avec 2 langages complémentaires, ayant chacun une fonction précise. D'autres langages peuvent venir "en option", nous ne nous y attarderons pas :

| Langage | Fonction |
|---|---|
| HTML | Gère le contenu de la page et sa structure (squelette) : il définit comment sont imbriqués les contenus entre eux au sein de "divisions", et définit le contenu et sa nature (Par exemple, <h1>Ceci mon titre</h1> indique que la page contient un titre principal (balise h1), et que le contenu du titre est "Ceci mon titre" |
| CSS | Gère le style des éléments de la page. C'est ce langage qui va définir la police à utiliser, la taille, la couleur, les marges autour de l'objet, la disposition de cet élément par rapport aux autres, les bordures, les effets d'ombre, les superpositions, les images d'arrière-plan etc. |
| Autres (javascript, ajax...) | Peuvent venir en complément pour gérer des éléments dynamiques (carroussels, filtrages de contenus etc) |

L'objectif d'Elementor est de ne pas se soucier de ça. Mais l'information est utile - sans apprendre ces langages - car elle est structurante dans le fonctionnement d'Elementor.

## Structurer sa page : des "conteneurs" et des "widgets"

### Les conteneurs

Les "conteneurs" correspondent aux "divisions" du code HTML : ce sont eux qui vont servir de squelette à la page en définissant l'imbrication des contenus les uns par rapport aux autres.

Il faut les voir comme des poupées russes ou des "cases ikea" qu'on agence comme on le souhaite. On peut imbriquer autant de conteneurs qu'on le souhaite, pour gérer au mieux l'affichage de la page et aboutir à un rendu organisé.

Plus la page est structurée correctement, mieux elle sera traduite sur toutes les tailles d'écrans (les navigateurs utilisent les conteneurs pour recouper les contenus, mettre côte-à-côte ou les uns sous les autres les différents contenus en fonction de la taille d'écran). Par exemple :

### Les widgets

## Interface du composeur

### Interface générale et banque de widgets

### Onglet "Contenu"

**Onglet "Style"**

**Onglet "Avancé"**

**Brouillons, modèles et publication des pages**

# Bonnes pratiques pour la construction des contenus : une rigueur obligatoire !

### Référencement et handicap : une même approche

Les outils de référencement et d'accessibilité ont un point commun important pour nous. Techniquement, les deux reposent sur une machine qui analyse le code source de la page (et pas ce qui apparait à l'écran), pour interpréter et restituer des informations clés de la page.

Cela présente un avantage pour nous : en optimisant notre manière de construire les pages, Elementor optimisera le code source, et nos bonnes pratiques amélioreront de concert le référencement du contenu et son accessibilité aux publics invalides.

### Renseigner correctement ses médias

Lorsque vous chargez un média, il est nécessaire de **renseigner ses métadonnées : titre et texte alternatif sont obligatoires**. Ces deux champs sont restitués dans le code source pour aider les machines à comprendre ce que représente l'image affichée. Si ces champs ne sont pas renseignés, les machines savent qu'il y a une image mais n'ont aucune notion de ce qu'elle représente.

C'est également un bon moyen d'augmenter les mots-clés qui pourront être trouvées au sein de la page, sans forcément apparaitre textuellement.

### Utiliser le bon widget

Dans Elementor (et dans les langages qui codent les sites internet en général), il est possible d'obtenir le même rendu visuel par plusieurs moyens. Par exemple, visuellement, on peut créer du texte qui ressemble à un titre (taille de police, couleur, centré etc.). Mais le rendu visuel ne suffit pas, un code source pertinent est indispensable pour que la page soit correctement chargée, interprétée pour le référencement, et adaptée par les outils d'accessibilité. Or un titre et un texte ne sont pas codés et interprétés de la même manière : pour les machines, un gros texte violet n'est qu'un gros texte violet, pas un titre !

Même si Elementor nous permet de ne pas coder (il le fait à notre place), l'outil a besoin de disposer des bonnes informations pour coder le plus correctement possible. Ainsi, le widget "Titre" insèrera une balise "titre" au bon endroit et avec différents niveaux de titres possibles (voir ci-dessous), tandis qu'un widget texte créera un simple paragraphe. De la même manière, un widget "éditeur de texte" nous permet d'inclure un média dans le texte, mais le code sera différent si vous procédez de

manière optimale,en utilisant un widget texte, un widget image, puis un widget texte (dans ce cas, l'image ne sera pas intégrée d'abord comme un paragraphe, et vous aurez tous les outils nécessaires pour gérer la taille du média, sa position, etc.).

### Les balises de titres

Pour tous sites internets, les titres sont classés par niveau grâce à des balises. Le titre principal de la page aura ainsi une balise h1 (pour header 1), les sous-titres une balise h2 etc.

- h1 : titre de la page
    - h2 : sous-titre
        - h3 : sous-sous-titre
            - h4 : sous-sous-sous-titre etc.

Respecter ces balises permet notamment :

- de faire apparaitre la structure de la page aux outils lisant le code source (référencement, lecteurs d'écran pour l'accessibilité)
- de hiérarchiser l'information dans le code source (sujets principaux, sous-thèmes etc) pour le référencement
- généralement d'avoir un style cohérent (tailles de police, couleur, gras) appliqués par défaut

## Documentation

La documentation officielle d'Elementor est disponible ici :
https://elementor.com/academy/guides-and-tutorials/

Tutoriels vidéos : https://www.youtube.com/c/elementor

D'une manière générale, de nombreuses documentations, tutoriels et vidéos se trouvent en ligne. Chat GPT peut également vous guider pour certaines manipulations simples !

From:
https://wiki-sinp.cbn-alpin.fr/ - **CBNA SINP**

Permanent link:
**https://wiki-sinp.cbn-alpin.fr/procedures/sinp-aura/tutoriels/cms/wp_elementor?rev=1753434745**

Last update: **2025/07/25 09:12**