

# Concepts de site Web et de codage

Ce document présente ce qu'est un site web, comment il fonctionne, les principales technologies web et comment ces concepts s'appliquent au projet actuel d'édition de newsletters KYOCERA.

## Qu'est-ce qu'un site Web ?

Un site Web est un ensemble de pages et d'actifs interconnectés diffusés sur Internet et affichés par un navigateur Web. Il peut être statique (fichiers prédéfinis) ou dynamique (le serveur génère du contenu à la demande).

## Comment fonctionnent les sites Web

- 1) Vous entrez une URL.
- 2) Le navigateur envoie une requête HTTP/HTTPS à un serveur.
- 3) Le serveur répond avec HTML, CSS, JS et médias.
- 4) Le navigateur analyse HTML, applique CSS pour la mise en page et le style, et exécute JavaScript pour l'interactivité.

## Technologies clés

1. **HTML** : Structure et contenu des pages (titres, paragraphes, images, liens, formulaires).
2. **CSS** : Présentation et mises en page réactives (couleurs, typographie, grilles).
3. **JavaScript** : Comportement et interactivité (événements, mises à jour du DOM, persistance des données).
4. **Côté serveur (PHP)** : gère l'enregistrement, le chargement et les opérations sécurisées sur le serveur.

## Frontend vs Backend

**Frontend (côté client)** : s'exécute dans le navigateur à l'aide de HTML/CSS/JS. Se concentre sur l'UX, le rendu et les interactions.

**Backend (côté serveur)** : Fonctionne sur un serveur (par exemple, PHP). Se concentre sur le traitement des données, la persistance, l'authentification et les E/S de fichiers sécurisées.

## Application des concepts dans ce projet

1. `creatNewsletter.html` : Point d'entrée qui charge l'interface utilisateur de l'éditeur et la structure de la page.

2. `editor.js` : logique d'édition de base, commandes de texte enrichi, insertion de médias, historique/annulation et enregistrement automatique.
3. `styles.css` : Styles globaux et mise en page réactive pour l'éditeur et la prévisualisation.
4. `preview.html` / `preview.js` : Affiche un aperçu en direct ou à la demande de la sortie de la newsletter.
5. `save_file.php` / `upload_image.php` : points de terminaison côté serveur utilisés pour enregistrer les newsletters et télécharger des médias sur le disque.
6. `TypeNews/` : Modèles de section (par exemple, edito, branding, com\_actus, animation) pour composer le contenu de la newsletter.
7. `Images/` et `media/` : Dossiers de ressources (images, icônes, vidéos) utilisés dans l'éditeur et newsletters générées.

## Comment les pièces du puzzle s'assemblent

1. Vous modifiez le contenu dans le navigateur (structure HTML, stylisé par CSS).
2. JavaScript dans `editor.js` met à jour le DOM, gère l'historique et conserve les modifications (local/serveur).
3. PHP côté serveur enregistre les fichiers dans le chemin configuré afin qu'ils puissent être rechargés ou partagés.

## Exécution locale

1. Ouvrez-creatNewsletter.html directement dans un navigateur pour accéder aux fonctionnalités réservées aux clients.
2. Pour l'enregistrement sur le serveur, placez le projet dans un serveur Web local (par exemple, XAMPP/WAMP) et accédez-y via `http://localhost/` afin que les points de terminaison PHP ( `save_file.php` , `upload_image.php` ) puissent s'exécuter.

## Apprentissage supplémentaire

1. MDN Web Docs (HTML, CSS, JS) — Référence complète des normes Web.
  - Web.dev — Bonnes pratiques en matière de performances, d'accessibilité et de fonctionnalités Web modernes.