

Introduction aux frameworks PHP : Développer plus rapidement

PHP est un langage de programmation côté serveur largement utilisé pour le développement d'applications web. Cependant, la gestion de projets complexes peut devenir fastidieuse sans une structure claire. C'est là qu'interviennent les **frameworks PHP** : des outils puissants qui organisent le code, accélèrent le développement et améliorent la maintenabilité des applications.

Dans cet article, nous explorerons en détail plusieurs frameworks PHP et micro-frameworks populaires, leurs caractéristiques, et leurs avantages pour vous aider à choisir celui qui convient le mieux à vos besoins. Nous vous offrons des formations pour chacun des Framework présent dans cet article.

Pourquoi utiliser un framework PHP ?

Les frameworks PHP ne se contentent pas de simplifier le développement ; ils offrent aussi des outils intégrés pour résoudre des problèmes récurrents et se concentrer sur la logique métier de votre application. Voici quelques avantages :

- **Automatisation des tâches courantes** : Gestion des routes, sécurité, requêtes SQL, validation des données, et plus encore.
- **Sécurité accrue** : Les frameworks intègrent des protections contre les vulnérabilités courantes telles que les injections SQL, les attaques XSS et CSRF.
- **Communauté active** : Avec des milliers de développeurs, les frameworks offrent une documentation riche et des ressources pour apprendre rapidement.
- **Meilleure organisation grâce au modèle MVC** : La plupart des frameworks PHP suivent l'architecture MVC (Modèle-Vue-Contrôleur), qui rend le code plus lisible, plus facile à maintenir et tester et plus évolutif pour des projets complexes.
- **Facilite le travail en équipe** : Les frameworks assurent une structure commune au code, ce qui facilite la collaboration au sein des équipes.
- **Performances optimisées** : Certains frameworks comme Phalcon sont spécialement conçus pour offrir des performances élevées en consommant moins de ressources. Même les frameworks généralistes incluent souvent des outils pour optimiser le caching, la gestion des sessions et la compression de ressources.

Frameworks PHP

1. Laravel

[Formation Laravel](#)

Laravel est sans doute le framework PHP le plus populaire aujourd'hui. Il propose un écosystème complet avec des outils comme **Artisan** pour les commandes en ligne, **Eloquent** pour l'ORM, et **Blade** pour le templating.

- **Avantages :**

- Facilité d'apprentissage : Syntaxe élégante, excellente documentation, et outils puissants comme Artisan CLI.
- Écosystème complet : Laravel Forge, Envoyer, Vapor, et Eloquent ORM.
- Support natif pour les tests unitaires.
- Communauté massive : Une abondance de ressources, extensions, et forums pour les développeurs.

- **Cas d'utilisation :** Développement d'APIs, applications SaaS, sites web complexes avec des interactions dynamiques.

2. Symfony

Formation Symfony: Initiation

Symfony est un framework professionnel connu pour sa robustesse et sa modularité. De nombreux frameworks, dont Laravel, utilisent des composants Symfony.

- **Avantages :**

- Modularité : Capacité à utiliser uniquement les composants nécessaires, ou le framework complet.
- Adoption industrielle : Utilisé par des entreprises pour des applications complexes et de grande envergure.
- Base solide : De nombreux autres frameworks (comme Laravel) utilisent des composants Symfony.

- **Cas d'utilisation :** Applications web évolutives, projets nécessitant une flexibilité maximale.

3. Yii Framework

Formation Yii Framework

Yii, qui signifie « Yes It Is! », est un framework PHP open source conçu pour développer des applications de grande envergure. Il est particulièrement adapté aux projets complexes nécessitant des performances élevées. Grâce à son générateur de code intégré, **Gii**, Yii permet de gagner un temps précieux dans la création des modèles, des contrôleurs et des vues.

- **Avantages :**

- Générateur de code (Gii) puissant pour accélérer le développement et générer du code pour des modèles, des contrôleurs, et des vues.

- Support natif des requêtes AJAX.
- Intégration facile de bibliothèques tierces.
- Idéal pour les projets d'entreprise.
- Sécurité intégrée : Yii intègre des fonctionnalités de sécurité robustes.
- **Cas d'utilisation** : Applications d'entreprise, plateformes e-commerce, systèmes CRM.

4. Phalcon

Formation Phalcon PHP Framework : Initiation

Phalcon est un framework PHP unique car il est écrit en langage **C** et fonctionne sous forme d'extension PHP. Cela lui confère des performances exceptionnelles en termes de vitesse et de consommation de ressources.

- **Avantages** :
 - Performance exceptionnelle : Temps de réponse rapide grâce à son architecture bas niveau.
 - Faible utilisation de mémoire.
 - Fonctionnalités robustes telles que l'ORM, le système de cache et le moteur de modèles.
 - Microservices et modularité : Phalcon offre une architecture modulaire qui permet de développer des **microservices** et des **applications modulaires**.
- **Cas d'utilisation** : Applications à haute performance nécessitant une faible latence, APIs critiques.

5. Zend Framework (Laminas)

Formation Zend Framework 2 et 3 : Initiation

Désormais connu sous le nom de **Laminas**, Zend Framework est un choix solide pour les projets nécessitant une flexibilité maximale. Sa nature modulaire permet aux développeurs d'utiliser uniquement les composants dont ils ont besoin.

- **Avantages** :
 - Architecture orientée services.
 - Extensibilité et flexibilité grâce à une conception modulaire. Les composants peuvent être utilisés indépendamment via Composer. Zend offre une liberté totale dans la manière de structurer vos projets.
 - Support pour des normes comme PSR-7 pour les middlewares HTTP.
- **Cas d'utilisation** : Projets complexes nécessitant des fonctionnalités personnalisées ou des intégrations spécifiques.

6. CakePHP

Formation CakePHP

CakePHP est l'un des frameworks PHP les plus anciens mais reste pertinent grâce à son approche **Convention over Configuration**. Il simplifie les tâches répétitives et réduit la complexité des configurations.

- **Avantages :**
 - Validation des données intégrée. CakePHP offre un système de validation puissant pour les données des formulaires.
 - Générateur de code pour démarrer rapidement. CakePHP inclut des outils pour générer rapidement des modèles, des contrôleurs et des vues, ce qui accélère le développement.
 - Communauté active avec une riche collection de plugins.
 - Rétrocompatibilité : Une attention particulière est accordée à la compatibilité entre les versions. Les migrations entre les différentes versions de CakePHP sont généralement simples, ce qui facilite la mise à jour des projets existants.
- **Cas d'utilisation :** Projets rapides nécessitant une mise en œuvre simple, solutions internes pour les petites entreprises.

Micro-Frameworks PHP : Simplicité et rapidité

1. Slim

Formation Framework Slim

Slim est un micro-framework minimaliste pour les développeurs qui recherchent une solution légère pour créer des **APIs** ou des applications simples.

- **Avantages :**
 - Léger et performant : Idéal pour des projets où la simplicité et la vitesse sont essentielles.
 - Facilité d'intégration : Peut être combiné avec d'autres bibliothèques ou composants pour étendre ses fonctionnalités.
 - Communauté croissante : De plus en plus utilisé pour des applications backend et des APIs.
- **Cas d'utilisation :** APIs RESTful, applications microservices.

2. Silex

Formation Framework Silex

Conçu par les créateurs de Symfony, Silex utilise ses composants tout en gardant une architecture simplifiée.

- **Avantages :**
 - Basé sur des composants éprouvés.
 - Parfait pour les prototypes ou les applications simples.
- **Cas d'utilisation :** Développement rapide d'applications web ou de services légers.

3. Lumen

Formation Framework Lumen

Lumen, conçu par l'équipe Laravel, est un micro-framework spécialisé pour les **APIs performantes**. Il conserve les bases de Laravel tout en étant plus rapide et léger.

- **Avantages :**
 - Basé sur Laravel, donc facile à migrer vers Laravel.
 - Idéal pour les performances grâce à une empreinte mémoire réduite.
- **Cas d'utilisation :** Microservices, backends pour applications mobiles.

Comparaison des frameworks PHP

Voici un tableau comparatif détaillé pour vous aider à choisir :

Framework	Points forts	Cas d'utilisation idéal
Yii	Performance, générateur de code	Projets d'entreprise complexes
Phalcon	Vitesse, faible consommation de mémoire	Applications critiques
Zend/Laminas	Modulaire, robuste	Projets personnalisés complexes
CakePHP	Simplicité, démarrage rapide	Projets légers
Laravel	Moderne, riche en outils	Applications SaaS, APIs complexes
Symfony	Extensible, modulaire	Projets sur mesure
Slim	Minimaliste, rapide	APIs RESTful
Silex	Basé sur Symfony	Prototypes, applications légères
Lumen	Performant, basé sur Laravel	Microservices, APIs performantes

Les frameworks PHP sont des outils incontournables pour organiser vos projets et améliorer leur qualité. Le choix du framework dépendra de vos besoins :

performance, flexibilité, ou rapidité. Prenez le temps d'évaluer vos exigences avant de vous lancer.

Formation PHP : Sécurisez vos développements Web

Projet Web

Les technologies web
Le réseau Internet et ses protocoles
Définir l'architecture
Les architectures applicatives
Web 2.0

Présentation sécurité Web

Connaitre les menaces et risques
Types d'attaques
Risques de sécurité
OWASP et WASC

Développer une application web avec PHP

Construire la structure du projet web avec PHP
Connaître l'environnement de développement
Penser objet avec PHP
Tester son application web

Failles de type XSS(Cross Site Scripting)

Qu'est qu'une attaque de type XSS ?
Démonstration d'attaque par type XSS
se protéger des attaques par type XSS ?
Utiliser un script pour contrôler

Bases de données, Failles de type SQL Injection

Qu'est qu'une attaque par injection SQL ?
Démonstration d'attaque par injection SQL
se protéger des attaques par injection sql ?

échapper les paramètres de données

Utiliser PDO

Mesures de sécurité pour MySQL

Faibles de type CSRF/Formulaires PHP

Qu'est qu'une attaque de type CSRF ?

Démonstration d'attaque par type CSRF

se protéger des attaques par type CSRF ?

Utiliser des jetons.

Faire un rappel sur les formulaires et les méthodes

Utiliser des captchas

Validation et limitations de l'approche JavaScript

Attaques HTTP et Ajax

Expressions régulières

Upload-Faibles et contre-mesures

Sécurité cookie et session

sécurité des cookies

Se faire voler une session

Protéger simplement les sessions

Modification de la technique de sauvegarde des sessions

Régénération d'identifiant de session

Sécurité serveur Web

Concepts de sécurité des serveurs.

faible de configuration

directives php.ini

directives httpd.conf

Protection des dossiers par htaccess

Droits d'accès des dossiers sur le serveur web

Limiter l'accès à certaines URL de l'application: robots.txt

Formation en lien combinée avec cybersécurité

[Formation cybersécurité – Applications web et cybersécurité](#)

Formation Tests Unitaires PHPUnit

Introduction Test Unitaire

Définition des tests unitaires

Le test-driven development (TDD)

Trouver des ressources sur les tests unitaires

PHPUnit

Installer PHPUnit via composer

Configurer PHPUnit

Les assertions

Utiliser la ligne de commande (CLI)

Configurer dans un fichier .XML

Connaître les options de la console

Ecrire des tests

Découvrir les assertions

Définir les dépendances de test

Ajouter un data provider

Utiliser le setUp() des tests

Skipper un test

Tester les exceptions

Stubs

Mocks

Couverture du code

Exclure un dossier de la couverture

Exclure des bouts de code

TDD

Formation Utile

[Formation PHP: la plus complète](#)

Formation PHP | Programmation orientée Objet – MVC

Introduction à la formation PHP

Environnement de développement

Présentation des IDEs, installation d'un IDE

Présentation et installation du débogueur

Programmation orientée Objet(P00)

principes de la P00

La syntaxe objet avec PHP

Classes et objets

constructeurs

destructeurs

Les attributs et méthodes statiques

L'encapsulation

les méthodes magiques

Surcharge

Le référent d'un objet

Atelier pratique: *Mise en oeuvre des différents principe de la P00 à travers de multiples exercices pratiques*

Rélations entre classe

L'héritage en P00

Aggrégation faible

Composition

L'envoi de messages objets

gestion des erreurs/gestion des exceptions

gestions des erreurs

gestions des exceptions

Autres concepts de la P00

Le polymorphisme

Les interfaces

Se connecter avec une base de donnée avec PDO

Présentation de PDO

Les classes de PDO

Traitement des erreurs

Requêtes préparées

Les transactions

Atelier pratique: *accéder à notre base de donnée avec PDO*

Architecture MVC

Fonctionnement du MVC

Controller

Vue

Modèle

Notion de route

Notion de template

Atelier pratique: *Mise en place d'un système MVC avec PHP*

Formation suggérée pour poursuivre votre apprentissage

[Formation PHP : Sécurisez vos développements Web](#)

Formation PHP : la plus complète

Introduction à la formation PHP

Apprendre les bases de PHP

Pourquoi utiliser PHP ?
structure d'une page PHP

Insertion de code PHP dans un document html

les types de variables

Déclaration et utilisation de tableaux

les constantes

les opérateurs

les boucles et conditions

variables superglobales

recupérer les données d'un formulaire

envoyer un email

Atelier pratique: *Revue des balises principales HTML et des commandes de style, mise en place d'une page modèle*

Création de fonctions, utilisation de fonctions prédéfinies

Déclaration de fonctions

Les paramètres et l'instruction return

Création de bibliothèques de fonctions

Fonctions prédéfinies : exemples, organisation, documentation

Fonctions pour les dates, tableaux, chaînes de caractères

Fonctions include et require

Atelier pratique: *Réalisation de fonctions personnalisées et utilisation de fonctions prédéfinies*

Traiter les données d'un formulaire

création d'un formulaire

Récupérer les données du formulaire

Upload de fichiers : la variable \$_FILES et les fonctions associées

envoyer un email

Atelier pratique: *Création d'un formulaire de candidature*

Interagir avec l'utilisateur

Récupérer et utiliser des paramètres GET

utiliser les cookies pour stocker des informations

Upload de fichiers : la variable \$_FILES et les fonctions associées

envoyer un email

Atelier pratique: *Création d'un formulaire de candidature*

PhpMyAdmin

Présentation de phpMyAdmin

Créer une base avec phpMyAdmin

Créer des tables avec phpMyAdmin, voir les différents types de champs.

Modifier la structure d'une table

Atelier pratique: *Création d'une base MySQL et création de tables*

SQL

Sélectionner des enregistrements (SELECT)

Insérer des enregistrements (INSERT INTO)

Modifier des enregistrements (UPDATE)

Supprimer des enregistrements (DELETE)

Opérateurs et critères.

Atelier pratique: *Ecriture de requêtes SQL simples*

Utiliser une base de données MySQL avec PHP

Présentation des SGBD

Types de données

Connection au serveur MySQL

Remplissage de la base à partir d'un formulaire.

Ajouter des données dans la base avec PHP

Modifier des données dans la base avec PHP

Supprimer des données dans la base avec PHP

Protéger les données à envoyer à MySQL

Atelier pratique: *Enregistrement en base des données de notre formulaire de candidature*

Cookies & sessions

Présentation des sessions

Avantages et inconvénients des Cookies et Sessions.

stocker et récupérer des variables de session

stocker et récupérer des variables de cookie

Atelier pratique: *Réalisation d'un panier d'achat simple, création interface utilisateur*

Manipuler les fichiers

Ouverture et écriture de fichier

SQL: Notions avancées

Tables et relations

Jointures: INNER, LEFT et RIGHT JOIN

Atelier pratique: *Faire des requêtes SQL sur plusieurs tables*

création d'un mini CMS

Les concepts du CMS

Les CMS existants

Introduction à la modélisation de données

Création de l'interface publique

Création de l'interface privée

Gestion des utilisateurs

Atelier pratique: *Application de gestion d'une bibliothèque*

XML

Présentation du format XML

Ecrire un document XML

introduction à XML

Atelier pratique: *Création d'un flux RSS, création d'un sitemap pour notre bibliothèque*

Pour faire suite à cette formation

[Formation PHP : Sécurisez vos développements Web](#)

Formation PHP/MySQL: création d'un système de gestion de contenu Web

Introduction à la formation PHP/MySQL

Utilisation avancées de PHP

Améliorer le traitement des formulaires

Les expressions régulières

utiliser les fonctions de filtres pour valider une donnée

Protéger un dossier avec un .htaccess

Apprendre à utiliser la documentation PHP!

Créer des images en PHP

Atelier pratique: *Validation avancée de formulaire, Créer une image dynamique en php*

Cookies & sessions

Présentation des sessions

Avantages et inconvénients des Cookies et Sessions.

stocker et récupérer des variables de session

stocker et récupérer des variables de cookie

Atelier pratique: *Réalisation d'un panier d'achat simple, création interface utilisateur*

Manipuler les fichiers

Ouverture et écriture de fichier

SQL: Notions avancées

Tables et relations

Jointures: INNER, LEFT et RIGHT JOIN

Atelier pratique: *Faire des requêtes SQL sur plusieurs tables*

création d'un mini CMS

Les concepts du CMS

Les CMS existants

Introduction à la modélisation de données

Création de l'interface publique

Création de l'interface privée

Gestion des utilisateurs

Atelier pratique: *Application de gestion d'une bibliothèque*

XML

Présentation du format XML

Ecrire un document XML

introduction à XML

Atelier pratique: *Création d'un flux RSS, création d'un sitemap pour notre bibliothèque*

Formation PHP/MySQL : créer un site Web dynamique

Introduction à la formation PHP/MySQL

[Rappel html/css](#)

Architecture client/server

Outils de développement: installer un server local, un IDE

Atelier pratique: *Installation des outils de développement*

Les bases pour apprendre le PHP

structure d'une page PHP

Insertion de code PHP dans un document html

les types de variables

Déclaration et utilisation de tableaux

les constantes

les opérateurs

les boucles et conditions

Atelier pratique: *Revue des balises principales HTML et des commandes de style, mise en place d'une page modèle*

Création de fonctions, utilisation de fonctions prédéfinies

Déclaration de fonctions

Les paramètres et l'instruction return

Création de bibliothèques de fonctions

Fonctions prédéfinies : exemples, organisation, documentation

Fonctions pour les dates, tableaux, chaînes de caractères

Fonctions include et require

Atelier pratique: *Réalisation de fonctions personnalisées et utilisation de fonctions prédéfinies*

Traiter les données d'un formulaire

création d'un formulaire

Récupérer les données du formulaire

Atelier pratique: *Création d'un formulaire de contact*

Formation en Lien : [PHP/MySQL: création d'un système de gestion de contenu WebPHP/MySQL](#)

Interagir avec l'utilisateur

Récupérer et utiliser des paramètres GET

utiliser les cookies pour stocker des informations

envoyer un email

Atelier pratique: *Validation d'un formulaire de contact*

PhpMyAdmin

Présentation de phpMyAdmin

Créer une base avec phpMyAdmin

Créer des tables avec phpMyAdmin, voir les différents types de champs.

Modifier la structure d'une table

Atelier pratique: *Création d'une base MySQL et création de tables*

SQL

Sélectionner des enregistrements (SELECT)

Insérer des enregistrements (INSERT INTO)

Supprimer des enregistrements (DELETE)

Opérateurs et critères.

Atelier pratique: *Ecriture de requêtes SQL simples*

Utiliser une base de données MySQL avec PHP

Présentation des SGBD

Types de données

Connexion au serveur MySQL

Remplissage de la base à partir d'un formulaire.

Ajouter des données dans la base avec PHP

Supprimer des données dans la base avec PHP

Protéger les données à envoyer à MySQL

Atelier pratique: *Enregistrement en base des données de notre formulaire de contact*

Pour bonifier votre apprentissage

[Formation PHP | Programmation orientée Objet – MVC](#)

Formation MariaDB – administration

Module 1. Présentation de MariaDB

- Vocabulaire et concepts communs aux SGBDR (tables, requêtes SQL, transactions, ...)
- Comprendre la signification d'une licence GPLv2 pour les applications MariaDB
- Impacts des changements de versions sur les applications
- Un modèle client-serveur traditionnel (mysql, mysqld)
- Les outils d'administration (PHPMyadmin, DBvizualizer, ...)

Module 2. Installation de MariaDB

- Installations (binaire, à partir des sources) du serveur et du client MariaDB
- Principaux fichiers de configuration
- Arrêt et démarrage du service MariaDB
- Test de connexion

Module 3. Architecture des tables

- Vue générale du fonctionnement de MariaDB en mémoire
- Liens entre mémoire et stockage des données dans MariaDB
- La table, objet de base pour MariaDB
- Les autres objets (index, vues, ...)
- Comprendre l'impact du choix d'un type de table (MyISAM, InnoDB, etc.)
- Création, modification et suppression de tables
- Les informations disponibles sur les tables (schemata, show tables, ...)
- Intérêts du partitionnement des tables (horizontal, vertical)
- Mise en œuvre du partitionnement par intervalles avec RANGE

Module 4. Utilisateurs, connexions et sécurité

- Principes d'authentification
- Les privilèges sur les objets proposés par MariaDB
- Bien comprendre la notion de couple (utilisateur,@IP) dans les règles de sécurité
- Précautions avec le wildcard '%'
- Les commandes du quotidien (GRANT, SHOW, SET PASSWORD, etc.)
- Comprendre l'effet « cascade » de la clause WITH GRANT OPTION
- Suppression de droits (REVOKE)
- Renommer un compte (RENAME)
- Apport des rôles dans la sécurité

Module 5. Compléments sur le moteur de stockage

- La richesse des plugins
- Fonctionnement transactionnel du moteur XtraDB/InnoDB
- Configurer la taille de l'espace INNODB
- Les journaux binaires : activation, lecture, rotation, purge, ...
- Verrouillage des tables
- Réglages de l'isolation et de la durabilité
- Configuration du serveur : fichiers auto-extensibles, compression transparente
- Cache de requêtes et cache d'index MyISAM/Aria

Module 6. Sauvegardes et restaurations

- Les différents types de sauvegardes (logique, physique, incrémentale, complète)
- Définir une stratégie claire de sauvegarde
- Journaux binaires et sauvegardes
- Sauvegarde des différents moteurs : MyISAM, InnoDB
- Informations sur les bases et les tables (show)
- Les outils disponibles (mysqldump, phpmyadmin)
- Commandes pour « réparer » les tables (check, analyze, optimize, repair, backup, ...)
- Mise en œuvre des sauvegardes et des restaurations avec mysqldump

- Restauration et PITR (Point IN TIME RECOVERY)
- Import/export manuelle des bases
- Les commandes mysqldump et mysqlrestore
- Sauvegardes distantes
- Réplication master – Slave

Module 7. Maintenance de MariaDB

- Paramétrage de sa session de travail (set global, set session, ...)
- Les variables de session (@@, @, set, select)
- Comprendre les problèmes d'encodage (UTF-8, etc.)
- Changer et/ou définir les jeux de caractères
- Chargement de données avec différents outils (load data, select into outfile, ...)
- Les journaux utiles (infos générales, erreurs, performances, ...)
- Informations globales sur les bases (information-schema)
- Statistiques sur les tables
- Problèmes de connexions classiques (daemon, port, droits)
- Terminer une session (kill)

Formation POSTGRESQL : Essentiels pour développeurs

Présentation de PostgreSQL

- Serveur de base de données relationnelle objet
- Historique
- Licence
- Mode de développement
- La version de PostgreSQL 16
- Les composants du serveur PostgreSQL

Installation et configuration

1. Installation de PostgreSQL
2. Installation avec un système de packaging

Interagir avec le serveur

1. Le concept du client serveur
2. Le client psql
3. Outils en ligne de commande et graphique. psql (shell SQL). PgAdminIII, phpPgAdmin, Pgpool

Définition des données

1. Création de schémas, tables, index, domaines...
2. Types de données,
3. Opérateurs et fonctions
4. Comparaison PostgreSQL/Oracle (Type de données, type d'objets)

Programmation côté serveur

1. Importation et exportation de données
2. Intérêt de la programmation côté serveur et ses limites
3. Les « rules »
4. Mise en oeuvre de déclencheurs (triggers)
5. Programmation de Fonctions et procédures stockées
6. Déclaration et utilisation de variables
7. Gestion des curseurs
8. Gestion des exceptions
9. Gestion des tâches planifiées (pg_cron)
10. Mise en oeuvre avec Pl/pgSQL et présentation des autres outils disponibles
11. PostGIS (Présentation, représentation et stockage des géométries, indexation, métadonnées, etc.)

Partitionnement sous PostgreSQL

1. Mise en œuvre du Partitionnement

La gestion de la concurrence d'accès et des transactions sous PostgreSQL

1. Transactions
2. Verrous
3. Isolation

Optimisation SQL

1. Anatomie de l'exécution d'une requête
2. Les composants du moteur
3. Estimation des coûts
4. Méthodes d'accès aux données
5. Méthodes de jointures
6. L'outil Explain
7. Démarche d'optimisation

Les codes informatiques et les infolettres : combiner les deux pour optimiser son marketing?

Dans le cadre d'une infolettre, vous devez savoir accrocher votre lecteur, les internautes qui vous liront devront avoir envie de vous lire. C'est ce qui permet de vous démarquer des concurrents qui distribuent aussi des courriels d'infolettres. Mais pourtant, avoir un beau contenu n'est pas tout! Vous devez aussi disposer judicieusement des images, des styles de votre texte (polices de caractère), puis avoir un bon logiciel d'infolettres pour avoir des styles les plus présentables à vos interlocuteurs.

Il peut arriver un temps où vous aimerez aller plus loin dans la personnalisation de vos courriels. Mais comment faire? Vous aurez certainement des manipulations à faire dans les styles prédéfinis. Avant de pouvoir créer votre propre style, il vaut mieux commencer par calquer un modèle libre de droit pour le modifier et le faire correspondre avec votre propre image de marque. Ceci demande du travail surtout si vous débutez dans la création de modèles d'infolettres.

Quand un créateur d'infolettres désire avoir le plein contrôle sur ce qu'il publie, il lui est recommandé d'apprendre au moins des [bases des codes informatiques](#). Si une terminologie réunit bien plusieurs codes informatiques, il s'agit de la terminologie dite HTML. [Le HTML est un peu la base](#) à la base des codes informatiques plus poussés tels que [les codes PHP](#) (partie intégrante de WordPress) et Javascript. C'est aussi une bonne base pour personnaliser des infolettres si vous avez déjà quelques notions puis que vous aimeriez les mettre en pratique. Avant d'avoir ces notions, un peu complexes, vous devez évidemment les apprendre, via une quelconque formation ou de façon autodidacte, en tout ou en partie.

Le marketing par infolettres et le HTML : quel lien peut être fait?

Pour vous, relier HTML et infolettre revient à utiliser le HTML non pour créer un site web ou monter des lignes de code hyper complexes dans le but de gérer un réseau de technologies de l'information; plutôt, vous allez maîtriser un compte de messagerie dit « open source » pour manier habilement le style entier d'un courriel et de ses envois aux différents contacts intéressés. Concrètement, vous pourrez créer des courriels plus personnalisés qu'en utilisant seulement un logiciel de messagerie où les modèles sont très peu personnalisables pour les personnes à l'œil plus aiguisé. Il est ensuite important d'avoir une excellente maîtrise du rendu,

et de pouvoir le tester AVANT d'envoyer vos courriels. En effet, toute personne qui emploie la terminologie HTML et qui est soucieuse des résultats que verront les autres gens va impérativement vérifier le résultat AVANT la diffusion. Recevoir un courriel avec des caractères spéciaux mal placés et pratiquement illisibles est frustrant, d'autant plus que ce genre de courriels se retrouve souvent à la corbeille. Il y a des façons de créer un courriel et d'en tester le visuel. Ces techniques de travail se trouvent [dans les cours de terminologie informatique de base](#). Apprendre ces notions peut paraître obscur si jamais vous ne connaissez rien à la terminologie. Par contre, c'est en mettant en pratique les notions à votre rythme que vous arriverez à maîtriser la terminologie.

HTML : ce que le HTML demande comme structure

Bien que la terminologie HTML soit bien simple pour un spécialiste des technologies de l'information, il y a bien des chances qu'une simple omission dans la structure d'un document HTML vienne le rendre inutilisable. [À quoi ressemble un document HTML?](#) Il s'agit d'une feuille où vous inscrivez les éléments terminologique requis et qui renvoie un résultat quand vous la rendez disponible à d'autres utilisateurs en ligne.

C'est important, rendre un document de ce type avec une structure précise. Par exemple, si vous placez ce qui devrait être à la première ligne du document à la dixième ligne, votre document, automatiquement, deviendra inutilisable, sauf si vous le corrigez par la suite et le remettez en ligne. Ce qui est tout de même plutôt choquant, surtout pour un projet d'envergure. C'est pourquoi nous conseillons, pour tous projets, de se doter d'une application ou d'un logiciel qui permet de tester le résultat avant de l'envoyer en ligne ou le rendre disponible à un groupe.

Les balises sont ce qui contrôle votre terminologie

Si la terminologie d'un document informatique était aussi malléable que le texte d'un roman ou d'un article de journal, il y aurait trop de risques que n'importe qui puisse publier absolument n'importe quoi et que n'importe qui puisse pirater un document. Le web serait une véritable cacophonie en quelque sorte! C'est pourquoi la structure du document, déjà mentionnée plus haut, est d'une assez haute importance.

Au début et à la fin de chaque ligne du document, vous devez placer des balises. Sans quoi, ce qui est contenu dans la ligne suivante et qui pour vous est quelque chose d'autre, pour le HTML, sera exactement la même ligne terminologique. Ou, pire, en omettant une balise, vous pourriez obtenir une partie de votre document qui sera illisible ou dont la présentation pourra être très étrange.

Chaque balise commence par « < » et se termine par « > ». C'est assez logique, puisque nous ouvrons le crochet pour mentionner quelque chose, et

nous le quand notre propos est terminé. De la même façon, ce que nous appelons une « ligne codée » dans un document comprend une première balise en « < » et se termine par une balise semblable, mais avec le caractère suivant « / ». Le caractère « /> » signifie « terminé » en quelque sorte.

Les balises permettent aussi de définir un code qui s'insère dans le HTML. Par exemple, insérer un code Javascript (pour animer votre site web et le rendre présentable sur tous les appareils, autant les ordinateurs que les iPhones), vous insérerez une balise avant votre code Javascript, avec une autre balise suivant ce code. Autrement, sans la balise de fin, votre animation ne sera pas prise en compte. Si ces quelques notions vous sont très peu familières, apprendre d'une de vos connaissances qui travaille en informatique ou qui connaît la terminologie HTML vous permettra de vous initier à ce monde un peu différent de l'écriture normale ou des mathématiques.

Pour en revenir aux infolettres, quand vous créez votre premier modèle, grâce au HTML, vous irez chercher une touche de personnalisation que vous n'aviez pas auparavant. En effet, parce que la terminologie HTML permet à elle seule de créer de façon presque illimitée, vous pourrez vous amuser avec vos premiers designs d'infolettres, mais il est recommandé de le faire à l'aide d'un « éditeur de code ». Le plus connu étant probablement [Visual Studio Code](#). Rendez vos infolettres attrayantes et collez-les à votre propre marque grâce à la terminologie HTML.

Bien dessiner un modèle d'infolettre pour bien la schématiser par la suite

Avoir un croquis, dessiner un plan de ce que vous aimeriez faire voir aux lecteurs de vos courriels, représente une méthode de travail couramment utilisée. Pourquoi les créateurs vont-ils, environ 90% du temps, construire un croquis, ou le faire construire, avant d'en écrire le contenu HTML? Pour la même raison, en fait, qu'un écrivain utilisera un plan avant d'écrire son prochain roman. Vous avez déjà [visuellement devant vous les éléments à mettre en place](#) pour votre visuel avant de vous aventurer dans la structure de ceux-ci à l'intérieur de votre éditeur de code. Puisque c'est une tâche difficile et assez abstraite à faire, monter un modèle d'infolettre à partir de HTML, mieux vaut avoir en tête déjà le résultat voulu pour votre premier chef-d'œuvre! En tant qu'artiste de l'infolettre, il est avantageux d'avoir sur papier un plan de votre infolettre, comme une entreprise de construction ne peut improviser la construction d'une maison. Construire une maison est toujours basé sur les plans d'un architecte, comme construire une infolettre est (presque) toujours basé sur le talent et l'habileté d'un graphiste.

De façon concrète, essayez de respecter le plus possible le plan qui vous est fourni. Autrement, vous risquez de dériver vers un contenu qui ne ressemble pas du tout à ce qui est recherché. De cette manière de travailler repose un risque : vous n'aurez pas produit ce que vous deviez produire, mais plutôt ce

que vous aimeriez faire. Il y a une grande différence entre faire ce que l'on VEUT faire, et se limiter à ce que l'on PEUT faire, cela s'applique même dans les lettrines d'une infolettre. Ce que l'on veut produire comme résultat n'est pas nécessairement ce que le graphiste nous permet de faire, puisque chacun est le professionnel de son métier, comme le graphiste est le professionnel censé reconduire votre courriel vers une place de choix dans la boîte aux lettres virtuelle Outlook de vos clients!

La police de caractère fait vraiment une différence?

Il existe plus d'une centaine de polices de caractères toutes différentes que vous pouvez placer dans une infolettre. Autant vous aimerez avoir la bonne police de caractère, autant elle sera difficile à trouver, cette police de caractère parfaite. Pourtant, est-ce que cela fera une vraie différence sur le rendu final, avoir une police de caractère à tout casser? Pour mettre tout cela au clair, il est important de se poser la question suivante : « Si j'avais mis une autre police, est-ce que cela aurait changé la perception de mes clients? » Si la réponse est non, c'est que votre police est bonne. Rien ne sert de perdre plusieurs jours à trouver une police de caractère parfaite, mais encore faut-il considérer que plusieurs autres choses peuvent être mises en place pour créer le courriel le meilleur possible.

Disposer vos paragraphes comme dans un tableau, permet de créer des textes à deux ou trois colonnes. Les images que vous allez présenter, rendez-les attrayantes et à propos! Vous gagnerez ainsi beaucoup plus de clients potentiels que si vos images ne sont pas tout-à-fait bien calibrées. Si elles sont trop sombres ou trop éclairées, les images ne permettent pas de faire valoir tout ce qu'elles peuvent délivrer comme message. Pour le web, il est vrai de dire qu'une image vaut mille mots.

Les lecteurs de votre infolettre liront en premier ce qu'ils voient, et non le texte en soi. Les gens peuvent lire un texte environ 15 secondes et s'y désintéresser si la présentation de celui-ci laisse à désirer, aussi attractif ce texte soit-il. Contrairement au roman, où le texte est l'essence même du succès, le texte sur le web repose sur l'image et le fait qu'on puisse saisir le contenu en lisant les deux ou trois premières lignes du texte et en regardant vite fait la mise en forme d'un courriel (par exemple). C'est en personnalisant votre courriel que vous arrivez à joindre votre public cible. Un modèle de courriel devrait se modeler au public qu'il veut aller chercher, aux futurs clients à qui il souhaite donner envie d'en savoir davantage. Malheureusement, aujourd'hui, plusieurs ignorent encore l'impact d'un bon et beau courriel!