













# Contexte d'utilisation des langages

# Sommaire

I. Utilisation commune.....	3
A. Sites simples.....	3
B. Sites avancés.....	5
C. Sites plus avancés.....	7

Crédits des illustrations:  
© Fotolia, Skill and You

## Les repères de lecture

 Retour au chapitre	 Définition	 Objectif(s)	 Espace Élèves	 Vidéo / Audio
 Point important / À retenir	 Remarque(s)	 Pour aller plus loin	 Normes et lois	 Quiz

## I. Utilisation commune

Nous allons, dans cette partie, voir quels peuvent être les langages les mieux adaptés dans des cas concrets.

Il est tout à fait possible d'utiliser d'autres langages pour arriver aux mêmes résultats, mais les choix suivants ont été faits en fonction du niveau de difficulté de réalisation des sites et d'utilisation des langages choisis.

### A. Sites simples

Il est entendu ici des **sites qui proposent peu de fonctionnalités, et donc, nécessiteront peu de langages pour être fonctionnels.**

Ce sont, par exemple, des sites de présentation de société, de produits ou encore de CV.

Un site vitrine peut compter autant de pages que le souhaite le développeur.

Le nombre de pages doit normalement avoir très peu d'incidence sur la complexité de réalisation du site.

Cependant, un site vitrine comportant beaucoup de pages sera plus long à créer, et surtout plus compliqué à administrer, car toutes les modifications apportées au site devront être réalisées manuellement en réécrivant les pages concernées.

Vous trouverez ci-dessous un schéma présentant la correspondance entre les écrans du site et les pages créées.

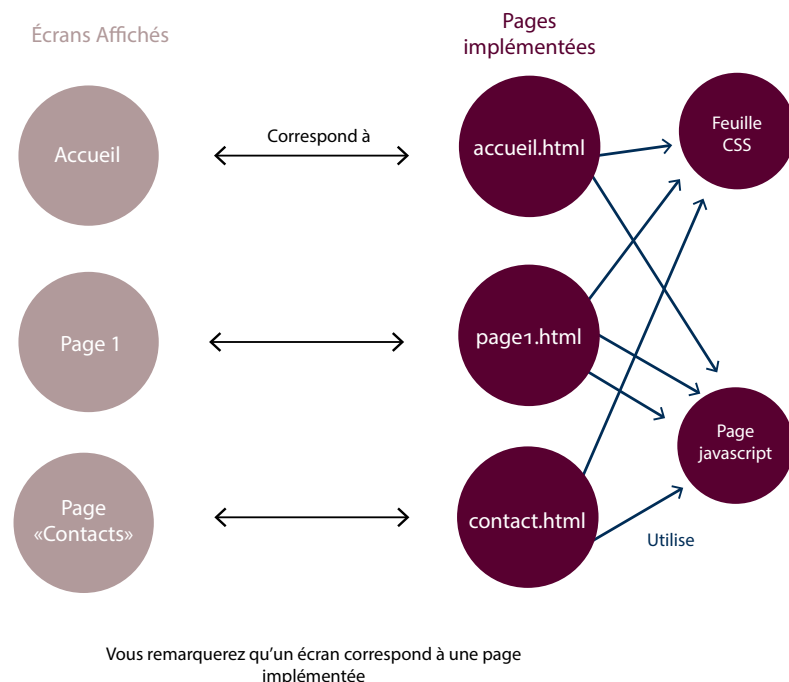


Fig. 1 Correspondances entre écrans affichés sur le navigateur et pages programmées

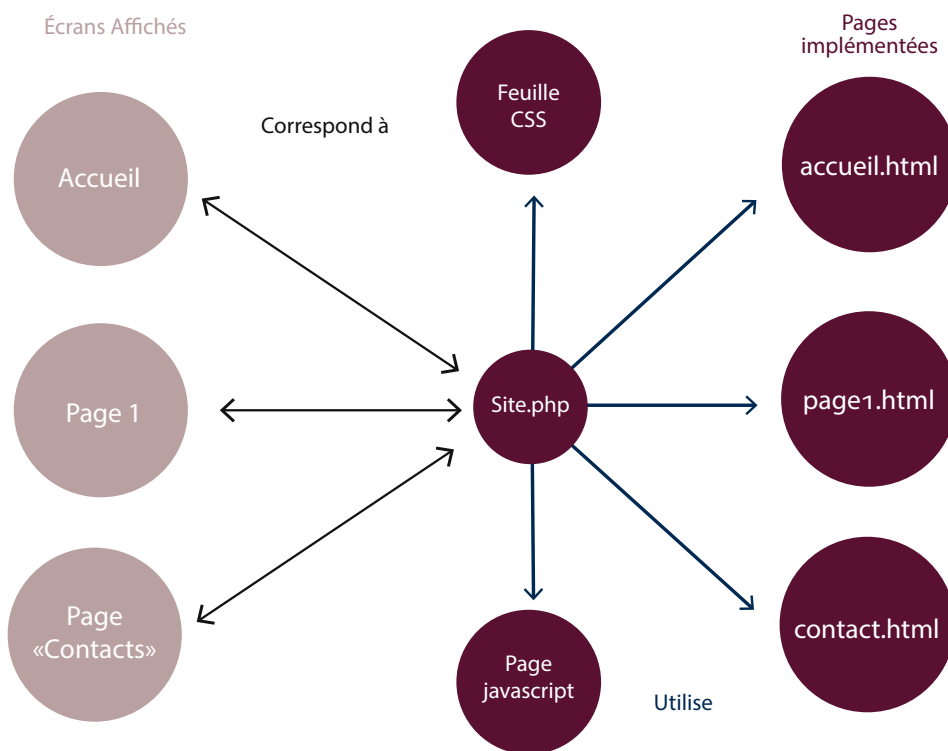
Pour ce type de sites, qui ne nécessite pas d'architecture poussée, on peut également préconiser l'utilisation du PHP comme langage serveur, et du HTML en langage client.

Le HTML en tant que langage client reste la solution la plus adaptée pour toutes les raisons que nous avons abordées plus tôt dans ce cours.

Côté serveur, notre choix pourra se porter sur le PHP principalement grâce à sa simplicité de mise en place.

Ce langage permettra de rapidement mettre en place une structure ne nécessitant pas d'avoir à écrire plusieurs fois les parties du site qui se retrouveront de page en page comme les en-têtes et pieds de page ainsi que les menus.

La structure du site que vous mettrez en place ressemblera à celle décrite dans le schéma ci-dessous :



Vous remarquerez qu'un écran correspond à une page implémentée

Fig.2 Structure d'un site simple, pages PHP, CSS et JavaScript.

## B. Sites avancés

Il est entendu ici **les sites intégrant une base de données, ce qui donnera plusieurs autres possibilités comme des formulaires de recherche, des accès restreints à certaines pages et surtout, des sections administrables via une interface.**

On peut classer dans cette catégorie les sites communautaires, les forums, les sites d'e-commerce, les blogs, les intranets...

Dans le cas des sites d'e-commerce, il est intéressant de noter que la partie paiement en ligne impose des contraintes de sécurité plus importantes que nous n'aborderons pas dans cette section.

On préconisera l'utilisation du PHP en tant que langage serveur, du HTML en tant que langage client et l'utilisation d'une base de données MySQL.

Pour ces types de sites dont le volume de connexions est moyennement important, ces solutions restent très adaptées.

Vous trouverez ci-dessous un schéma présentant la correspondance entre les écrans du site et les pages créées.

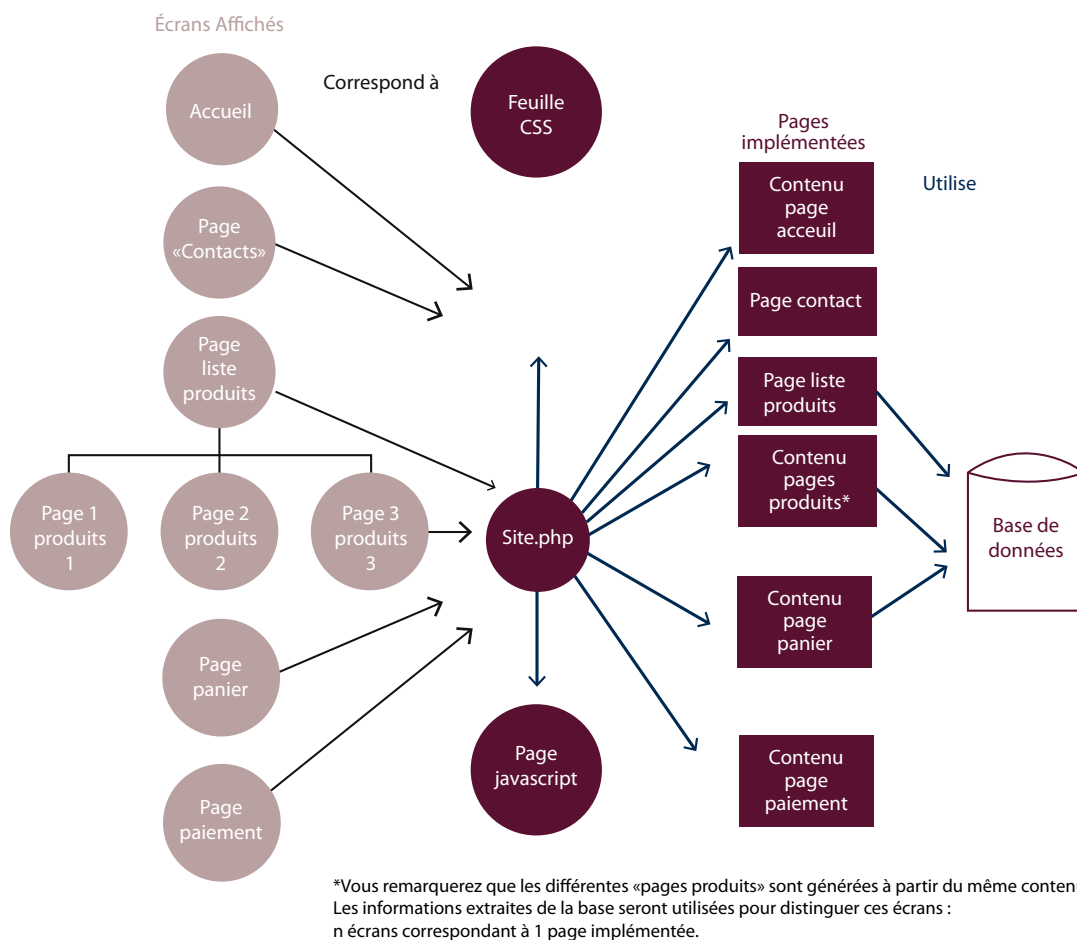


Fig.3 Correspondances entre écrans affichés sur le navigateur et pages programmées dans le cas d'un site de vente en ligne.

Vous trouverez ci-dessous quelques exemples de sites avancés :

Un site marchand

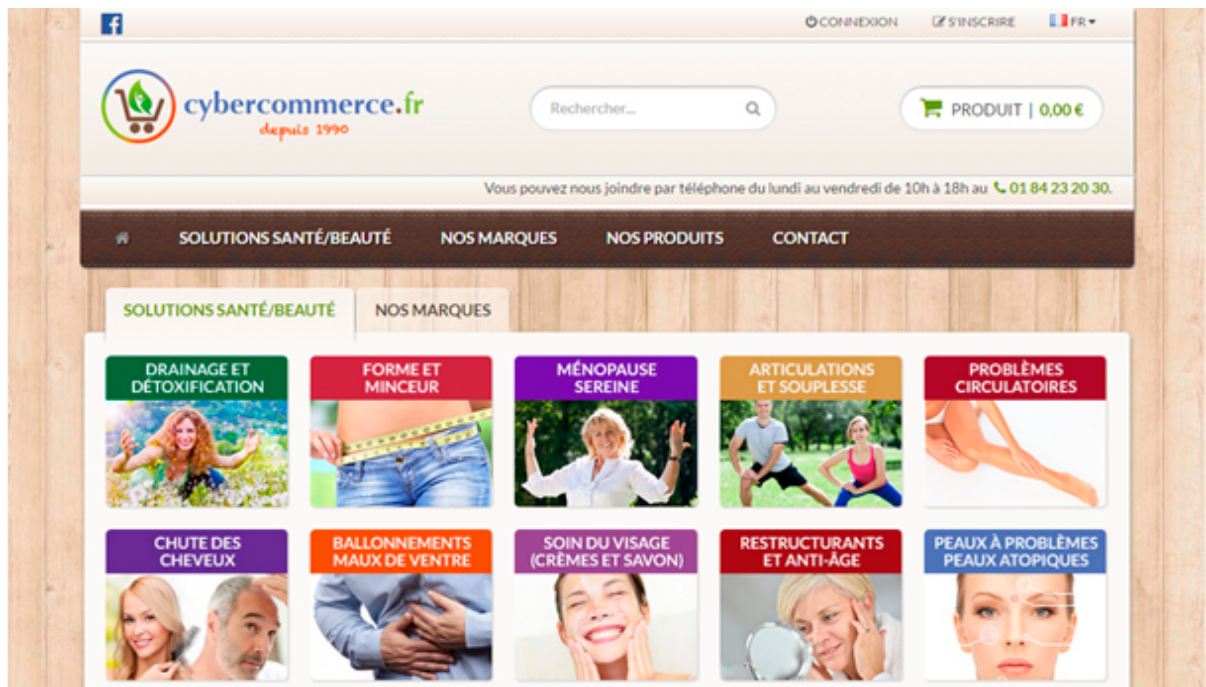


Fig. 4 <http://www.cybercommerce.fr>

Un forum



Fig. 5 <http://www.developpez.net>

Ou encore, le très célèbre : <http://fr.wikipedia.org/>

De par leur besoin de base de données, ce type de site nécessite une étude plus poussée et donc une architecture plus conséquente.

Cela dit, les sites comme « Facebook », sont des sites communautaires dont le principe, techniquement parlant, n'est pas compliqué, mais le nombre d'internautes utilisant ce service est si important qu'ils appartiennent à une catégorie encore supérieure.

## C. Sites plus avancés

En qualifiant un site de « plus avancés », nous voulons faire référence à des sites professionnels, comme les **CRM** (Customer Relationship Management : sites de gestion de clients), ou **Middle Office** (sites de gestion d'activité).

Ces applications nécessitent en général une plus grande disponibilité, un plus grand volume de données et surtout un grand nombre de traitements automatisés de données.

Ces différentes contraintes orientent naturellement le développeur vers des langages poussés, comme le Java ou le C#.

Vous rencontrerez souvent un trio de langages gagnants :

- un langage pour l'interface ;
- un langage pour les traitements serveur ;
- un langage pour accéder à une base de données.

La plupart des langages abordés dans ce cours peuvent être utilisés ensemble, mais ils sont très généralement groupés comme suit :

- **HTML / PHP / SQL** : des langages relativement simples d'approche. Ils permettent de réaliser des sites très complets, tout en étant libres de droits.
- le trio complet et gratuit, **JSP / Java / SQL** : des langages plus complexes, mais qui permettent en contrepartie plus de possibilités. Il s'agit de solution libre de droits.
- la solution intégrée, **ASP / C# / SQL** : des langages permettant globalement la même chose que le groupe précédent, mais cette fois-ci, ils sont payants.

Leur principal avantage étant qu'il existe des environnements de développement officiels, supportés par l'éditeur même du langage.

**Votre choix devra toujours être fait en prenant en compte l'objectif, les particularités, et le budget du site à développer.**