

Les différents langages web

ELAN

202 avenue de Colmar - 67100 STRASBOURG

☎ 03 88 30 78 30 📠 03 88 28 30 69

✉ elan@elan-formation.fr

www.elan-formation.fr

SAS ELAN au capital de 37 000 € -

RCS Strasbourg B 390758241 – SIRET 39075824100041 – Code APE : 8559A

N° déclaration DRTEFP 42670182967 - Cet enregistrement ne vaut pas agrément de l'Etat

SOMMAIRE

I.	<i>Introduction</i>	3
II.	<i>Les langages serveurs</i>	3
1.	PHP.....	3
2.	Java	4
3.	ASP .Net.....	4
4.	SQL	4
5.	Les autres.....	4
III.	<i>Les langages client</i>	4
1.	HTML.....	4
2.	CSS.....	5
3.	JavaScript.....	5
4.	Flash	5
5.	XML & JSON	5
IV.	<i>Les frameworks</i>	5
1.	Définition.....	5
2.	RubyOnRails.....	6
3.	Symfony.....	7
4.	jQuery.....	7
V.	<i>Cas particulier : AJAX</i>	8

I. Introduction

Pour mieux s'y retrouver dans la jungle des langages de programmation web, voici une description des principaux langages utilisés, leurs points forts, et leurs points faibles.

Si la spécialisation à outrance permet de devenir très compétent dans son milieu, elle a aussi comme malheureuse conséquence de compliquer le dialogue entre les différents corps de métiers. Comment savoir si la remarque qu'on veut faire est pertinente, si notre petite modification sur une maquette ne cache pas en réalité un travail de titan, voir même quelque chose de techniquement infaisable, si l'on ne comprend pas ce qu'elle implique?

Pour assouplir la frontière entre webdesigner et développeur web, il est important de pouvoir partager une base de connaissance commune, ainsi qu'un vocabulaire précis et bien employé.

Comment se passe l'affichage d'une page web, de façon très schématique? Le client (le navigateur de l'utilisateur, en général), demande une page. La requête est traitée par le serveur, et renvoie la réponse au client. Le client, ensuite, traite la réponse de son côté. Les langages utilisés sur le web peuvent naturellement se séparer en deux grandes catégories, en fonction de l'endroit où ils s'exécutent : les langages **côté serveur**, et les langages **côté client**.

II. Les langages serveurs

La première partie des langages que nous dirons serveur servent à traiter la demande du client et préparer la réponse à envoyer. Le code y est exécuté par le serveur, et ne pourra jamais être vu par le client.

1. PHP

Personne n'a pu passer à côté du PHP, utilisé par les plus grands CMS¹ (WordPress, Joomla, Drupal, Magento, ...). Il est très facile à déployer, a une excellente documentation, et permet de faire des choses simples très vite.

Il n'est néanmoins pas exempt de défauts: il est très lourd, car tous les scripts doivent être lus à chaque appel de page. C'est aussi un langage très "mal rangé", car il s'est construit au petit bonheur la chance, au hasard des versions. Si votre site est en PHP, il sera très simple de trouver de petits bouts de code à ajouter à droite à gauche, pour avoir, par exemple, un fond qui change en fonction du jour de la semaine, de l'heure, ou autre,...

¹ CMS est l'acronyme de content management system, soit, en français, « système de gestion de contenu ». Il s'agit d'un programme informatique utilisant une base de données et permettant de gérer de A et Z l'apparence et le contenu d'un site web.

2. Java

Le java permet de faire des sites et applications web bien plus rapides et sécurisés que le PHP, mais le prix à payer est assez lourd: La consommation en ressources est assez aléatoire, et l'application pas forcément stable. De plus, il n'est pas simple de trouver un hébergeur acceptant autre chose que le PHP. Si vous voulez faire un serveur pour un chat, ou quoi que ce soit qui demande beaucoup d'échanges entre serveur et client sans recharger la page, le Java est une bonne option.

3. ASP .Net

C'est le "langage" serveur de Microsoft. Pourquoi langage est-il entre guillemet? Car la technologie .net permet de choisir son langage! (le plus utilisé restant C#). Avec cette technologie, le code est compilé, et donc beaucoup plus rapide à l'exécution que du PHP! Son inconvénient est qu'il est assez compliqué à faire tourner sur Linux, principal système d'exploitation des serveurs web.

4. SQL

Le langage qui permet d'interroger les bases de données (généralement MySQL). Les bases de données sont indispensables pour tout site web conséquent, elles permettent de stocker et gérer de grandes quantités d'informations.

5. Les autres

ColdFusion (le serveur d'adobe), Ruby (principalement utilisé avec Rails), Python ... Il en existe beaucoup d'autres, largement moins utilisés. Cette liste est donc loin d'être exhaustive!

III. Les langages client

1. HTML

Sans conteste le langage client phare pour les sites web, le HTML gère tout ce qui est mise en forme d'information (bien que le HTML5 tende à faire plus que de l'affichage). Il est composé d'un ensemble de balises, et qui ressemble (voir même qui est, si nous simplifions) au XML (voir plus bas). Il est assez particulier car il n'est en réalité qu'un ensemble de règles énoncées par le W3C². L'interprétation de ces règles est ensuite laissée à la discrétion des créateurs de navigateurs, qui ne se concertent pas, ce qui se traduit par des affichages folkloriques, sur certains navigateurs.

² Le W3C est un sigle utilisé pour définir le World Wide Web Consortium qui est une organisation non lucrative permettant définir des standards pour les technologies liées aux web.

2. CSS

Il s'utilise conjointement avec le HTML, et définit des règles d'affichage. Il est lui aussi maintenu par le W3C, et n'est aussi qu'une suite de recommandations. Sa version 3 permet de faire des choses qui s'approchent du JavaScript (voir plus bas), en permettant de faire des transitions (un moyen de passer d'un état, une couleur, une position, à une autre).

3. JavaScript

Comme pour les deux précédents, le JavaScript est une interprétation de l'ECMAScript³ maintenu par le W3C. C'est le moyen de rendre dynamique des pages, en modifiant les propriétés HTML avec des transitions (pour faire de jolies animations, par exemple), effectuant des requêtes vers le serveur, discutant avec des éléments embarqués tels du flash, et autres.

4. Flash

Le nom réel (et actuel) du langage est l'AS3. C'est l'interprétation de l'ECMAScript d'Adobe. C'est un langage très orienté objet⁴, avec les spécificités de l'ECMAScript, qui sont des listeners et des prototypes. Ses avantages sont qu'il est précompilé, et donc que le code source n'est pas disponible pour le premier passant venu, et qu'il tourne avec le plugin Flash directement fourni par Adobe. Nous n'avons donc pas de mauvaises surprises en fonction des navigateurs.

5. XML & JSON

Ces deux langages sont très éloignés l'un de l'autre dans la forme, mais ont des buts identiques: transmission d'informations et stockage. C'est en quelque sorte la version cliente du SQL.

IV. Les frameworks

1. Définition

Littéralement, cadre ou structure ; dans la programmation en général, on le traduit plus précisément par « cadre d'applications ».

Pour commencer, donc, qu'est-ce qu'un framework ? S'il y avait une réponse simple, on se passerait de ces lignes. La difficulté majeure tient à ce qu'il y a en fait dans la définition deux dimensions principales, complémentaires mais en partie indépendantes : un framework c'est à la fois une boîte à outils et une méthodologie : en quelque sorte, on

³ ECMAScript est un langage de programmation de type script standardisé par Ecma International dans le cadre de la spécification ECMA-262. Il s'agit donc d'un standard, dont les spécifications sont mises en œuvre dans différents langages script, comme JavaScript ou ActionScript. C'est un langage de programmation orienté objet.

⁴ La programmation orientée objet (POO), ou programmation par objet, est un paradigme de programmation informatique élaboré par les norvégiens Ole-Johan Dahl et Kristen Nygaard au début des années 1960 et poursuivi par les travaux d'Alan Kay dans les années 1970. Il consiste en la définition et l'interaction de briques logicielles appelées objets ; un objet représente un concept, une idée ou toute entité du monde physique, comme une voiture, une personne ou encore une page d'un livre. Il possède une structure interne et un comportement, et il sait communiquer avec ses pairs. Il s'agit donc de représenter ces objets et leurs relations ; la communication entre les objets via leurs relations permet de réaliser les fonctionnalités attendues, de résoudre le ou les problèmes.

pourrait dire qu'un framework est à un développeur ce qu'une boîte à outils est à un bricoleur.

Sous plusieurs aspects, un framework peut effectivement fonctionner comme une boîte à outils, en ce sens qu'il propose un certain nombre d'éléments pour faciliter le travail, l'accélérer, voire automatiser tout ou partie des tâches qu'on rencontre lors du développement d'une application web. C'est le côté *work* du framework, si l'on veut.

Enfoncer un clou, à la main, c'est douloureux et très long ; avec une pierre, moins douloureux, mais pas très précis. Avec un marteau, l'outil est assez optimal – même s'il ne dispense toutefois pas de taper. En plus, ce marteau a été testé industriellement, donc a de bonnes garanties de solidité.

Dans le développement web, on ne plante pas des clous, mais il y a des tâches équivalentes qui sont répétitives, peu gratifiantes, et d'autant plus sujettes à erreur qu'elles sont répétitives et peu gratifiantes : construction de bases de données à partir de leur schéma, import de données de test, initialisation d'interfaces d'administration, exécution de tests, gestion du cache, des urls.

Pour tout ça, le framework offre marteaux et tournevis, voire dans certains cas pistolets à clous. Ces « utilitaires », testés et mis à l'épreuve du feu par de nombreux développeurs, ont bénéficié de leurs retours d'expérience, leur efficacité et stabilité sont assurées par ce processus. Mais pas plus qu'un marteau ils ne dispensent pas de taper. Vous pouvez vous procurer une belle boîte à outils, ça ne fixera pas pour autant vos étagères au mur.

A un niveau un peu plus complexe, le framework favorise l'intégration en deux coups de cuiller à pot de plug-ins, qui sont un peu comme les briques évoquées plus haut. Il en existe un grand nombre, pour des fonctionnalités plus ou moins évoluées (authentification, gestion de médias, de contenu...). On dépasse là le cadre de la « boîte à outils », et on est plutôt sur des objets de base qui font gagner du temps. C'est comme acheter des planches déjà découpées plutôt que de les scier soi-même. Mais ces objets conviennent rarement tels quels. Il faut les adapter à ses besoins. Par exemple peindre ses planches de la couleur qui va bien, choisir combien on en pose, à quelle distance les unes des autres...

Point crucial : dans le cas du framework, l'adaptation sera grandement facilitée par la méthodologie qui doit avoir été respectée par les créateurs des plug-ins.

Ce qui nous fait une transition toute trouvée pour le second aspect, non des moindres, de la définition d'un framework : en plus d'être une boîte à outils, un framework, c'est aussi et surtout une façon de travailler.

2. RubyOnRails

Ruby on Rails, également appelé RoR ou Rails est un framework web libre écrit en Ruby. Il suit le motif de conception Modèle-Vue-Contrôleur aussi nommé MVC. En tant que framework, il impose une structure au programmeur, et ainsi l'oblige à avoir une logique et une démarche. Il ajoute aussi un grand niveau d'abstraction dans la programmation de l'application, par un ensemble de fonctions de haut niveau.

Rails est basé sur deux principes fondamentaux :

Ne vous répétez pas

Tenter autant que possible de n'avoir des éléments de l'application qu'à un seul endroit. L'architecture MVC et la méta programmation en Ruby peut faciliter cela.

Convention plutôt que configuration

Ne préciser les éléments de configuration que s'ils ne respectent pas les conventions établies. Rails exploite cela en proposant des comportements par défaut pour la plupart de ses fonctionnalités. Cette vision sera conservée, même après la fusion avec Merb et la future modularisation du framework.

Au démarrage d'un projet Rails, peu d'éléments de configuration sont présents. Il ne contient généralement que l'identifiant et le mot de passe d'accès à la base de données, incontournables en MySQL par exemple. Toutefois on peut commencer un projet sans rien configurer, avec l'utilisation par défaut de SQLite.

Les fichiers de configuration sont de type YAML.

3. Symfony

Symfony est un framework MVC libre écrit en PHP 5. En tant que framework, il facilite et accélère le développement de sites et d'applications Internet et Intranet.

Le site du framework Symfony a été lancé en octobre 2005. À l'origine du projet, on trouve une agence web française, Sensio, qui a développé ce qui s'appelait à l'époque Sensio Framework¹ pour ses propres besoins et a ensuite souhaité en partager le code avec la communauté des développeurs PHP.





Le projet est alors devenu Symfony (car le créateur voulait garder les initiales SF comme "Sensio Framework"), puis Symfony à partir de la version 2.02.

La version 2 de Symfony casse la compatibilité avec la branche 1.x.

4. jQuery

jQuery est une bibliothèque JavaScript libre qui porte sur l'interaction entre JavaScript (comprenant Ajax) et HTML, et a pour but de simplifier des commandes communes de JavaScript. La première version date de janvier 2006.

La bibliothèque contient notamment les fonctionnalités suivantes :

-  Parcours et modification du DOM (y compris le support des sélecteurs CSS 1 à 3 et un support basique de XPath)
-  Événements
-  Effets visuels et animations
-  Manipulations des feuilles de style en cascade (ajout/suppression des classes, d'attributs...)

- ✚ Ajax
- ✚ Plugins
- ✚ Utilitaires (version du navigateur...)

V. Cas particulier : AJAX

Et non, AJAX n'est pas un langage. C'est une manière de construire des applications Web en utilisant JavaScript de façon à ce que le serveur et le client travaillent de concert, sans recharger la page. Cela permet de faire de très puissantes applications web.