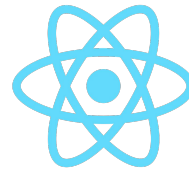


Les frameworks JS



Les frameworks :

Le JavaScript prit un essor avec l'apparition du *Web 2.0* dans lequel JavaScript est devenu de plus en plus utilisé. Combiné au CSS, il devint une alternative à Flash pour le développement de pages dynamiques.

Un framework peut être perçu comme un environnement de travail

On trouvera de la matière première (variables..), des outils (fonctions ..), et surtout des consignes à respecter (formalisme..).

Même si les frameworks peuvent avoir beaucoup d'avantages, ils dictent néanmoins une façon de faire stricte et parfois même très rigide .

Les bibliothèques :

Une bibliothèque peut être considéré comme une boîte à outil

Plutôt que réinventer pourquoi ne pas directement utiliser un outil déjà existant ?

Une bibliothèque est un inventaire d'outils disponibles et vous faites appel à ces outils avec votre code. Vous avez simplement besoin de prendre votre outil et ensuite de vous concentrer sur votre travail !

Vous êtes donc plus efficace et vous y gagnez du temps.



Les principaux frameworks :



Tiré de "The State Of JavaScript 2018" (enquête sur 20 000 développeurs dans 153 pays)



React est une bibliothèque JavaScript open-source développée par Facebook depuis 2013, qui se concentre sur la **gestion de l'interface utilisateur**.

Le but principal de cette bibliothèque est de faciliter la création d'application web monopage (évite le chargement d'une nouvelle page à chaque action demandée donc fluidifie l'expérience utilisateur), via la création de "**composants**".

Les composants peuvent être imbriqués dans d'autres composants pour permettre la création d'applications complexes à partir de blocs de construction simples.

React utilise une syntaxe spéciale appelé JSX, qui permet de mixer l'HTML et le Javascript.



React intègre la notion de **DOM virtuel**. *

Comme les navigateurs sont plutôt lent à réagir face aux changements du DOM, React a l'avantage de limiter les interactions avec ce dernier. Il effectue donc les opérations sur le DOM virtuel et le compare au vrai DOM pour réaliser les changements à effectuer.

Cette bibliothèque est utilisée notamment par Netflix (côté serveur uniquement depuis le 25 octobre 2017 pour gagner 50 % de performance), Yahoo, Airbnb, Sony, ainsi que par les équipes de Facebook, Instagram ou encore WhatsApp.

React Native: équivalent pour la création d'applications natives iOS et Android.
(depuis 2015)



Avantages

:

- très grande communauté (vaste bibliothèque 100% open source, besoin d'aide, conseils en développement)
- produit du code propre (simple à lire)
- flexible et pérenne (composants)
- performance (rapidité du rendu)
- contenu référençable (code généré côté client et serveur)



Inconvénients :

- très peu de documentation officielle
- la prise en main peu peut s'avérer difficile (apprendre et maîtriser les concepts de l'écosystème React)

Ressources:

| | | |
|---|---------------------|-----|
| https://fr.reactjs.org/ | (site web officiel) | fr |
| https://www.w3schools.com/react/default.asp | (tutos) | eng |
| https://codepen.io/search/pens?q=react | (pour s'inspirer!) | eng |
| https://youtu.be/Y7rbJRjaYCY | (tutos vidéo) | fr |



* DOM

Document **O**bject **M**odel signifie « modèle d'objets de document ».

Le DOM est une interface de programmation normalisée par la W3C, qui permet à des scripts d'examiner et de modifier le contenu du navigateur web.

En décomposant :

Modèle : il représente un plan. Le DOM représente le document qui se trouve dans le navigateur internet.



* **DOM**

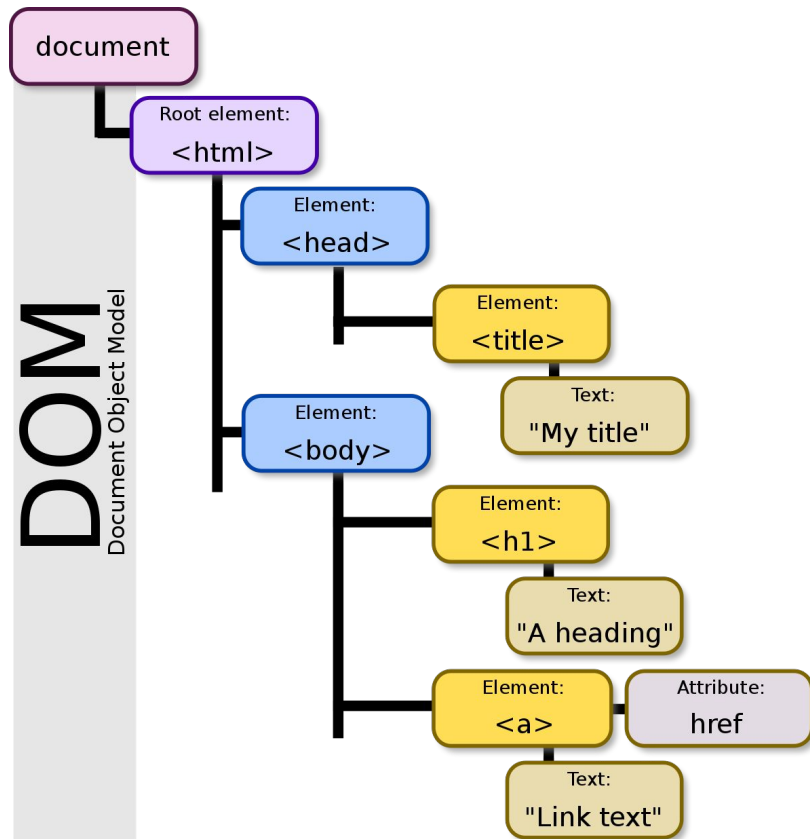
Objet : un objet est un conteneur qui comporte des propriétés et des méthodes

Les objets du DOM peuvent représenter une fenêtre, une phrase, un style...

Document : le DOM concerne un document comme une page web affichée dans un navigateur internet. Le DOM représente le document par une structure en arbre, comportant des nœuds (branches et feuilles).



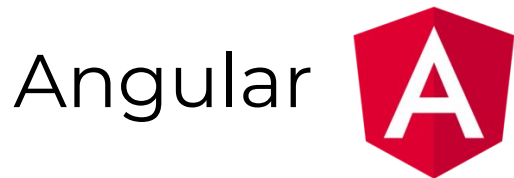
* DOM





AngularJS est un framework JS libre développé par Google.
Ce framework est fondé sur la programmation déclarative qui est utilisée pour construire des interfaces utilisateurs et des composants logiciel.

Une page web conçue avec AngularJS suit l'architecture MVC (modèle-vue-contrôleur). Ce patron a **pour avantage de proposer un couplage faible entre la présentation, les données, et les composants métier**. Dans un langage web, cette séparation permet de **diminuer l'importance des manipulations DOM et d'améliorer la testabilité du code**.



AngularJS utilise l'injection de dépendances, qui permet à votre application web côté client, le support de services traditionnellement apportés côté serveur.

Par ailleurs, AngularJS a la particularité d'utiliser des promesses, pour exécuter une opération asynchrone (qui ne bloque pas l'exécution du reste du programme).

La documentation officiel de AngularJS (anglais)

<https://docs.angularjs.org/tutorial>

Documentation et bootcamps:

<https://www.freecodecamp.org/> (anglais)

<https://www.grafikart.fr/> (français)



Vue a été créée par Evan You après avoir travaillé pour Google en utilisant Angular.js. Son processus de réflexion : "Et si je pouvais juste **extraire la partie que j'aime vraiment dans Angular et construire quelque chose de vraiment léger**".

Vue.js est un framework javascript open-source destiné au développement d'interface utilisateur, qui se dit comme "polyvalent" : c'est à dire à mi-chemin entre une bibliothèque javascript et un framework.

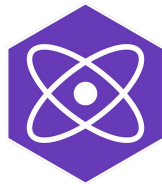


Vue présente une architecture qui se concentre sur la composition des composants. **Les fonctionnalités avancées** requises pour les applications complexes (routage, gestion d'état ..) **sont offertes par le biais de bibliothèques et de paquets**. C'est un framework javascript qui s'adapte, facile à intégrer et n'imposant aucunes contraintes sur la gestion du routeur, dépendances etc..

Avantages :

- S'intègre très facilement au framework PHP Symfony
- Performant car se concentrant sur la couche de présentation et déléguant le routage et la gestion d'état à des bibliothèques connexes
- Plus simple à prendre en main pour un débutant

Preact

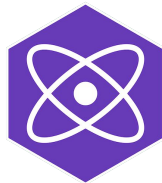


La bibliothèque Preact a été publiée dans sa version 1.3 en septembre 2015 sur github par Jason Miller, développeur travaillant pour Google Chromes Labs.

Preact est un framework front-end, c'est une alternative à React, souvent considéré comme une version miniature de ce dernier.

Il se distingue de son paronyme de plusieurs façons, notamment en **se focalisant sur la manipulation du DOM**. Il n'a pas pour but de remplacer React mais se base sur son ADN tout en faisant des compromis pour optimiser son poids.

Preact



La ou Preact se démarque, c'est dans sa philosophie : il se restreint uniquement à la manipulation du DOM, et permet entre autres de créer un DOM virtuel et d'utiliser une multitude de composants UX.

Avantages

- Très léger
- Permet une architecture épurée
- Optimise les performances d'un site web
- Code plus simple, moins "fouilli"

Inconvénients

- Quelques changements à noter au niveau du comportement du navigateur (par exemple : onChange)
- Preact-compat ajoute 2Kb à la taille du bundle du final en cas de migration de React vers Preact.

Ember



Ember est un framework JavaScript open source qui permet de **créer des applications web dynamiques**, en se chargeant de modifier son contenu, contrairement aux sites traditionnelles qui se contentent de passer d'une page à l'autre.

Il s'appuie sur l'architecture MRC Modèle Route Composant et utilise en son sein le moteur de template Handlebars qui se charge de mettre à jour en temps réel les données lorsqu'elles sont modifiées.

Ember



Créé par Yehuda Katz, Ember a des objectifs propre :

Proposer des solutions pour les app web ambitieuses.

Plus productif en étant une bibliothèque d'outils travaillant ensemble pour proposer un développement complet.

Une stabilité sans fluctuation

Ember consiste en 5 points importants qui sont: Routes - Models - Templates - Components - Services

Ember



Avantages

- Une séparation simplifiée logique (Javascript), et des vues (HTML CSS)
- Courbe d'apprentissage moyenne
- Documentation complète et intuitive

Inconvénients

- Petite communauté d'utilisateurs
- Lourd
- Beaucoup de conventions

Les frameworks PHP





CakePHP est un framework PHP, relativement simple à apprendre et à prendre en main (par rapport à Symfony par exemple).

D'après le blog « [jesuisunpapageek](#) » : « en seulement 4 jours, vous pouvez être capable de réaliser une application basique du style blog ou livre d'or.

Il **simplifie grandement le développement d'applications web** et offre un ORM* flexible et un système de génération rapide.

* object-relational-mapping : type de programme informatique qui se place en interface entre un programme applicatif et une base de donnée relationnelle pour simuler une base de données orientée objet.



D'après le site officiel, CakePHP prime par la rapidité qu'il offre, par le fait qu'**il n'est quasiment pas besoin d'être configuré**, par sa licence MIT, par les fonctionnalités incluses, ses sets de conventions et surtout par sa sécurité.

S'il y a un défaut à prendre en compte sur ce framework, c'est qu'il est **très peu demandé et utilisé sur le marché du travail**.



Symfony est un framework MVC open-source, écrit en PHP et 100% français. Il est un puissant framework PHP utilisé pour développer des applications web ou des sites web complexes, voire haut de gamme. Relativement simple d'accès, Symfony est un ensemble de composants (ou « bibliothèques ») qui facilitent le développement web en réduisant de façon considérable le temps et l'effort requis pour créer des composants génériques.

Avantages :

- Plus de flexibilité.
- Facilité d'utilisation et grandes performances.
- De l'extensibilité.
- Facilité de débogage.
- Facilité des tests.



Inconvénients:

- Nécessite dévouement et sacrifice pour maîtriser les compétences: Pour réussir en tant que développeur Symfony, vous devrez être dévoué et sacrifier plus de temps pour obtenir tous les concepts derrière ce framework. Comme il s'agit de l'un des framework PHP les plus stables et les plus fiables, il est livré avec des compétences assez difficiles à apprendre qui nécessitent du dévouement et plus d'efforts.