

Abstract

Ce document companion a été produit lors de l'évaluation de plusieurs frameworks disponibles pour une application web.

Frameworks

Liste non-exhaustive et partiellement subjective basée sur les données exposées ci-dessous.

Langage	Framework	C	M	Pe	Po	T	Total
JavaScript	Angular	M	M	B	B	M	3.5
	MeteorJS	B	B	M	N	B	3.5
	NodeJS	M	M	M	B	N	3.5
	React	B	B	M	B	B	4.5
Python	Django	M	M	B	B	M	3.5
	Flask	B	B	B	B	B	5
	Pyramid	B	B	B	N	B	4
Ruby	Rails	N	M	B	B	B	3

Critères - [B]on, [M]oyen, [N]égatif

- C** Complexité - Difficulté d'utilisation / apprentissage
- M** Maintenabilité - Aisance et possibilités de maintenance
- Pe** Performance - Capacités du framework en termes de vitesse
- Po** Popularité - Popularité du langage à travers le monde
- T** Taille - Espace de stockage requis

C - Complexité

Évaluer la complexité est hautement subjective en fonction des préférences et habitudes du développeur. Nous pouvons néanmoins comparer les langages utilisés et la structure du framework. Une mesure de la complexité se base sur l'imbrication des blocs de code. Un code comportant beaucoup de structures imbriquées sera moins lisible, et donc plus complexe. Les résultats¹ en figure 1 montrent que cette mesure favorise le langage Python.

La figure 2 représente un comparatif² effectué par Wolfram©(plateforme mathématique bien connue) mesurant la quantité de lignes de code nécessaires dans un langage pour reproduire un programme donné dans un autre langage. Le nombre de lignes de code n'est toutefois que partiellement une mesure de la complexité car toute ligne de code ne s'équivaut pas en terme de complexité.

1. Étude de Seerene©, 2015

2. <http://blog.wolfram.com/2012/11/14/code-length-measured-in-14-languages/>

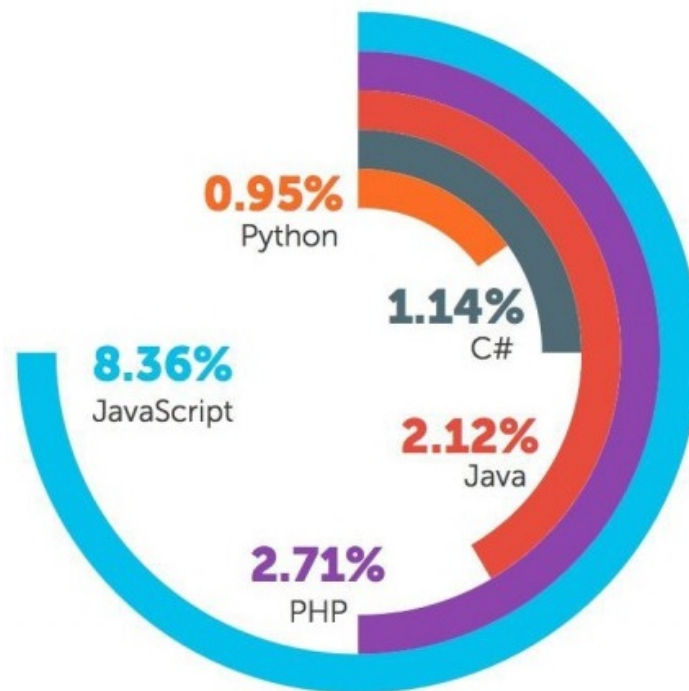


FIGURE 1 – Pourcentage de code imbriqué

Large tasks – Line count ratio

Larger numbers indicate that the language on the top needs longer code	C	C++	Fortran	Java	Common Lisp	Python	C sharp	JavaScript	R	MATLAB	Clojure	Pascal	Haskell	Ruby
Mathematica	17.	9.1	8.1	6.4	6.3	7.2	6.4	5.	3.2	3.2	1.6	5.8	3.5	5.2
Ruby	2.7	1.8	1.9	1.3	1.1	1.1	1.5	0.96	0.72	0.94	0.39	1.4	0.7	
Haskell	3.6	2.7	2.5	2.	1.6	1.7	2.2	1.5	1.1	1.5	0.67	2.1		
Pascal	2.2	1.5	1.2	0.83	0.77	0.8	1.	0.79	0.46	0.61	0.2			
Clojure	8.8	5.3	5.2	3.6	3.7	3.3	3.8	2.5	1.9	2.9				
MATLAB	3.6	2.4	1.8	1.1	1.4	1.1	1.7	0.94	0.75					
R	4.7	3.3	2.4	1.9	1.8	1.7	2.1	1.5						
JavaScript	2.8	2.1	1.9	1.2	1.2	1.1	1.6							
C sharp	2.	1.4	1.3	0.91	0.8	0.82								
Python	2.2	1.6	1.5	1.1	0.88									
Common Lisp	2.8	1.8	1.6	1.3										
Java	2.1	1.4	1.5											
Fortran	1.4	1.												
C++	1.4													

FIGURE 2 – Nombre de lignes de code nécessaires pour traduire un programme

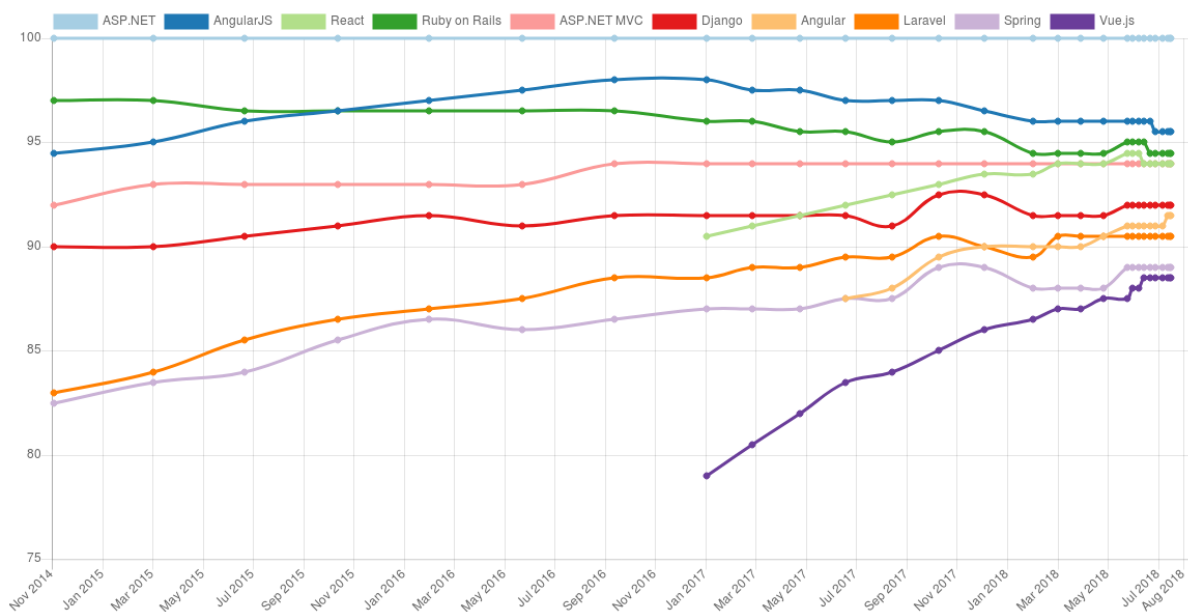


FIGURE 3 – Popularité des frameworks de novembre 2014 à aujourd’hui

M - Maintenabilité

L’aisance avec laquelle un code est maintenu à jour est inversement lié à la complexité. A cela, nous pouvons ajouter les fréquences de mise à jour des frameworks. La structure du framework impacte également la maintenabilité ; moins on modifie de fichiers pour implémenter une fonctionnalité, mieux c’est.

Pe - Performance

Bien que moins importantes que dans une application critique (e.g. : aérospace), les performances sont mesurées en termes de temps de réponse.

Po - Popularité

La popularité est sans doute la plus importante des mesures de ce comparatif. Un langage populaire dispose de plus de tutoriels et aide en ligne que d’obscurs langages / frameworks peu ou pas utilisés.

La figure 3 montre une évolution³ de la popularité des frameworks les plus courants.

T - Taille

Un framework léger sera plus intéressant pour ce projet. D’une part pour la complexité, car moins il est nécessaire de modifier/ajouter des fichiers, plus l’utilisation est aisée. D’autre part car le serveur de destination nous est inconnu et nous ne savons donc pas quel espace de stockage sera mis à disposition.

3. <https://hotframeworks.com/>

Critères spéciaux

Notons qu'en lien avec un autre projet, il serait utile d'employer le système de géo-représentation GeoNode. **GeoNode est orienté Django** et ne peut pas être utilisé avec d'autres frameworks.

D'autres critères spéciaux pourront se rajouter avec la réunion du 03 septembre 2018.

Autres

La taille de l'application sera peu dépendante du framework utilisé en back-end. La majeure partie du poids du site (hormis les données) sera attribuée au front-end et donc dépendante du système front-end utilisé

Conclusion

Le choix d'un framework repose au final sur de nombreux critères dont plusieurs sont subjectifs. L'expérience est bien souvent facteur décisif.

Il existe de très nombreux frameworks et la liste ci-dessus ne saurait être exhaustive. Les frameworks lourds, peu performants ou inadaptés à la construction de l'application décrite n'ont pas été explorés (Java, PHP, ...).

Au final, le choix se limitera surtout entre les frameworks JavaScript et Python en raison de leur très grande popularité dans les applications web pour petites entreprises. Les framework Python se démarquent positivement et si Flask semble être le meilleur choix pour notre projet, Django n'en reste pas moins un second choix de qualité qui pourrait être choisi en raison du module **GeoNode**.