GANATAN

Démarrer une Application Web avec React 19

Guide Complet

Mise à jour du 4 mars 2025

Nous allons créer une application Web en utilisant

React version **19.0.0**

Nous suivrons les conseils donnés sur le site **reactjs.org**

- React a été créé par **Facebook**.
- React est une **librairie Javascript Frontend**.
- React utilise **Javascript**.

www.ganatan.com

Comment le faire ?

Pour débuter notre projet voici un résumé de ce que nous allons faire.

- **React est une Librairie Javascript : c'est quoi ça**
  Qu'est ce que javascript et pourquoi l'utiliser ?

- **Dans la catégorie Framework les gagnants sont**
  Librairie ou Framework lequel choisir

- **Frontend vs backend**
  Ou se situe React

- **Angular vs React vs Vuejs**
  Quel Framework javascript choisir

- **Installation des outils nécessaires**
  Un descriptif rapide des pré requis pour commencer à travailler.
  Node.js, Visual studio Code et Git

- **Initialisation du projet**
  Nous utiliserons create-react pour mettre en place notre projet,
  en utilisant les best practices (les meilleures pratiques) préconisées par Facebook.

- **Mise à jour du projet**
  Vérifier les dépendances utilisées et les mettre à jour.

- **Code source**
  Le code complet du projet est disponible sur Github.

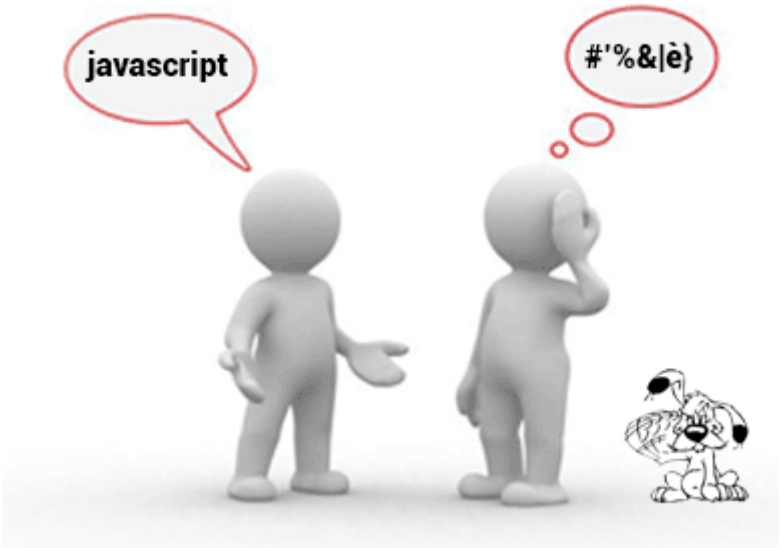
React est une Librairie Javascript : c'est quoi ça ?

Commençons par le mot **Javascript**.

Si vous voulez communiquer avec un être humain le plus simple est de **parler la même langue**.
Si vous voulez communiquer avec un ordinateur le plus simple est de **parler le même langage informatique.**

Langue ou un **langage informatique** c'est la même chose, c'est pour **communiquer**.

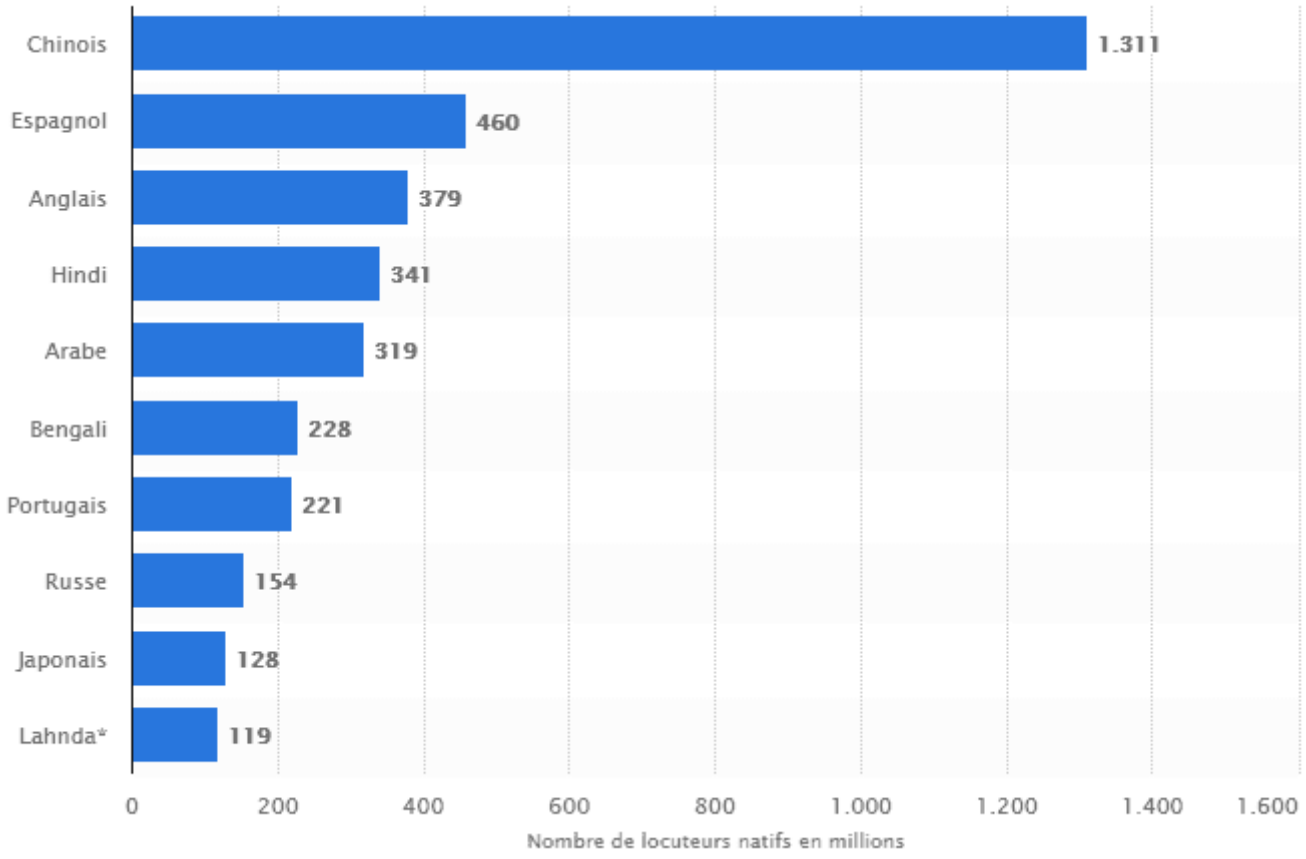
Et ce qui nous intéresse ici
Javascript est un **langage informatique (**ou **computer language** en anglais**).**



Peu importe la langue me direz-vous et je suis d'accord avec vous elles se valent toutes.
Mais si vous voulez communiquer avec le plus grand nombre de personnes possible, vaudra mieux parler le chinois ou l'anglais plutôt que le péruvien ou le finlandais.

Y a qu'à voir la liste ci-dessous.

Classement des langues les plus parlées dans le monde



Maintenant si votre dada c'est la programmation et que vous voulez créer **un site web ou une application web,** il va falloir demander de l'aide à un ordinateur, pas le choix.

Et donc parler le même langage que lui.

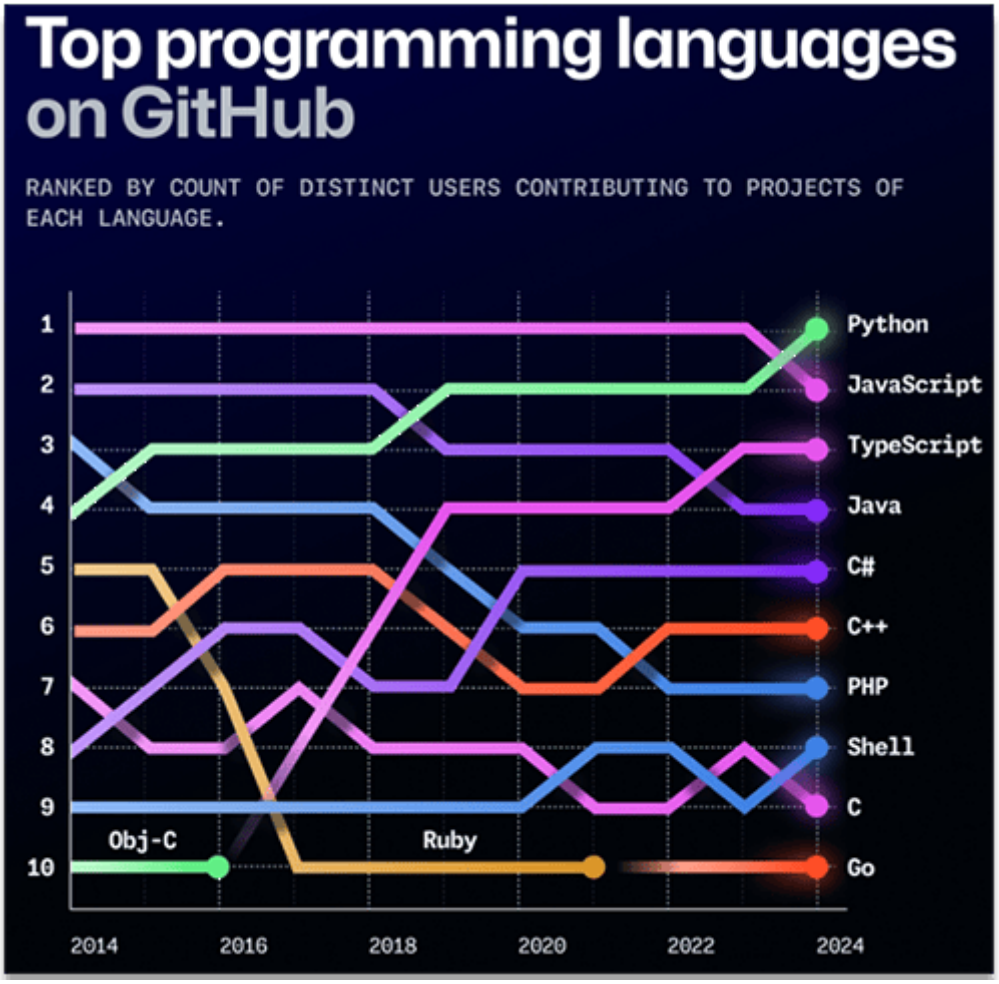
Et là aussi certains langages peuvent s'avérer plus utiles que d'autres.

Des langages malheureusement, Il en existe toute une tripotée.

La question que l'on est en droit de se poser : **Quel langage informatique choisir ?**

Voyons voir par exemple l'avis de Github
Le service d'hébergement et de gestion de logiciels.

Principaux langages de programmation



Bon si vous voulez trouver du travail dans la programmation vous savez ce qu'il vous reste à faire.

En tout cas **React utilise Javascript**.

Si vous choisissez React c'est plutôt **rassurant pour l'avenir.**

Sur le site officiel de React on peut lire que react est une **librairie Javascript**.

L'adresse est ici https://reactjs.org/

React est considérée comme une librairie car il fournit de nombreux outils de développement.

Mais ne fournit pas tous les outils nécessaires comme le feraient Angular ou React.

Par exemple pas d'outils permettant le SSR ou server side rendering , il faudra dans ce cas faire appel à d'autres outils.

Néanmoins on peut considérer que React fournit une très large palettes de fonctionnalités qui suffisent à la plupart des applications.

Une librairie certes mais une sacrée librairie que l'on pourrait aisément qualifier de Framework.

Donc pour nous simplifier la vie dans ce tutoriel lorsque je parlerai de React je parlerai de **Framework**.

Ceci étant dit, intéressons-nous maintenant au mot **Framework**.

Dans la catégorie Framework les gagnants sont :

Framework pourrait se traduire littéralement par cadre de travail.
Moi je trouve que **Boite à outils** est une traduction bien plus parlante.

Pas la peine de réinventer la roue à chaque fois, si on peut trouver de l'aide quelque part.
Un Framework sait faire beaucoup de choses et est prêt à **nous aider** alors on en profite**.**
Avec un Framework il suffit de **choisir les outils** qu'il nous propose et qui correspondent le mieux au **travail à effectuer**.

Si vous voulez produire ce qui se fait de mieux sur le marché faut les bons outils.

Tu connais le business, et je connais la chimie

Nous avons parlé plus haut des langages les plus utilisés **Javascript, Python, Java et PHP**.

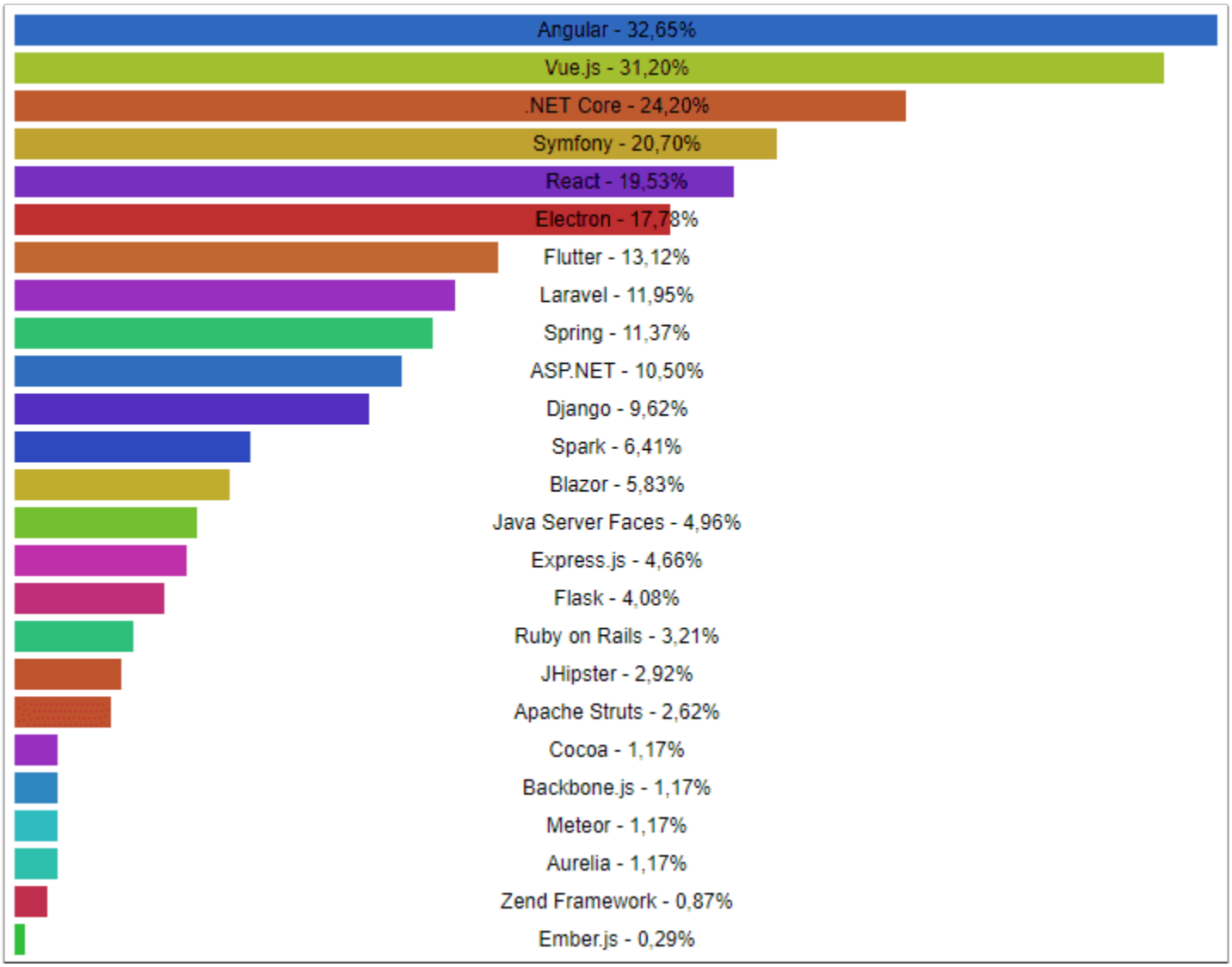
Chacun de ces langages disposent de Frameworks et pas qu'un seul d'ailleurs.

Le site **developpez.com** nous offre une petite liste intéressante, si vous êtes curieux c'est ici

Le résumé est là.

L'article date un peu mais il permet d'avoir une synthèse Front et Back.

Quels sont les frameworks que vous aimeriez apprendre en 2019 ?



Ce n'est pas la peine de chercher plus loin les Frameworks à utiliser sont donc les suivants.

- Le Framework **Spring** pour **Java**
- le Framework **Django** pour **Python**
- Les Frameworks **Laravel ou Symfony** pour **PHP**

Quant à Javascript y'a pas photo les gagnants sont

- **Angular**
- **React**
- **Vuejs**

Les meilleurs Frameworks ?

Donc React est un Framework Javascript.

- **Javascript** est classé n° 1 parmi les langages informatiques.
- **React** est parmi les 3 meilleurs **frameworks javascript**.

Frontend vs Backend

Avant d'aller plus loin parlons du web en 2023.

Depuis quelques années une tendance forte se dessine dans le développement.

La conception d'une application Web se divise en deux parties

- la partie **front-end** :

  Il s'agit de la partie visible de l'iceberg, celle que l'utilisateur voit dans son navigateur.

  Elle utilise en général les langages **CSS, HTML et Javascript.**

  Elle concerne le design du site web, et constitue ce que l'on appelle l'**UX** (expérience utilisateur) et l'**UI** (interface utilisateur).

- la partie **back-end** ;

  Elle correspond à la partie non visible.

  Elle est souvent opposée à la partie Frontend parce qu'elle est plus compliquée et demande plus de travail (**ouais! ça c'est à voir**)

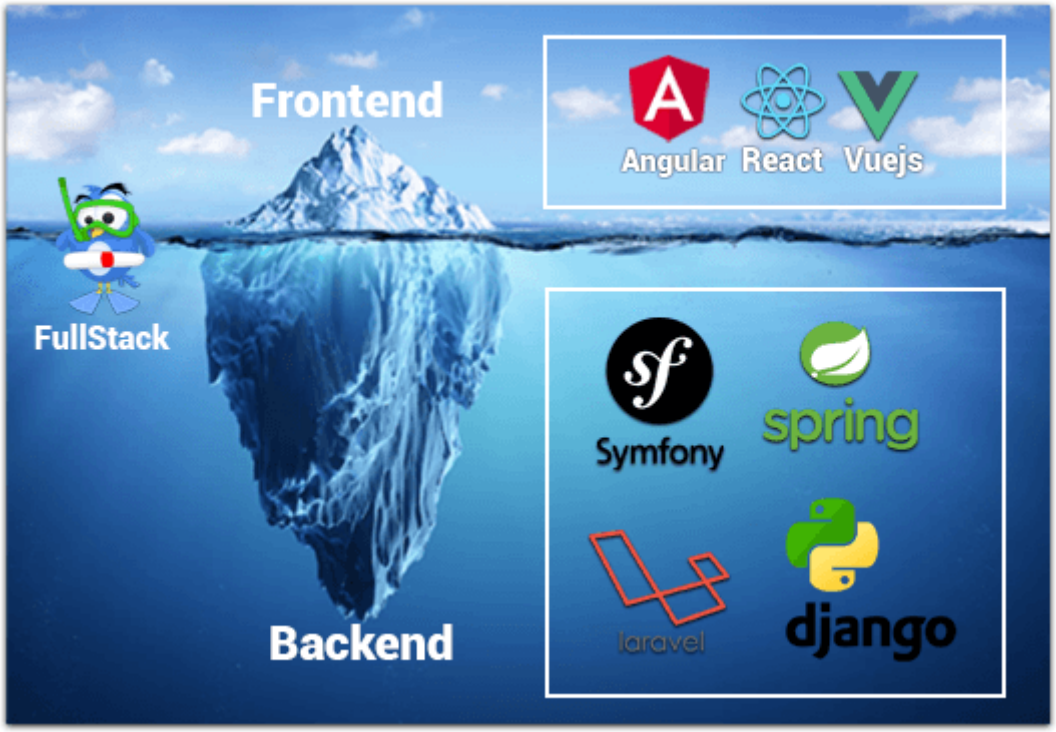
  Elle concerne des éléments comme le serveur, la base de données, l'administration ou la sécurité.

Il est de plus en plus admis que les deux camps ont leurs partisans.

- **Javascript** pour le **Front-end**
- **Java, Python** et **PHP** pour le **Back-end**

Ce qui pourrait donner une représentation comme celle-ci

Frontend vs Backend



Pour terminer prenons l'exemple classique mais simple d'une boutique.
Il va nous permettre de comprendre plus facilement le concept.

Imaginons un **Vidéo club**.

Avec **Netflix** les plus jeunes d'entre vous ne savent même plus ce que c'est.
Un vidéo club c'était une bonne vieille boutique ou on allait louer nos **DVD** voire même nos **K7 vidéo** pour les plus anciens.

Concurrencé par la VOD il n'en existe presque plus malheureusement. Grande tristesse.

En tant que client
• On entrait dans la boutique pour choisir nos DVD et être conseillé par notre vendeur préféré, c'est le **front-end.**

• On n'allait jamais dans l'arrière boutique où sont gérés le stockage et l'approvisionnement des DVD, ça c'est le **back-end**.

Facile à comprendre.
• Le **front-end** : comme dit le président c'est de la poudre de perlimpinpin et on en met plein les yeux.
• Le **back-end** : les travailleurs de l'ombre dont on ne connait presque pas le

nom (salut quentin !).

Ah ! Les années 80

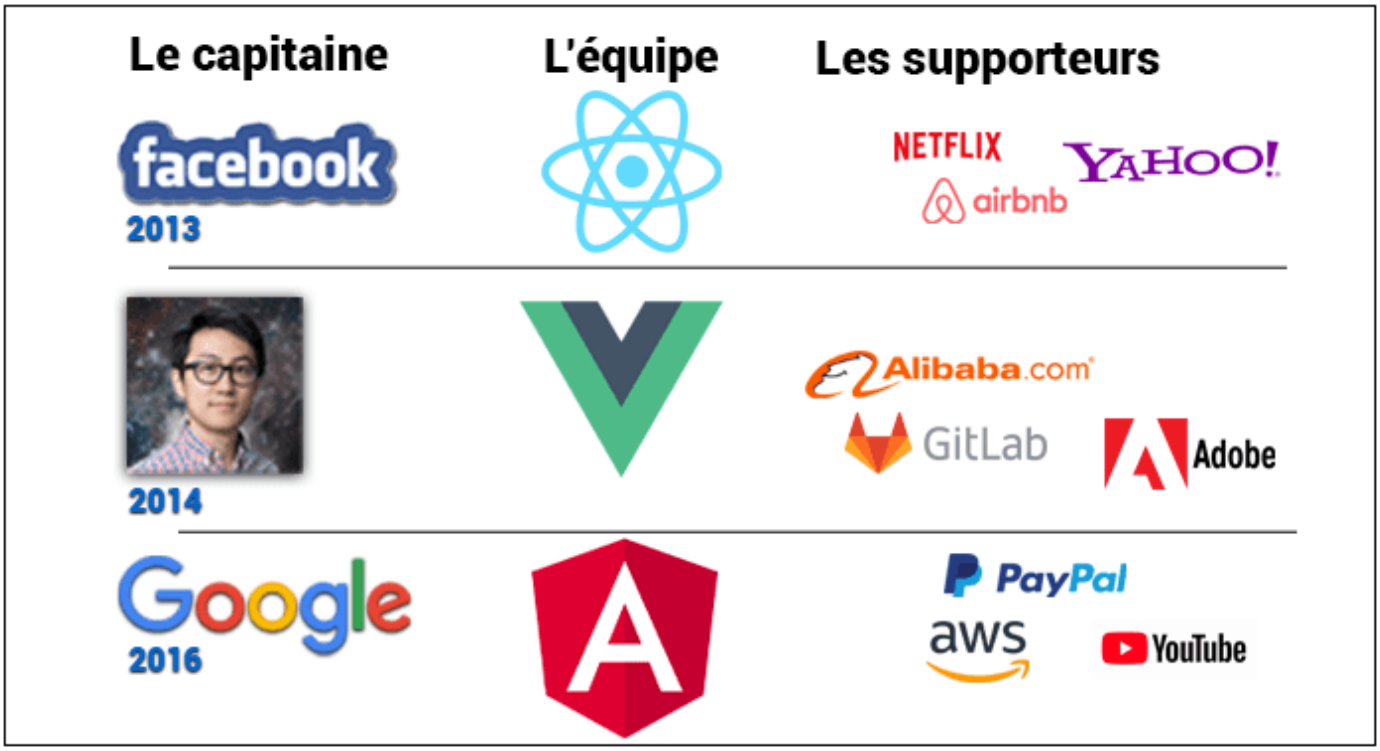
Angular vs React vs Vuejs

Frontend vs Backend c'est bien beau tout ça mais avec quoi on le fait.

Depuis quelques années, **3 Frameworks** Javascript dominent sans partage le **développement Frontend**.

- **React** (Créé par **Facebook** le 29 mai **2013**)
- **Vue.js** (Créé par **Evan You** le 11 Février **2014**)
- **Angular** (Créé par **Google** le 14 Septembre **2016**)

Angular vs React vs Vuejs : Choisissez votre camp



React est un framework javascript open-source basé sur Javascript.
Il a été développé par l'équipe de Facebook.

Dans ce tutoriel nous utiliserons donc React pour créer une application Web.

Passons donc à la pratique.

Un peu d'histoire

Les versions successives de React ont été les suivantes :

- **React** est sortie le 29 mai 2013
- React 0.3.0 est sortie le 2 juillet 2013
- React 0.13.0 est sortie le 19 avril 2015
- React 0.14.0 est sortie le 9 octobre 2015
- React 15.0.0 est sortie le 8 avril 2016
- React 16.0.0 est sortie le 26 septembre 2017
- React 17.0.1 est sortie le 14 Octobre 2020
- **React 17.0.2** est sortie le **22 Mars 2021**

Conditions préalables

Pour pouvoir travailler avec React quelques outils sont requis.

Les outils nécessaires pour réaliser ce didacticiel sont
- **React** version 18.2.0
- **Visual studio code** version 1.81.1
- **node.js** version 18.17.1 LTS (Long Term Support)

Les dernières versions de ces outils sont disponibles ci-dessous
- https://github.com/facebook/react/releases
- https://code.visualstudio.com/updates/v1_81
- https://nodejs.org/fr/download/

Initialisation du projet

La documentation React est accessible sur le site officiel

Nous essaierons le plus souvent possible de respecter les *best practices* préconisées par l'équipe de Facebook.

La création de projet est détaillée à l'adresse suivante

https://react.dev/learn/start-a-new-react-project#create-react-app

Nous utiliserons un package runner **npx**

```
# Générer un projet appelé react-starter
npx create-react-app react-starter

# Se positionner dans le projet
cd react-starter

# Exécuter
npm start
```

La commande npm start éxécute le script **react-scripts** contenu dans le fichier **package.json.**

Cette commande éxécute le projet sur un port par défaut (**3000**)

Le script start lance automatiquement l'application dans votre navigateur par défaut.
Mais ci dessous la commande si vous voulez l'éxécuter manuellement.

```
# Tester
http://localhost:3000/
```

Visual studio code

Dans la suite du tutoriel nous utiliserons **Visual Studio Code**.
VS code est un éditeur de code développé par Microsoft pour Windows, Linux et OS X.

Après avoir lancer VS code ouvrez un dossier dans le répertoire **react-starter.**

Ouvrez ensuite le fichier **package.json**.
Celui-ci contient un certain nombre de commandes (ou scripts) que nous utiliserons tout au long de ce tutoriel.

Ouvrez une console VS code (sélectionner Afficher/Terminal) pour éxécuter les scripts suivants

- **npm run start** : Execute l'application en mode développement
- **npm run build** : Compile l'application dans le répertoire build
- **npm run test** : Execute les tests unitaires

En mode développement si nous voulons personnaliser le port il suffit de modifier le script start dans le fichier package.json.
Par exemple pour utiliser le port 3001 le script serait le suivant dans le fichier package.json
"start": "cross-env PORT=3001 react-scripts start",

En prenant soin d'installer au préalable la librairie cross-env si nécessaire.
npm install -g cross-env

package.json

```
"scripts": {
  "start": "react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test",
  "eject": "react-scripts eject"
},
```

Mise à jour

Le fichier package.json contient les différentes dépendances de votre projet.

Concernant les dépendances et leur version la documentation npm est la suivante

https://docs.npmjs.com/files/package.json#dependencies

Les spécificateurs de version sont nombreux.

Nous pouvons utiliser par exemple

- **version** Doit correspondre à la version exactement
- **~version** "Approximativement équivalente à la version"
- **^version** “Compatible avec la version”

Nous obterons quant à nous pour le premier spécificateur "**version**", qui est le plus simple, le plus explicite mais aussi le plus restrictif.

Nous allons mettre à jour le fichier package json avec les dernières dépendances

- Pour contrôler les dépendances à mettre à jour lancer la commande

npm outdated

- A ce jour toutes les dépendances peuvent être mises à jour

- Modifier le fichier package.json comme suit puis executer le script

npm install

package.json

```
"dependencies": {
  "@testing-library/dom": "10.4.0",
  "@testing-library/jest-dom": "6.6.3",
  "@testing-library/react": "16.2.0",
  "@testing-library/user-event": "14.6.1",
  "react": "19.0.0",
  "react-dom": "19.0.0",
  "react-scripts": "5.0.1",
  "web-vitals": "4.2.4"
},
```

Commande eject

Le fichier package.json contient un script particulier.

- *npm run eject*

Cette commande éxécute le script **react-scripts eject**

Elle est destinée aux développeurs expérimentés.

Elle va générer automatiquement un certain nombre de fichiers.

Ces fichiers permettront de personnaliser certains aspects du développement react.

Les fichiers créés seront

- **script/build.js**
- **script/start.js**
- **script/test.js**

Enfin le fichier **package.json** sera aussi modifié profondément.

Cette commande est à utiliser avec précaution, un retour à l'état précédent n'étant pas possible.

Modifications

Nous allons procéder à quelques modifications histoire d'améliorer l'application.

La premère modification concerne le linter.

Rajoutons les dépendances nécessaires avec les commandes suivantes.

```
npm install -D eslint
npm install -D eslint-plugin-import
npm install -D eslint-plugin-jsx-a11y
npm install -D eslint-plugin-react
```

Les dépendances sont automatiquement rajoutées dans package.json.

Autant les mettre à jour.
On vérifie les dépendances avec la commande
npm outdated

Il ne reste qu'à modifier les versions des dépendances dans le fichier.
Je vous donne le résultat final.

```
"devDependencies": {
  "eslint": "8.29.0",
  "eslint-plugin-import": "2.26.0",
  "eslint-plugin-jsx-a11y": "6.6.1",
  "eslint-plugin-react": "7.31.11"
}
```

On crée un fichier qui contiendra les règles du linter.
Le fichier .eslintrc.json

.eslintrc.json

```
{
  "env": {
    "browser": true,
    "es2021": true,
    "jest": true
  },
  "extends": [
    "eslint:recommended",
    "plugin:react/recommended"
  ],
  "parserOptions": {
    "ecmaFeatures": {
      "jsx": true
    },
    "ecmaVersion": "latest",
    "sourceType": "module"
  },
  "plugins": [
    "react"
  ],
  "rules": {
    "react/react-in-jsx-scope": "off",
    "linebreak-style": 0,
    "no-undefined": "error",
    "no-var": "error",
    "prefer-const": "error",
    "func-names": "error",
    "id-length": "error",
    "newline-before-return": "error",
    "space-before-blocks": "error",
    "no-alert": "error",
    "react/prop-types": 0,
    "indent": [
      "error",
      2
    ]
  },
  "settings": {
    "react": {
      "version": "detect"
    }
  }
}
```

Il faut pouvoir éxécuter le linter.

Pour cela rajoutons 1 script dans le fichier package.json

npm run lint

Ce qui donne

```
"scripts": {
  "start": "react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test",
  "eject": "react-scripts eject",
  "lint": "eslint \"src/**/*.{js,jsx}\""
},
```

Seconde modification **personnaliser** les informations affichées dans le navigateur.

On modifie les informations dans le répertoire **public**

- Modifier le titre dans le fichier i**ndex.html**
- Remplacer le **favicon.ico**

Conclusion

Il ne reste plus qu'à effectuer le debuggage, les tests et le passage en production.

Commencons par le debugage.

Toutes les modifications entrainent une recompilation du code.

```
# Exécuter
npm run start

# Tester
http://localhost:3000/
```

Par exemple Modifier le fichier app.js
Compilation Tests : Edit src/App.js and save to reload.

La compilation est alors éxécutée automatiquement et le navigateur se réactualise.

Les tests à effectuer sont les suivants.

```
# Tester
npm run test
```

Enfin la compilation et la mise en production.

```
# Production
npm run build

# Tester
http-server -p 8080 -c-1 build
http://localhost:8080/
```

Code source

Le code source obtenu à **la fin de ce tutoriel** est disponible sur github
https://github.com/ganatan/angular-react-starter

Les étapes suivantes vous permettront **d'obtenir une application prototype.**

- Etape 2 : Routing avec React