React JS

**Introduction :**

ES6 est une étape majeure dans l'évolution de JavaScript.

Après l'essor d'ES6, de nombreuses bibliothèques et frameworks sont apparus. React.js est considéré comme l'un des plus populaires.

C'est la raison pour laquelle nous avons choisi de nous concentrer sur l'apprentissage de React dans cette super compétence.

Au cours de cette super compétence, nous allons :

1. **Explorez React.js et comprenez les raisons de sa popularité.**
2. **Apprenez à configurer un environnement afin que nous puissions commencer à utiliser React.js dès que possible.**
3. **Découvrez comment démarrer avec React.js et faire nos premiers pas.**

**What is React?**

Comme nous l'avons dit précédemment, React.js **est une bibliothèque JavaScript permettant de créer des interfaces utilisateur.**

Eh bien, vous vous demandez peut-être en ce moment : qu'est-ce qu'une bibliothèque ? comment peut-on le définir ?

Une bibliothèque peut être définie comme un ensemble de fonctions et de classes regroupées dans un package.

L'idée principale derrière les bibliothèques **est de créer un code réutilisable que n'importe quel membre de la communauté de développement peut utiliser et auquel il peut contribuer**.

Si vous recherchez "qu'est-ce que React.js?" sur google, vous trouverez des centaines d'articles informatifs et complets sur React.js. Mais nous sommes ici pour garder les choses courtes et douces.

**React.js est une bibliothèque JavaScript open source utilisée pour créer des interfaces utilisateur spécifiquement pour les applications à page unique.**

Il s'agit de gérer la couche de vue pour les applications Web et mobiles.

Les applications modernes ont tendance à adhérer à ce que l'on appelle un modèle d'application à page unique (SPA).

Dans notre navigation quotidienne, nous ne chargeons jamais de pages entièrement nouvelles et différentes, même si nous rechargeons la page. Au lieu de cela, les différentes vues de votre application sont chargées et déchargées dynamiquement, écrites et réécrites dans la même page.

Le composant App.js sera le point de départ de notre SPA. App.js est le composant principal.

**Why use React?**

Souvent, dans les applications du monde réel, développer un site Web en utilisant HTML, CSS et JavaScript peut être très difficile et monotone. Voici pourquoi:

Vous devez créer manuellement des fichiers HTML, CSS et JavaScript pour chaque page de site Web.

Si vous souhaitez déplacer un élément qui existe dans la page A vers une autre page B, vous devrez copier tout son HTML, CSS et JavaScript pour le faire fonctionner et cela est non seulement fastidieux mais aussi chronophage.

Le partage de CSS et de JavaScript peut rendre la maintenance très difficile à mesure que le site Web se développe.

Afin de résoudre les problèmes évoqués précédemment, des bibliothèques et des frameworks ont heureusement vu le jour. Ils rendent le développement Web beaucoup plus facile et plus pratique pour les développeurs. L'une des principales solutions est React.js.

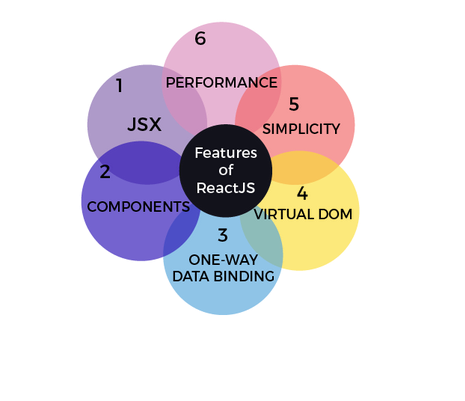
Pourquoi choisir d'apprendre React.js ?

La réponse à cette question est très simple, React est :

1. Facile à apprendre.
2. Hautement maintenable.
3. Super-rapide dans les vues de rendu.
4. Basé sur les composants, avec la possibilité de diviser le code autant que nous le voulons
5. Simple dans le processus de débogage.

**Caractéristiques de React (Features of React):**

React possède plusieurs excellentes fonctionnalités qui sont très utiles lors de la création de l'interface utilisateur.



1. **JSX** : React utilise la syntaxe JSX dans l'écriture des composants, ce qui les rend plus indépendants
2. **Component:** React adopte l'approche d'application basée sur les composants qui nous permet d'utiliser le même composant à plusieurs reprises.
3. **One way data flow:** React nous permet uniquement de transmettre des données de parent à enfant, ce qui est très utile pour tracer les données lors du débogage.
4. **Virtual DOM:** React utilise le DOM virtuel qui rend le rendu de l'interface utilisateur super rapide.
5. **Simplicity: React** est très simple à apprendre et à utiliser, en particulier pour les nouveaux arrivants.

# Apps built with react

Le monde numérique est en constante évolution et il change en ce moment même. Dans un monde comme celui-là, il est décidément difficile de s'adapter aux tendances. Cependant, c'est exactement ce que font les entreprises Big Tech de l'industrie.

Les principales applications comme Facebook, Instagram, Netflix et d'autres améliorent constamment leur expérience et s'adaptent aux nouveaux cadres et tendances.

Depuis peu, le bouche à oreille circule autour de ReactJS et de ses fonctionnalités impressionnantes.

La preuve de sa popularité est mieux décr ite par les applications qui utilisent ReactJS. Aujourd'hui, nous allons vous montrer la liste des applications les plus impressionnantes basées sur ReactJS.

La plupart sont : Facebook, Instagram, ADA, Netflix, Dropbox, Paypal, etc...

**Exemple :**

Regardons un exemple pour mettre en évidence la différence entre le HTML normal et React.

Essayons de reconstruire la liste de films de netflix en utilisant HTML, puis reconstruisez-la en utilisant React. Nous comparerons les deux versions plus tard.

# Netflix movies using HTML

Using HTML, we end up with this incomprehensible mess of a code:

<div class="nm-content-horizontal-row">

<ul class="nm-content-horizontal-row-item-container">

<li class="nm-content-horizontal-row-item" role="menuitem">

<a

class="nm-collections-title nm-collections-link"

href="https://www.netflix.com/tn-en/title/70079583"

data-uia="collections-title"

>

<img

class="nm-collections-title-img"

src="..."

data-title-id="70079583"

alt="The Dark Knight"

/>

<span class="nm-collections-title-img placeholder"></span>

<span class="nm-collections-title-name">The Dark Knight</span>

</a>

</li>

<li class="nm-content-horizontal-row-item" role="menuitem">

<a

class="nm-collections-title nm-collections-link"

href="https://www.netflix.com/tn-en/title/80013871"

data-uia="collections-title"

>

<img

class="nm-collections-title-img"

src="..."

data-title-id="80013871"

alt="American Sniper"

/>

<span class="nm-collections-title-img placeholder"></span>

<span class="nm-collections-title-name">American Sniper</span>

</a>

</li>

<li class="nm-content-horizontal-row-item" role="menuitem">

<a

class="nm-collections-title nm-collections-link"

href="https://www.netflix.com/tn-en/title/80064516"

data-uia="collections-title"

>

<img

class="nm-collections-title-img"

src="...."

data-title-id="80064516"

alt="The Revenant"

/>

<span class="nm-collections-title-img placeholder"></span>

<span class="nm-collections-title-name">The Revenant</span>

</a>

</li>

</ul>

</div>

# Netflix movies using React.js

On the other hand, after refactoring this code in React using components, we end up having a much cleaner code.  
Notice that <Movie/> tag is not part of HTML, it’s a tag that we created using React. It’s called a **React component**.

<Container>

<Row>

<Movie image={...} title=”The Dark Knight”

href=”https://www.netflix.com/tn-en/title/70079583”/>

<Movie image={...} title=”American Sniper”

href=”https://www.netflix.com/tn-en/title/80013871”/>

<Movie image={...} title=”The Revenant”

href=”https://www.netflix.com/tn-en/title/80064516”/>

</Row>

</Container>