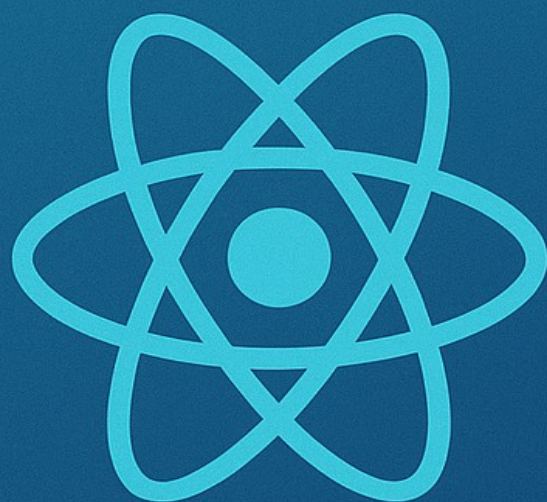


FORMATION REACT

NIVEAU
DÉBUTANT



React JS pour Débutants

Du Zéro à la Pratique

 **Installation et Premier Projet** 

Partie 1 : Débuter avec React (Pages 1-15)

Table des Matières - Partie 1

1. Introduction à React
2. Installation de Node.js
3. Créer votre Premier Projet React
4. Structure d'un Projet React
5. Comprendre les Composants

1. Introduction à React

Bienvenue dans le monde de React ! Si vous voulez créer des sites web modernes et interactifs, vous êtes au bon endroit. React est comme des blocs LEGO pour le web : vous créez des petits morceaux (composants) et vous les assemblez pour construire quelque chose de génial !

💡 *Pourquoi les développeurs adorent React ? Parce qu'ils peuvent enfin réutiliser leur code au lieu de tout réécrire comme des fous !*

Qu'est-ce que React ?

React est une bibliothèque JavaScript créée par Facebook (Meta) en 2013. Elle permet de :

- Créer des interfaces utilisateur interactives
- Réutiliser du code avec les composants
- Mettre à jour l'interface rapidement (Virtual DOM)
- Gérer l'état de votre application facilement

Pourquoi apprendre React ?

React est utilisé par des géants du web comme :

- **Facebook/Meta, Instagram, WhatsApp**
- **Netflix, Airbnb, Uber**
- **Twitter, Dropbox, Discord**

2. Installation de Node.js

Avant de créer votre premier projet React, vous devez installer Node.js. Node.js est comme le moteur qui fait tourner React sur votre ordinateur !

Étape 1 : Télécharger Node.js

1. Allez sur le site officiel : <https://nodejs.org>
2. Téléchargez la version **LTS (Long Term Support)** - c'est la version stable
3. Choisissez selon votre système : Windows, macOS ou Linux

✨ **Astuce : Toujours choisir la version LTS pour un développement stable !**

Étape 2 : Installer Node.js

Sur Windows :

4. Double-cliquez sur le fichier .msi téléchargé
5. Suivez l'assistant d'installation (cliquez sur 'Next' partout)
6. Cochez 'Automatically install necessary tools' si demandé
7. Cliquez sur 'Install' et attendez

Sur macOS :

8. Double-cliquez sur le fichier .pkg téléchargé
9. Suivez l'assistant d'installation
10. Entrez votre mot de passe admin si demandé

Sur Linux (Ubuntu/Debian) :

Ouvrez le terminal et tapez ces commandes :

```
# Mettre à jour les paquets
sudo apt update

# Installer Node.js et npm
sudo apt install nodejs npm
```

```
# Vérifier l'installation  
node --version  
npm --version
```

Étape 3 : Vérifier l'Installation

Ouvrez votre terminal (ou invite de commandes sur Windows) et tapez :


```
node --version
```


Vous devriez voir quelque chose comme : **v18.17.0**

Ensuite, vérifiez npm (le gestionnaire de paquets) :

```
npm --version
```

Vous devriez voir : **9.6.7** (ou une version similaire)

 *Si vous voyez ces versions, félicitations ! Vous avez survécu à l'installation. C'était le plus dur... je plaisante !*

 **Si les commandes ne fonctionnent pas, redémarrez votre terminal ou votre ordinateur !**

3. Créer votre Premier Projet React

Maintenant, la partie excitante : créer votre premier projet React ! Create React App est un outil officiel qui configure tout automatiquement. C'est comme avoir un assistant qui prépare votre cuisine avant de cuisiner !

Créer le Projet

Ouvrez votre terminal et naviguez vers le dossier où vous voulez créer votre projet :

```
# Exemple : aller sur le Bureau (Desktop)
cd Desktop

# Ou créer un dossier pour vos projets
mkdir mes-projets-react
cd mes-projets-react
```

Maintenant, créez votre projet React avec cette commande magique :

```
npx create-react-app mon-premier-projet
```

Note : Remplacez 'mon-premier-projet' par le nom que vous voulez (en minuscules, sans espaces, utilisez des tirets).

✨ **Exemples de noms valides : todo-app, portfolio-site, blog-perso, super-calculatrice**

Cette commande va :

11. Télécharger tous les fichiers nécessaires
12. Installer React et toutes ses dépendances
13. Configurer votre environnement de développement
14. Créer la structure de base du projet

⚠ **L'installation peut prendre 2-5 minutes selon votre connexion internet. Patience !**

💡 *Pendant que ça s'installe, vous pouvez aller vous faire un café. Ou trois.*

Démarrer votre Projet

Une fois l'installation terminée, entrez dans votre projet :

```
cd mon-premier-projet
```

Maintenant, démarrez le serveur de développement :

```
npm start
```

🎉 Magie ! Votre navigateur devrait s'ouvrir automatiquement à l'adresse :

<http://localhost:3000>

Vous devriez voir une page avec le logo React qui tourne. Félicitations, votre premier projet React fonctionne !

✨ **Pour arrêter le serveur, appuyez sur Ctrl+C dans le terminal.**

4. Structure d'un Projet React

Ouvrons notre projet dans un éditeur de code. Je recommande Visual Studio Code (gratuit). Explorons les fichiers que Create React App a créés pour nous !

Structure des Dossiers

```
mon-premier-projet/  
├─ node_modules/      ← Toutes les dépendances (NE PAS TOUCHER)  
├─ public/             ← Fichiers statiques  
│   ├─ index.html      ← Page HTML principale  
│   └─ favicon.ico     ← Icône du site  
├─ src/                ← Votre code source (ZONE DE TRAVAIL)  
│   ├─ App.js          ← Composant principal  
│   ├─ App.css         ← Styles du composant App  
│   ├─ index.js        ← Point d'entrée de l'application  
│   └─ index.css       ← Styles globaux  
├─ package.json       ← Configuration du projet  
└─ README.md          ← Documentation
```

Les Fichiers Importants

1. public/index.html

C'est la page HTML de base. React injectera votre application dans la div avec `id='root'`.

```
<!DOCTYPE html>  
<html lang="fr">  
  <head>  
    <meta charset="utf-8" />  
    <title>Mon Premier Projet React</title>  
  </head>  
  <body>  
    <div id="root"></div>  
    <!-- React s'injecte ici -->  
  </body>  
</html>
```

2. src/index.js

C'est le point d'entrée de votre application React. Il connecte React au DOM.

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

3. src/App.js

C'est votre composant principal. C'est ici que nous allons travailler !

```
import './App.css';

function App() {
  return (
    <div className="App">
      <h1>Bonjour React !</h1>
      <p>Mon premier composant</p>
    </div>
  );
}

export default App;
```

💡 *App.js est comme la salle de séjour de votre maison : c'est là que tout commence !*

5. Comprendre les Composants

Un composant React est comme une brique de LEGO : une pièce réutilisable qui affiche quelque chose à l'écran. Créons notre premier composant personnalisé !

Modifier App.js

Ouvrez **src/App.js** et remplacez tout le contenu par :

```
import './App.css';

function App() {
  return (
    <div className="App">
      <h1>🚀 Mon Premier Projet React</h1>
      <p>Bienvenue dans le monde de React !</p>
      <button>Cliquez-moi !</button>
    </div>
  );
}

export default App;
```

Sauvegardez (Ctrl+S ou Cmd+S). Le navigateur se rafraîchit automatiquement !

C'est le **Hot Reload** - une fonctionnalité géniale de React.

Créer un Composant Personnalisé

Créez un nouveau fichier **src/Carte.js** :

```
function Carte() {
  return (
    <div style={{
      border: '2px solid #3498db',
      borderRadius: '10px',
      padding: '20px',
      margin: '20px',
      backgroundColor: '#ecf0f1'
    }}>
      <h2>Ma Première Carte</h2>
      <p>Ceci est un composant React réutilisable !</p>
    </div>
  );
}
```

```
    </div>  
  );  
}  
  
export default Carte;
```

Utiliser votre Composant

Retournez dans **src/App.js** et utilisez votre composant :

```
import './App.css';
import Carte from './Carte'; // Importer le composant

function App() {
  return (
    <div className="App">
      <h1>🚀 Mon Premier Projet React</h1>

      { /* Utiliser le composant 3 fois */ }
      <Carte />
      <Carte />
      <Carte />
    </div>
  );
}

export default App;
```

Sauvegardez et regardez votre navigateur. Vous voyez 3 cartes identiques ? C'est la magie des composants : **écrivez une fois, utilisez partout !**

💡 *Réutiliser du code, c'est comme réutiliser vos chaussettes : écologique et pratique !*

Anatomie d'un Composant

Décortiquons un composant :

- 15. **function Carte()** : Nom du composant (toujours avec une majuscule !)
- 16. **return (...)** : Ce que le composant affiche
- 17. **<div>...</div>** : JSX - qui ressemble à HTML mais c'est du JavaScript
- 18. **export default Carte** : Permet d'utiliser ce composant ailleurs

✨ **Règle d'or : Un composant = Une fonction qui retourne du JSX !**

Exercice Pratique

Créez un composant Bouton personnalisé dans **src/Bouton.js** :

```
function Bouton() {
  return (
    <button style={{
      backgroundColor: '#3498db',
      color: 'white',
      padding: '15px 30px',
      fontSize: '18px',
      border: 'none',
      borderRadius: '5px',
      cursor: 'pointer',
      margin: '10px'
    }}>
      Super Bouton !
    </button>
  );
}

export default Bouton;
```

Puis utilisez-le dans App.js :

```
import './App.css';
import Carte from './Carte';
import Bouton from './Bouton';

function App() {
  return (
    <div className="App">
      <h1>🚀 Mon Premier Projet React</h1>

      <Carte />
      <Bouton />
      <Bouton />
    </div>
  );
}

export default App;
```

🎉 **Félicitations !**

Vous avez créé votre premier projet React avec des composants personnalisés !
Dans la suite (Partie 2), nous allons créer une vraie application avec des fonctionnalités interactives !

Fin de la Partie 1

Dans la Partie 2, nous créerons une application de compteur interactive !