

React js

I Installation de environnement de travail

Télécharger Node Js à l'adresse suivante :

<https://nodejs.org/en/download/>

[HOME](#) | [ABOUT](#) | [DOWNLOADS](#) | [DOCS](#) | [GET INVOLVED](#) | [SECURITY](#) | [NEWS](#)


Downloads


Latest LTS Version: 12.16.3 (includes npm 6.14.4)


Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS
Recommended For Most Users

Current
Latest Features


Windows Installer
node-v12.16.3-x64.msi


macOS Installer
node-v12.16.3.pkg


Source Code
node-v12.16.3.tar.gz

Windows Installer (.msi)
Windows Binary (.zip)
macOS Installer (.pkg)
macOS Binary (.tar.gz)
Linux Binaries (x64)
Linux Binaries (ARM)
Source Code

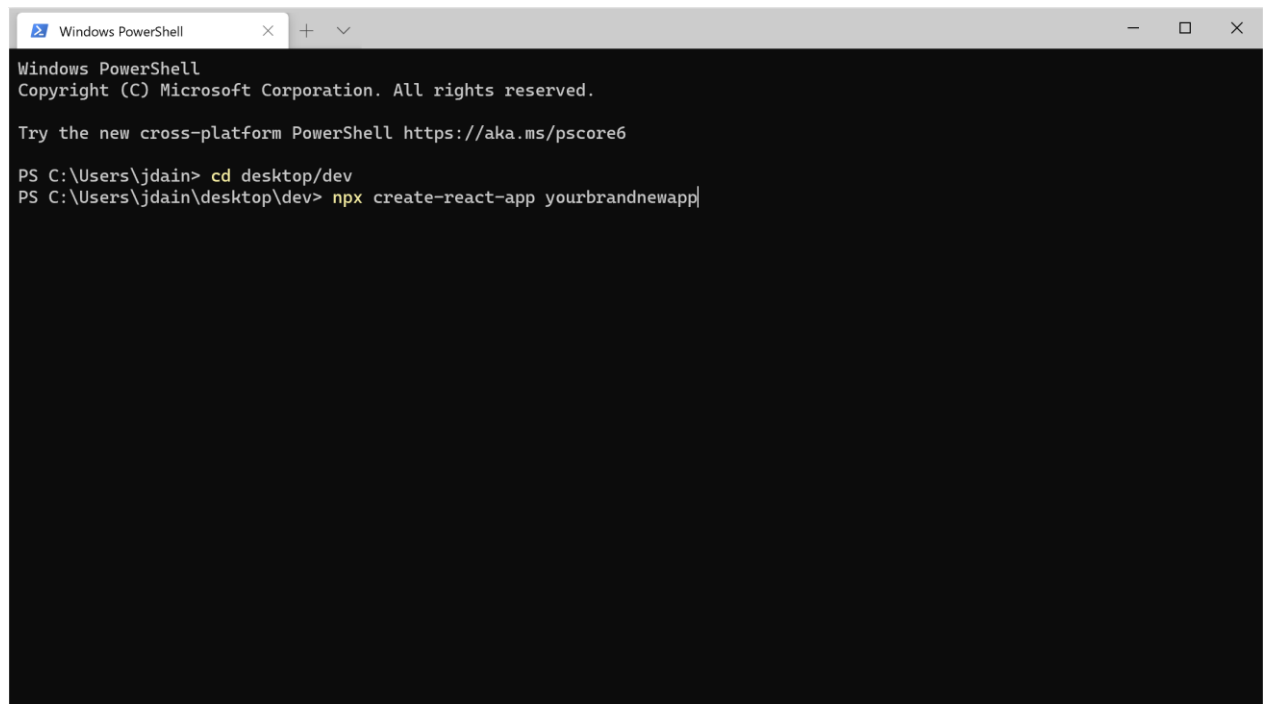
32-bit	64-bit
32-bit	64-bit
64-bit	
64-bit	
64-bit	
ARMv7	ARMv8
node-v12.16.3.tar.gz	

Additional Platforms

SmartOS Binaries
Docker Image
Linux on Power LE Systems
Linux on System z

64-bit
Official Node.js Docker Image
64-bit
64-bit

Ouvrir le terminal, se placer dans le dossier où l'on veut créer une nouvelle application web react js en utilisant la ligne de code suivante :

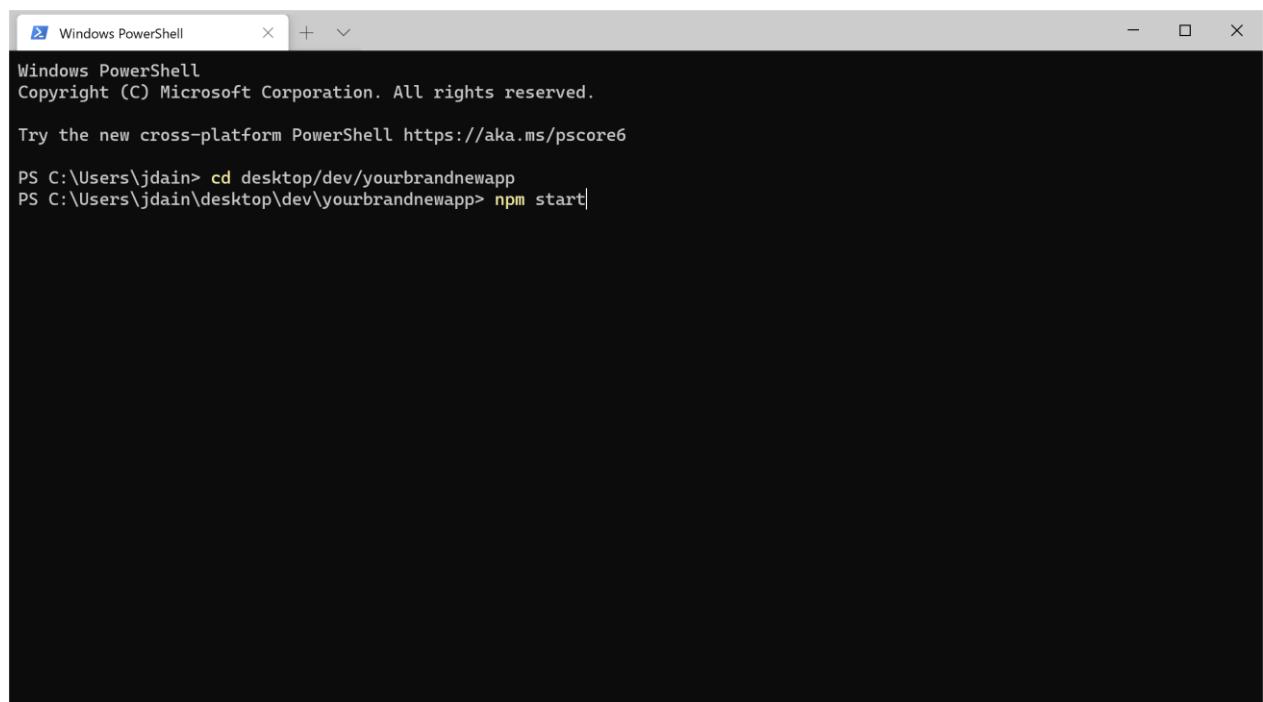


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\jdain> cd desktop/dev
PS C:\Users\jdain\desktop\dev> npx create-react-app yourbrandnewapp
```

Ensuite Naviguer dans le dossier et lancer le serveur comme ceci :

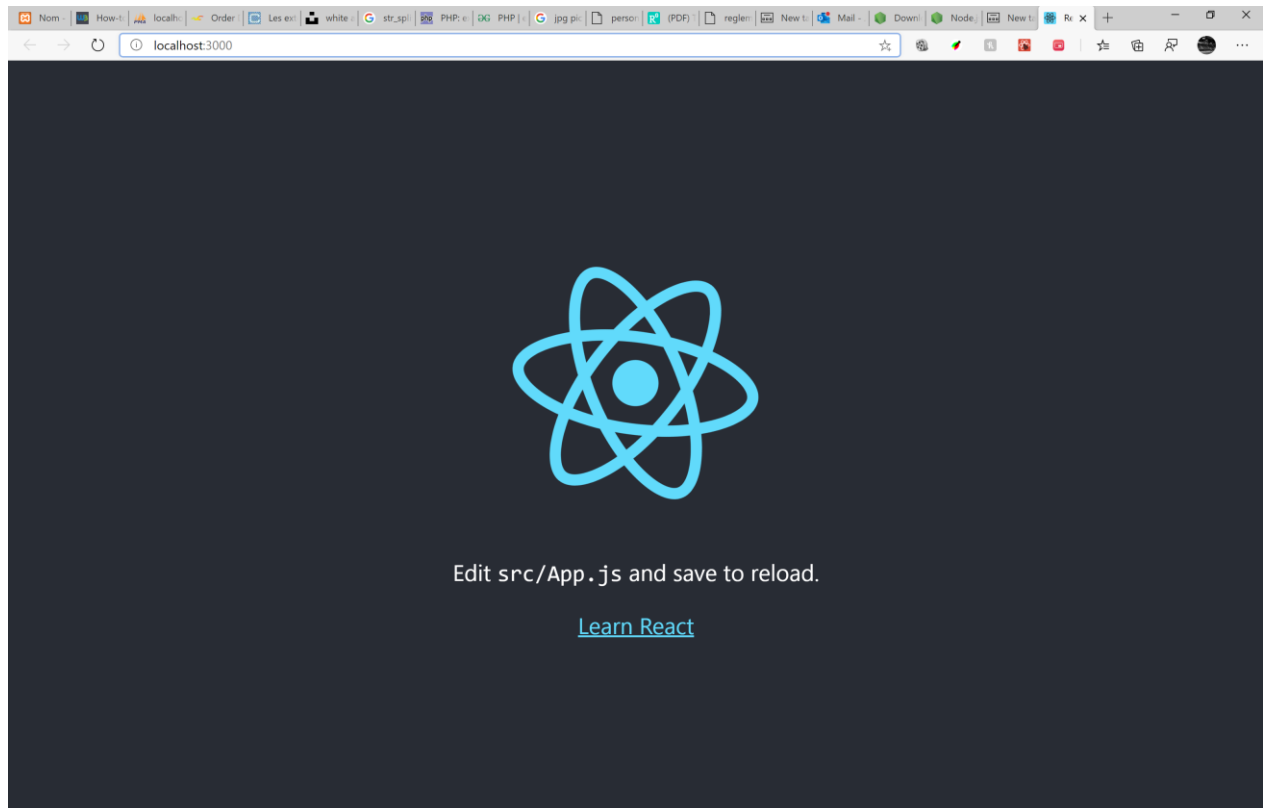


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

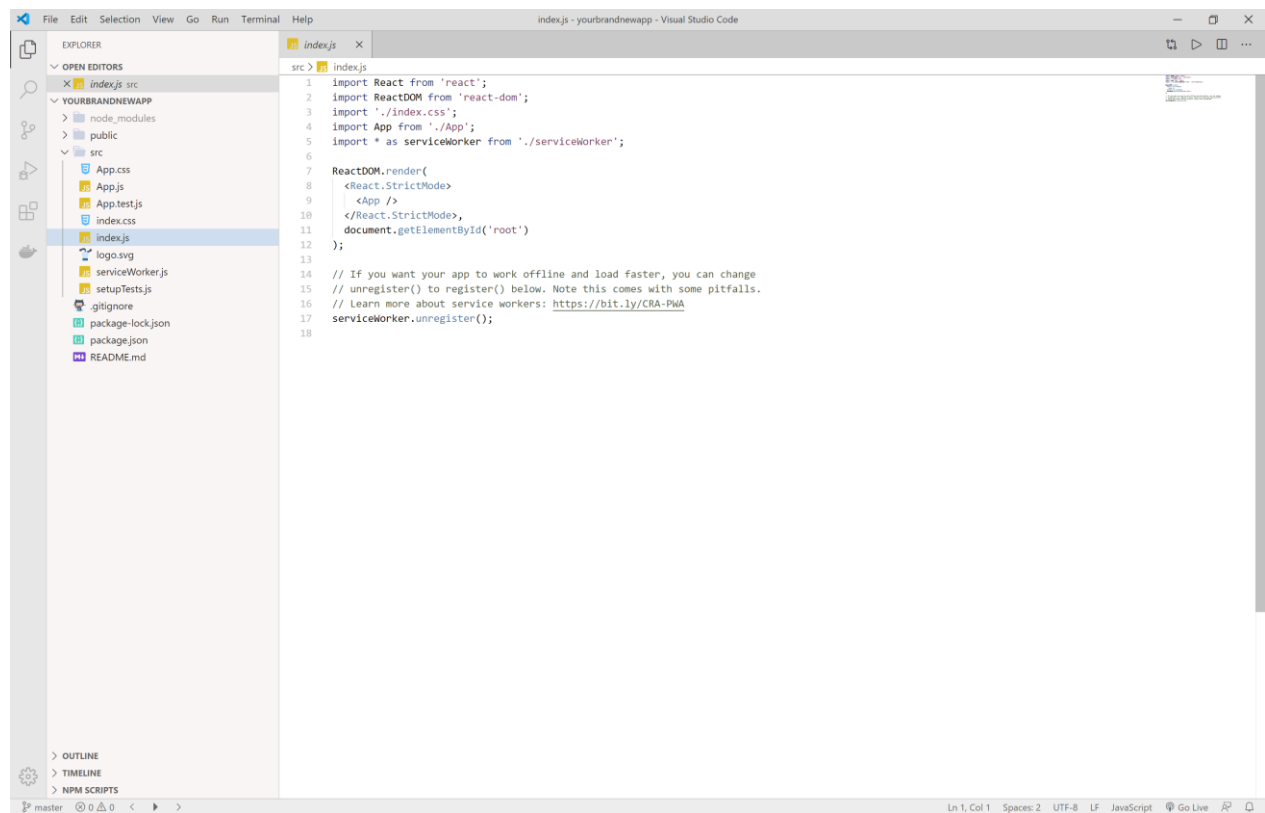
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\jdain> cd desktop/dev/yourbrandnewapp
PS C:\Users\jdain\desktop\dev\yourbrandnewapp> npm start
```

Une fenetre devrais s'ouvrir dans votre navigateur



Modifier l'Application



2 – ReactDOM

ReactDOM est importer dans l'index.js il est ensuite appeler dans ce même fichier et on appelle la methode render pour afficher a l'ecran des éléments comme du html normal.

3-Composants

Le grand avantage de React est de pouvoir créer des composants réutilisables cependant il existe de types de composants : les composants Stateless et les Composants avec un états.

3.1 Composants Stateless

Pour créer ces composants nous utilisons les fonctions chaque composants aura un nom commençant par une majuscule chacune de ces fonctions doit être contenu dans un fichier avec son nom.

Créons un premier composants, une carte pour afficher les informations d'un compte :

```
import React from "react";

const styles = {
  img:{
    borderRadius: "50%",
    backgroundColor: "lightblue",
    margin: "auto",
    marginTop:5,
    display: "block"

  },
  container:{
    backgroundColor: "#F5F5F5",
    width:"50%",
    margin: "auto",
    borderRadius: 14,
    paddingLeft: 22,
    paddingTop: 8,
    paddingBottom: 8
  }
}

function Carte() {
  return (
    <div style={styles.container}>
      
      <h1>Your Display Name</h1>
    </div>
  )
}
```

```
        <p>@accountname</p>
        <p>Bio...</p>
    </div>
)
```

```
}
```

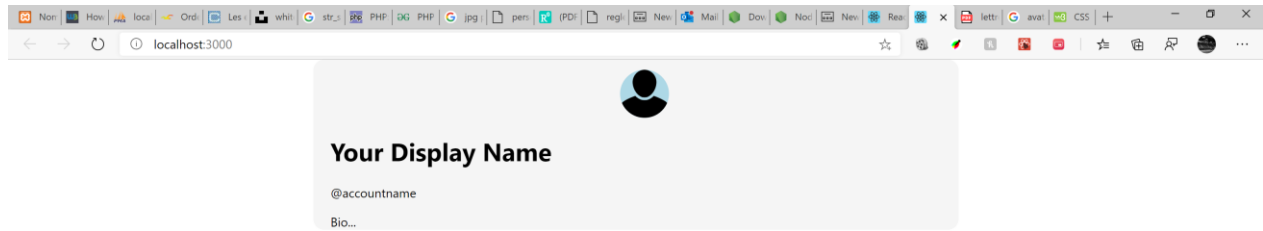
```
export default Carte;
```

```
import React from 'react';
import Carte from "../Carte";
```

```
function App() {
  return (
    <Carte />
```

```
  );
}
```

```
export default App;
```



Nous Vons Creer notre premier composant React on peut l'utiliser dans tous les fichiers mais un composant tel que celui-ci est très peu utile on aurait pu faire la même chose en html classique nous allons le modifier pour qu'il prenne des attributs et affiche ces attributs.

```
import React from "react";

const styles = {
  img:{
    borderRadius: "50%",
    backgroundColor: "lightblue",
    margin: "auto",
    marginTop:5,
    display: "block"
  },
  container:{
    backgroundColor: "#F5F5F5",
    width:"50%",
    margin: "auto",
    borderRadius: 14,
    paddingLeft: 22,
```

```

        paddingTop: 8,
        paddingBottom: 8
    }
}

function Carte(props) {
    return (
        <div style={styles.container}>
            <img style={styles.img} src={props.photo} height="124" width="124" />
            <h1>{props.nom}</h1>
            <p>@{props.username}</p>
            <p>{props.bio}</p>
        </div>
    )
}

export default Carte;

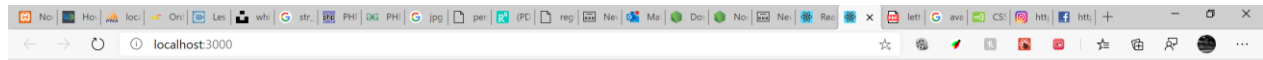
import React from 'react';
import Carte from "./Carte";

function App() {
    return (
        <Carte photo="https://scontent-cdg2-1.cdninstagram.com/v/t51.2885-19/s150x150/93574581_229247324994241_4522028129446789120_n.jpg?_nc_ht=scontent-cdg2-1.cdninstagram.com&_nc_ohc=j9tkT6zVLscAX9ag1xa&oh=9ee504d5049dba80fa0e581552a28f29&oe=5EDA2B7E" nom="Yanis Fatmi" username="Fatmax" bio="Je suis le plus fort en php et en web en général 📧" />
    );
}

export default App;

```

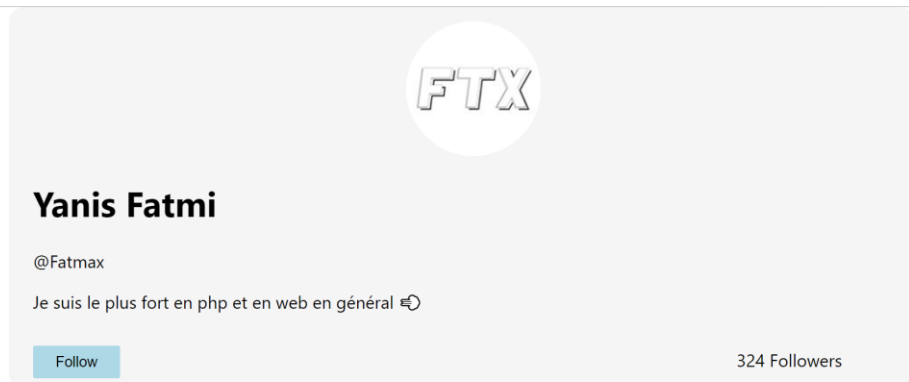
On utilise les attributs appeler props pour ajouter des éléments a notre design.



Maintenant ajoutons a ce composant un bouton s'abonner pour mettre a jour le nombre d'abonner de ce bon vieux fatmax.

2.3 Les Composants d'etats

Nous allons désormais utiliser une approche par classe pour créer un compteur du nombre d'abonnées.





Yanis Fatmi

@Fatmax

Je suis le plus fort en php et en web en général 🇪🇺

Follow

346 Followers

On observe que en pressant le bouton Yanis a gagner des abonnés

```
import React from "react";

const styles = {
  img: {
    borderRadius: "50%",
    backgroundColor: "lightblue",
    margin: "auto",
    marginTop: 5,
    display: "block"
  },
  container: {
    backgroundColor: "#F5F5F5",
    width: "50%",
    margin: "auto",
    borderRadius: 14,
    paddingLeft: 22,
    paddingTop: 8,
    paddingBottom: 8
  },
  btn:{
    backgroundColor: "lightblue",
    border: "none",
    outline: 0,
    height: 30,
    width: 80
  },
  followerCount:{
    display: "inline-flex",
```

```

        paddingLeft: "70%"
    }
}

```

```

class Carte extends React.Component {

```

```

    constructor(){
        super();
        this.state = {
            followers: 323
        }
        this.handleClick = this.handleClick.bind(this);
    }

```

```

    handleClick(){
        let currentCount = this.state.followers;
        this.setState({
            followers: currentCount+1,
        })
    }

```

```

    render() {
        return (
            <div style={styles.container}>
                <img style={styles.img} src={this.props.photo} height="124" width
="124" />
                <h1>{this.props.nom}</h1>
                <p>@{this.props.username}</p>
                <p>{this.props.bio}</p>
                <button onClick={this.handleClick} style={styles.btn}>Follow</but
ton>
                <p style={styles.followerCount}>{this.state.followers} Followers<
/p>
            </div>
        )
    }

```

```

}

```

```

export default Carte;

```

On note l'utilisation du `this` pour les props et les fonctions.