

## 【React編】

### 1.nvm, Node.js, npmのインストール

<https://zenn.dev/antez/articles/a9d9d12178b7b2>

上記のサイトを参考にして、Node.jsのインストールまで完了させる。

「nvm use x.x.x」を行った後、「npm -v」でnpmが入っていることまで確認する。

「nvm install x.x.x」でインストールするLTSは18.17.0とする。

```
C:\Users\aosan>nvm list available
```

CURRENT	LTS	OLD STABLE	OLD UNSTABLE
20.5.0	18.17.0	0.12.18	0.11.16
20.4.0	18.16.1	0.12.17	0.11.15
20.3.1	18.16.0	0.12.16	0.11.14
20.3.0	18.15.0	0.12.15	0.11.13
20.2.0	18.14.2	0.12.14	0.11.12
20.1.0	18.14.1	0.12.13	0.11.11
20.0.0	18.14.0	0.12.12	0.11.10
19.9.0	18.13.0	0.12.11	0.11.9
19.8.1	18.12.1	0.12.10	0.11.8

```
C:\Users\aosan>npm -v  
9.6.7
```

## 2. 簡単なプロジェクトでサーバーを動かしてみる。

自身のPC上に適当なディレクトリを作成&コマンドプロンプト上で移動する。  
その後、「`npx create-react-app app-react`」でEnter  
(`app-react`はプロジェクト名になるので何でもよい。)  
「`cd app-react`」で中に入り、「`npm start`」を実行する。  
画面に表示されたLocal: の右側にあるURLをコピーしてブラウザに貼り付ける。

```
Compiled successfully!

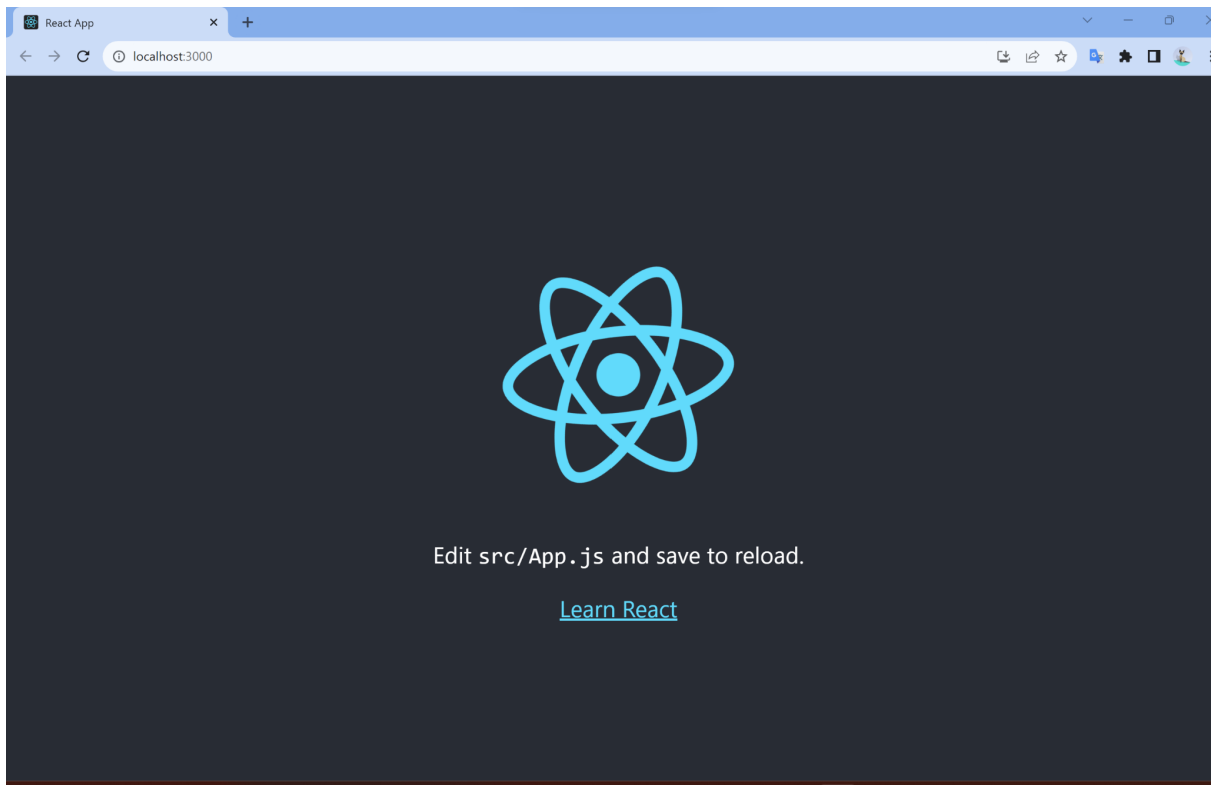
You can now view app-react in the browser.

Local:      http://localhost:3000
On Your Network:  http://192.168.189.109:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

以下の画面が表示されればOK



### 3. PythonのFlaskで作った簡単なサーバーと連携してみる。

app-reactディレクトリがある場所と同じ階層で「app-flask」ディレクトリを作成する。  
anacondaのflask用仮想環境に入って、「pip install flask-cors」をする。

※ [UdemyのPython講座](#)を参考

app-flask配下にapp.pyファイルを作成する。内容は添付のapp.pyを参照  
仮想環境に入ったまま、「python app.py」でサーバーをたてる。

app-reactディレクトリにもどる。

app-react/src/App.jsを添付のApp.jsの内容に書き換える。

その後、コマンドプロンプトでapp-react直下に移動して、「npm start」を実行する。

先ほどと同様に、Local: の右側にあるURLへアクセスして以下の画面が表示されればOK

