



Annexes : Notions en React JS

Annexe 1 : Template literals

```
import "./styles.css";
export default function App() {
  const salutations="salut";
  const nom="Rami"
  const className="class1"
  return (
    <div className="App">
    <h1 className={` ${className}`} >`${salutations}  ${nom}`</h1>

    </div>
  );
}
```

Le rendu :

salut Rami

Exemple 2 :

```
import "./styles.css";
import React,{useState} from 'react'
const diplomes=['bac','technicien','technicien
spécialisé','licence','master']
const salutations="salut";
const nom="Rami"

export default function App() {
const [current,setCurrent]=useState(0)

  return (
    <div className="App">
      <h1 >`${salutations}  ${nom}`</h1>
      <ul>
        {diplomes.map((d,i)=>{
          let className=''
```



```
        if (current===i) {className='class1
active'}else{className='class1'}
        return (<li className={`${className}`} key={d}
onClick={() =>setCurrent (i) }>{d}</li>
        )})
    </ul>
</div>
);
}
```

CSS

```
.App {
  font-family: sans-serif;
  text-align: center;
}
.class1{
  list-style:none ;
  color:green
}
.active {
  background-color: #666;
  color: white;
}
```

Le rendu :

salut Rami

bac
technicien
technicien spécialisé
licence
master



Annexe 2 : L'opérateur conditionnel ternaire

```
import "./styles.css";
import React, { useState } from "react";
export default function App() {
  const [isMember, setIsMember] = useState(false);
  return (
    <div className="App">
      <label>est membre?</label>
      <input
        type="checkbox"
        onChange={() => setIsMember(!isMember)}
        checked={isMember}
      />
      <p>votre remise est {isMember?0.2:0.1}</p>
    </div>
  );
}
```

Le rendu :

est membre?

votre remise est 0.2

est membre?

votre remise est 0.1



Annexe 3 : Destructuring

```
import './styles.css';
import React from 'react'
import Presentation from './Presentation'
const personne={
  nom:"fatihi",
  age:23,
  adresse:{rue:14,ville:"casa"},
}
export default function App() {
  return (
    <div className="App">
      <Presentation personne={personne}/>
    </div>
  );
}
```

Composant Presentation

```
import React from 'react'
function Presentation(prop) {
  return(<div>
    <h1>nom: {prop.personne.nom}</h1>
    <h3>age: {prop.personne.age}</h3>
    <hr/>
    <h4>Adresse:</h4>
    <p>Rue: {prop.personne.adresse.rue+' ' }
      Ville: {prop.personne.adresse.ville}</p>
    </div> )
}
export default Presentation
```



Composant Presentation en utilisant destructing object

```
import React from 'react'
function Presentation({personne}) {
  const {nom,age,adresse:{rue},adresse:{ville}}=personne
  return(<div>
    <h1>nom:{nom}</h1>
    <h3>age:{age}</h3>
    <hr/>
    <h4>Adresse:</h4>
    <p>Rue:{rue+'    '}
      Ville:{ville}</p>
    </div> )
}
export default Presentation
```



Annexe 4 : La méthode filter

Les noms des personnes qui ont l'âge est supérieur à 35

```
import "./styles.css";
import React from "react";
const personnes = [
  { nom: "Rami", age: 33, estMember: true },
  { nom: "Fatihi", age: 24, estMember: false },
  { nom: "Chakib", age: 45, estMember: true },
  { nom: "Mounir", age: 37, estMember: false }
];
export default function App() {
  return (
    <div className="App">
      <ul>
        {personnes
          .filter((p) => {
            return p.age >= 35;
          })
          .map((p) => {
            return <li>{p.nom}</li>;
          })}
      </ul>
    </div>
  );
}
```

Le rendu :

- Chakib
- Mounir