



Direction Régionale RABAT-SALÉ-KENITRA

Examen régional de fin de module
Année de Formation 2022/2023



Nom :

Prénom :

Groupe :

Établissement :

Réservé à l'établissement Code :

Code module : M204
Intitulé du module : Développement Front-end

Filière :	Développement Digital Option Applications Mobiles	Durée :	2h30min
Année :	2 ° A	Note finale :	/ 40
Nom & Prénom du correcteur :		Émargement :	

Partie Théorique

/20 pts

Exercice 1 :

/4 pts

Citez quelques avantages de Redux.

.....

.....

.....

.....

.....

Exercice 2 :

/10 pts

Choisir la bonne réponse

1- Comment appeler en JSX une fonction mafonction() depuis un élément h2 ?

/2 pts

- A. `<h2>{mafonction()}</h2>`
- B. `<h2>${mafonction()}</h2>`
- C. `<h2>{mafonction}</h2>`
- D. `<h2>${mafonction}</h2>`

2- Comment incorporer ce style inline qui spécifie font-size :14px et color : green avec du code JSX ? /2 Pts

- A. style={{font-size:14,color:'green'}}
- B. style={{fontSize:'14px',color:'green'}}
- C. style={fontSize:'14px',color:'green'}
- D. style={{font-size:14px,color:'green'}}

3- On utilise quelle fonction pour modifier l'état d'un composant React? /2 Pts

- A. this.getState()
- B. this.State = {}
- C. this.setState
- D. this.changeState

4- La méthode componentDidMount est invoquée quand /2 Pts

- A. L'état du composant est mis à jour
- B. Le composant est créé pour la première fois
- C. Le composant est supprimé
- D. Le composant est retiré du DOM

5- Comment faire pour rendre une liste dynamique de composants en fonction des valeurs d'un tableau. /2 Pts

- A. Inclure une boucle for ou while
- B. Utiliser la méthode reduce du tableau
- C. Utiliser le composant <Each />
- D. Utiliser la méthode Array.map()

Exercice 3 :

/6 Pts

Expliquer l'intérêt des codes suivants :

Code 1

```
componentDidMount(){  
  fetch('https://jsonplaceholder.typicode.com/users')  
  .then((response)=>{ console.log(response); return response.json();})  
  .then((users)=>{this.setState({utilisateurs:users})})  
}
```



Réservé à l'établissement Code :

Page 2 / 7

Surveillant1 :

Surveillant2 :

Num Module : M204

Code 2

```
import React, { useState, useEffect } from 'react';

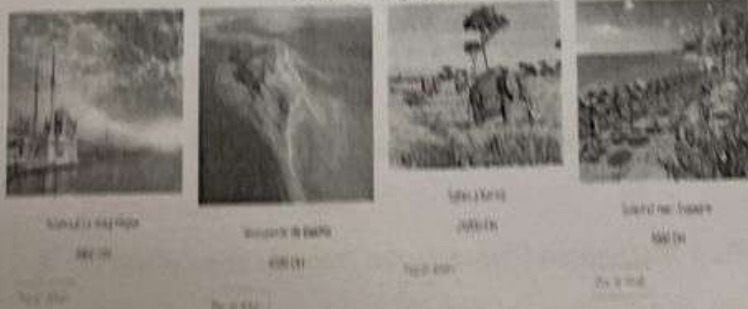
function Example() {
  const [count, setCount] = useState(0);
  useEffect(() => { document.title = `vous avez cliqué ${count} fois`; });
  return (
    <div>
      <p>vous avez cliqué {count} fois</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

Partie Pratique :

/20 pts

On souhaite dans cette partie réaliser un site web affichant un ensemble de voyages, lors du clic sur le bouton **plus de détails**, on affichera les informations du voyage choisi dans une nouvelle page comme les deux figures suivantes :

Page d'accueil
Circuits réguliers



Page contenant les détails du voyage



En utilisant le tableau suivant :

```
const products = [  
  {  
    id: 1,  
    intitule: 'Istanbul La magnifique',  
    prix: '8900 DH',  
    source : 'image1.jpg'  
  },  
  {  
    id: 2,  
    intitule: 'Découverte de Dakhla',  
    prix: '4500 DH',  
    source : 'image2.jpg'  
  },  
  {  
    id: 3,  
    intitule: 'Safari à Kenya',  
    prix: '25000 DH',  
    source: 'image3.jpg'  
  },  
  {  
    id: 4,  
    intitule: 'Soleil et mer: Espagne',  
    prix: '7000 DH',  
    source: 'image4.jpg'  
  },  
];
```

Travail à faire :

- 1- Ecrire le code du (des) composant(s) permettant d'afficher la liste des voyages comme présentée dans la page 1 du site. /8 pt
- 2- Ecrire le code du (des) composant(s) permettant d'afficher les détails du voyage. /6 pt
- 3- Ecrire le code du composant App.is en utilisant les routages. /6 pt

1/

- Séparation claire de la logique de l'application
- Prévisibilité
- Facilité de test
- Possibilité de développer des applications distribuées
- Possibilité de "undo" et "redo"

ex2

1-c

2-b

3-c

4-d

5-d

Exercice 3/ Full Stack::👉

Code 1 est utilisé pour effectuer une requête HTTP à l'aide de la fonction `fetch()` pour récupérer des données à partir de l'URL spécifiée, qui renvoie une réponse en format JSON. La méthode `then()` est utilisée pour traiter cette réponse, en utilisant `console.log(response)` pour afficher la réponse dans la console, puis en utilisant `response.json()` pour convertir la réponse en un objet JavaScript. Enfin, la méthode `setState()` est utilisée pour mettre à jour l'état de l'application en utilisant les données récupérées.

ex 3

code 2

un composant React pour gérer l'état (nombre de clics sur un bouton) et les effets secondaires (mise à jour du titre de la page en fonction des clics). `useState` permet de déclarer une variable `count` et une fonction pour la mettre à jour, tandis que `useEffect` permet de gérer les effets secondaires (mise à jour du titre de la page) en fonction de cette variable. Le composant affiche également un message indiquant le nombre de clics sur le bouton.

Partie pratique- full stack ::👉👉

1/

```
import React from 'react';
```

```
const ProductList = (props) => {  
  return (  
    <div>  
      {props.products.map((product) => (  
        <Product  
          key={product.id}  
          id={product.id}  
          intitule={product.intitule}  
          prix={product.prix}  
          source={product.source}  
        />  
      ))}  
    </div>  
  );  
};
```

```
const Product = (props) => {  
  return (  
    <div>  
      <h2>{props.intitule}</h2>  
      <img src={props.source} alt={props.intitule} />  
      <p>Prix: {props.prix}</p>  
    </div>  
  );  
};
```

```
export default ProductList;
```

2/

```
import React from 'react';
```

```
const ProductDetail = (props) => {  
  return (  
    <div>  
      <h1>{props.intitule}</h1>  
      <img src={props.source} alt={props.intitule} />  
      <p>{props.prix}</p>  
      <p>{props.description}</p>  
    </div>  
  );  
};
```

```
export default ProductDetail;
```

3//

```
import React, { Component } from 'react';  
import { BrowserRouter as Router, Route } from 'react-router-dom';  
import ProductList from './ProductList';  
import ProductDetail from './ProductDetail';
```

```
class App extends Component {  
  render() {
```

```

return (
  <Router>
    <div>
      <Route
        exact
        path="/"
        render={({props}) => <ProductList {...props} products={products} />}
      />
      <Route
        path="/product/:id"
        render={({props}) => (
          <ProductDetail
            {...props}
            products={products}
            product={products.find((p) => p.id === Number(props.match.params.id))}
          />
        )}
      />
    </div>
  </Router>
);
}
}

```

```

export default App;

```