



5. Travaux pratique

5.1. Quiz

- 1) **Quelle est la commande pour créer un projet React**
 - a) npx create-react-app
 - b) npm create-react-app
 - c) npx create-react-app myApp
 - d) npm create-react-app myApp
- 2) **c'est quoi myApp dans cette instruction**
npx create-react-app myApp
 - a) le dossier dans le quel va être créer l'application
 - b) le type de l'application
 - c) le nom de la nouvelle application
 - d) fait référence a une application existant
- 3) **quelle est la commande pour démarrer le React local development**
 - a) npm start
 - b) npm build
 - c) npm run dev
 - d) npm serve
- 4) **quelle est le local host port utilisé par React Développement server**
 - a) 3000
 - b) 5000
 - c) 8080
 - d) 3500
- 5) **Pour développer et exécuter React code, Node.js est obligatoire**
 - a) 3000
 - b) 5000
 - c) 8080
 - d) 3500
- 6) **Quel est le rendu du composant Voiture**

```
const element=document.getElementById("root");
const root=ReactDOM.createRoot(element)
function Voiture(){
  const marque="Toyota"
  return (
    <>
      {marque==="Toyota" && <h1>Toyota</h1>}
      {marque==="Honda" && <h1>Honda</h1>}
    </>)}
root.render(<Voiture/>)
```

- a) Toyota
Honda
- b) Honda
- c) Toyota



- 7) la copie de DOM qui est dans la mémoire est appelée :
- a) virtual DOM
 - b) react DOM
 - c) local DOM
 - d) copy DOM
- 8) le nom de React composant doit avoir la première lettre en majuscule :
- a) oui
 - b) non
- 9) quelle opération peut on utiliser pour avoir un conditionnel rendu :
- a) ?
 - b) &
 - c) &&
 - d) ||
 - e) :
 - f) ??
- 10) quelle outil utilise React pour compiler JSX :
- a) babel
 - b) JSX Compiler
 - c) React Router
 - d) ReactDOM
- 11) quelle la correcte syntaxe pour importer Component de React:
- a) import React.Component from 'react'
 - b) import {Component} from 'react'
 - c) import [Component] from 'react'
 - d) import Component from 'react'
- 12) React sépare user interface dans les composants, comment les composants sont combinés pour créer une user interface complète :
- a) On regroupe les composants dans un dossier
 - b) Avec webpack
 - c) Avec imprecation des composants
 - d) En découpant le code JSX
- 13) Que retourne le code suivant:

```
const element=document.getElementById("root");
const root=ReactDOM.createRoot(element)
function Voiture(){
const marque='Honda'
const chevaux=8
const voiture={marque,chevaux}
return ( <>
  <h3>Marque:{voiture.marque}</h3>
  <h3>chevaux:{voiture.chevaux}</h3>
  </>)
}
root.render(<Voiture/>)
```

- a) erreur
- b) Marque:Honda
chevaux:8
- c) Marque :
chevaux :
- d) N'affiche rien

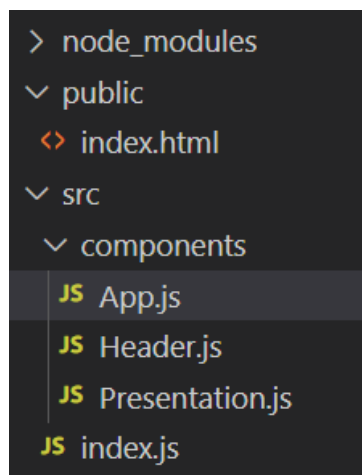
5.2. TP1 fonctionnels composants :



Application mon premier fonctionnel composant

Salut Rami Ahmed

La structure du projet



Le composant App

On dispose des composants fonctionnels suivants :

Header.js

```
import React from "react";
export default function Header(){
const titre="Application mon premier fonctionnel composant"
return (
  <div>
    <h1>ici afficher le titre</h1>
    <hr/>
  </div>
)
```

Presentation.js

```
import React from 'react'
export default function Presentation(props){
  const nom="Rami"
  const prenom="Ahmed"
  return (
    <div >

      <h2>Salut -nom prenom </h2>

    </div>
  )
}
```

App.js

```
import React from "react"
import Header from "./Header"
import Presentation from "./Presentation"

function App(){
  return(
    <div>
      <Header/>
      <br/>
      <Presentation />
    </div>
  )
}
export default App
```

index.js

```
// 1) importer React et ReactDOM
import React from 'react';
import ReactDOM from 'react-dom/client'
import App from './components/App';

const element=document.getElementById("root");
const root=ReactDOM.createRoot(element)

root.render(<App/>)
```

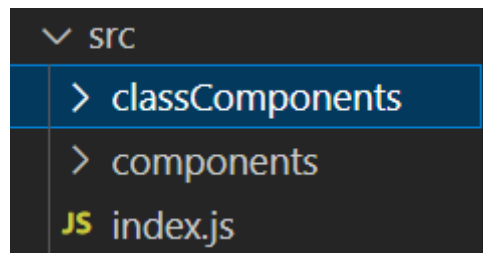
Créer ces trois composants dans le dossier components puis créer le fichier index.js qui se trouve directement dans le dossier src

Faire les changements nécessaires pour Afficher le titre,le nom et le prenom

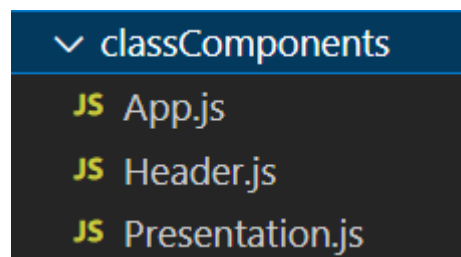


5.3. TP1 classes composants :

Ajouter un autre dossier classComponents



Le dossier classComponents contient les classes composants App,Header et Presentation



Header.js

```
import React from "react";
export default class Header extends React.Component{
  constructor(props) {
    super(props)
    this.titre="Application mon premier classe composant"
  }
  render(){
    return (
      <div>
        <h1>ici le titre</h1>
        <hr/>
      </div>
    )
  }
}
```



Presentation.js

```
import React from 'react'
export default class Presentation extends React.Component{
  constructor(props) {
    super(props)
    this.nom='Rami';
    this.prenom="Ahmed"
  }
  render(){
    return (
      <div >
        <h2>Salut ici le nom et prenom </h2>
      </div>
    )
  }
}
```

App.js

```
import React from "react"
import Header from "./Header"
import Presentation from "./Presentation"
class App extends React.Component{
  render(){
    return(<div>
      <Header/>
      <br/>
      <Presentation />
    </div>)
  }
}
export default App
```

index.js

```
// 1) importer React et ReactDOM
import React from 'react';
import ReactDOM from 'react-dom/client'
import App from './classComponents/App';
const element=document.getElementById("root");
const root=ReactDOM.createRoot(element)
root.render(<App/>)
```



Créer ces trois classes composants dans le dossier classComponents puis créer le fichier index.js qui se trouve directement dans le dossier src

Faire les changements nécessaires pour Afficher le titre, le nom et le prenom

5.4. TP2 composant Voiture données à partir d'un objet :

Fonctionnel composant

Créer le composant Voiture qui permet d'afficher les informations d'une voiture

Information voiture

voiture matricule:**19873X9**

marque:Toyota nombre de chevaux:8

Les informations sont récupérées de l'objet javascript myVoiture, la matricule doit prendre la couleur rouge

Voiture.js

```
import React from "react";
export default function Voiture(){
  const myVoiture={matricule:'19873X9',
  marque:'Toyota',
  chevaux:8}
  return(
    <div>
      <h1 style={{color:'green'}}>Information voiture</h1>
      <h4>voiture matricule:<span>-----</span></h4>
      <h5>marque:----- nombre de chevaux:-----</h5>
    </div>
  )
}
```

1. Compléter le code pour avoir le rendu ci-dessus
2. Créer la classe composant équivalent en complétant la classe Voiture ci-dessous



```
import React,{Component} from "react";
export default class Voiture extends Component{
  constructor() {
    super()
    this.myVoiture={matricule:'19873X9', marque:'Toyota',chevaux:8}
  }
  render(){
    return(
      <div>
        <h1 style={{color:'green'}}>Information voiture</h1>
        <h4>voiture matricule:<span>-----</span></h4>
        <h5>marque:----- nombre de chevaux:-----</h5>
      </div>
    )
  }
}
```

Le matricule doit être affiché en rouge