

**Rappel :**

Pour utiliser le code JSX nous aurons besoin de charger *Bootstrap*, *React JS*, *Babel Browser* et spécifier le type *text/babel* pour les fichiers JSX :

```
<html>
<head>
<script src="https://unpkg.com/react@16/umd/react.development.js"
crossorigin></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/react-
dom/16.14.0/umd/react-dom.development.min.js" crossorigin></script>
<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
</head>
<body>
<div id="app"></div>
</body>
<script type="text/babel">
  // Code JSX
</script>
</html>
```

**Exercice 1:**

Afficher les éléments suivants en utilisant le langage JSX :

**Page1/index.html :**

**Modules du semestre 1**

1. Méthodes agiles
2. Gestion des données
3. Développement front-end

En utilisant le render :

ReactDOM.render(MaListe, document.getElementById("app"));

**Page2/index.html :**

**Emploi du temps**

Jours	08:00 à 10:00	10:00 à 12:00	14:00 à 16:00	16:00 à 18:00
Lundi	Math	Arabe	Français	S.V.T
Mardi			Math	Français
Mercredi	Arabe	Français		P.C
Jeudi			H.G	Math
Vendredi	E.I	Français		

En utilisant le render :

ReactDOM.render(MaTable, document.getElementById("app"));

**Exercice 2:**

A partir du tableau suivant :

```
var elements=[[ "Jours", "08:00 à 10:00", "10:00 à 12:00", "14:00 à 16:00", "16:00 à 18:00"],
               [ "Lundi", "Math", "Arabe", "Français", "S.V.T"],
               [ "Mardi", "", "", "Math", "Français"],
               [ "Mercredi", "Arabe", "Français", "", "S.V.T"],
               [ "Jeudi", "", "", "H.G", "Math"],
               [ "Vendredi", "E.I", "Français", "", "" ]
               ];
```

Afficher l'élément suivant à l'aide de JSX :

Jours	08:00 à 10:00	10:00 à 12:00	14:00 à 16:00	16:00 à 18:00
Lundi	Math	Arabe	Français	S.V.T
Mardi			Math	Français
Mercredi	Arabe	Français		S.V.T
Jeudi			H.G	Math
Vendredi	E.I	Français		

En utilisant le render :

```
ReactDOM.render(MaTable, document.getElementById("app"));
```

### Exercice 3:

Créer et afficher le composant MaTable suivant :

Jours	08:00 à 10:00	10:00 à 12:00	14:00 à 16:00	16:00 à 18:00
Lundi	Math	Arabe	Français	S.V.T
Mardi			Math	Français
Mercredi	Arabe	Français		S.V.T
Jeudi			H.G	Math
Vendredi	E.I	Français		

En utilisant une fonction et le tableau suivant :

```
var elements=[[ "Jours", "08:00 à 10:00", "10:00 à 12:00", "14:00 à 16:00", "16:00 à 18:00"],
               [ "Lundi", "Math", "Arabe", "Français", "S.V.T"],
               [ "Mardi", "", "", "Math", "Français"],
               [ "Mercredi", "Arabe", "Français", "", "S.V.T"],
               [ "Jeudi", "", "", "H.G", "Math"],
               [ "Vendredi", "E.I", "Français", "", "" ]
               ];
```

Et le render : `ReactDOM.render(<MaTable/>,`

`document.getElementById("app"));`

Puis en utilisant les props, appliquer le gras à toutes les cellules de la table :

Jours	08:00 à 10:00	10:00 à 12:00	14:00 à 16:00	16:00 à 18:00
Lundi	Math	Arabe	Français	S.V.T
Mardi			Math	Français
Mercredi	Arabe	Français		S.V.T
Jeudi			H.G	Math
Vendredi	E.I	Français		

En utilisant le render :

```
ReactDOM.render(<MaTable elements={elements}
style={{fontWeight:"bold"}}/>, document.getElementById("app"));
```

#### Exercice 4:

Créer maintenant un composant **Ligne** en utilisant une fonction avec deux propriétés : la ligne à parcourir dans le tableau (Exercice 3), et celle permettant de mettre les cellules de la ligne en gras. Le composant Ligne correspond à une ligne dans laquelle nous allons parcourir les cellules.

Créer ensuite le composant **MaTable** en utilisant une fonction et en se basant sur le composant **Ligne** permettant de parcourir les lignes de la table. Ce composant est caractérisé par les propriétés lignes (les lignes du tableau qui sont les éléments du tableau **elements**, la propriété **weight** permettant de mettre les cellules en gras et la propriété **border** permettant de définir l'épaisseur de la bordure de la table.

Afficher le composant MaTable en utilisant le render :

```
ReactDOM.render(<MaTable lignes={elements} weight="bold"
border="1"/>, document.getElementById("app"));
```

#### Exercice 5:

Créer maintenant un composant **Ligne** en utilisant une classe avec deux propriétés : la ligne à parcourir dans le tableau (Exercice 3), et celle permettant de mettre les cellules de la ligne en gras. Le composant Ligne correspond à une ligne dans laquelle nous allons parcourir les cellules.

Créer ensuite le composant **MaTable** en utilisant une classe et en se basant sur le composant **Ligne** permettant de parcourir les lignes de la table. Ce composant est caractérisé par les propriétés lignes (les lignes du tableau qui sont les éléments du tableau **elements**, la propriété **weight** permettant de mettre les cellules en gras et la propriété **border** permettant de définir l'épaisseur de la bordure de la table.

Afficher le composant MaTable en utilisant le render :

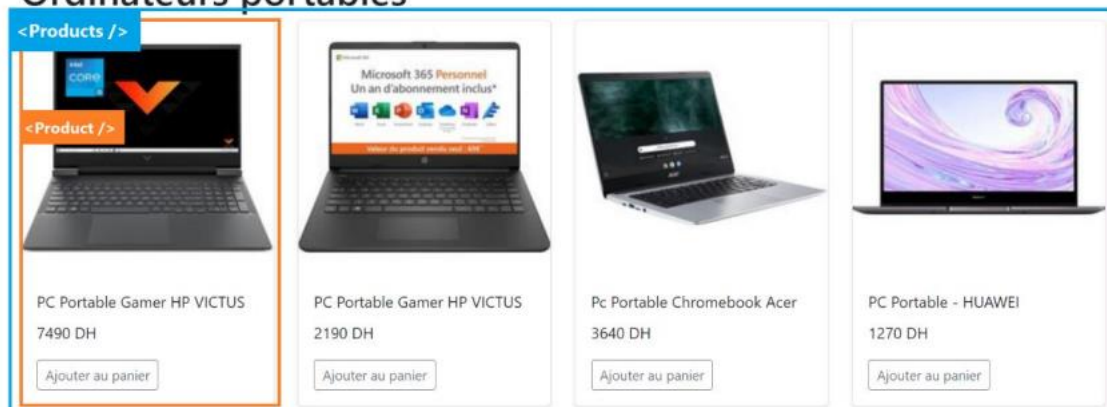
```
ReactDOM.render(<MaTable lignes={elements} weight="bold"
border="1"/>, document.getElementById("app"));
```

#### Exercice 6:

**Pensez en composants :**

Avec React, chacune des parties de site qu'on va pouvoir réutiliser correspond à un composant.

## Ordinateurs portables



Les composants nous permettent d'utiliser la même structure de données, et de remplir ces structures avec différents jeux de données. Peu importe le nombre de produits que vous aurez à mettre dans le site, vous pourrez les exploiter pour afficher vos données sans aucun effort.

### Construire vos composants :

1. Créer un composant **<Product/>** qui permet d'afficher un produit :

```
<div className="col">
  <div className="card shadow-sm">
    
    <div className="card-body">
      <p className="card-title">Titre de produit</p>
      <p className="card-text">Prix de produit</p>
      <div className="d-flex justify-content-between align-items-
center">
        <div className="btn-group">
          <button type="button" className="btn btn-sm btnoutline-
secondary">Ajouter au panier</button>
        </div>
      </div>
    </div>
  </div>
</div>
```

2. Créer un composant **<Products/>** qui permet d'afficher tous les produits. Soit la variable `products` qui contient les informations des produits :

```
const products = [
  {
    id: 1,
    title: 'PC Portable Gamer HP VICTUS',
    price: '7490 DH',
    thumbnail : 'HP16D0195NF.jpg'
  },
  {
    id: 2,
    title: 'PC Portable Gamer HP VICTUS',
    price: '2190 DH',
    thumbnail : 'HP14424U3EA.jpg'
  },
  {
    id: 3,
    title: 'Pc Portable Chromebook Acer',
    price: '3640 DH',
    thumbnail: 'NXATHEF002.jpg'
  },
  {
    id: 4,
    title: 'PC Portable - HUAWEI',
    price: '1270 DH',
    thumbnail: 'HUA6901443442959.jpg'
  }
];
```

3. Utiliser le code suivant pour parcourir les produits :

```
{products.map((product) => (  
  <Product key={product.id} title={product.title}  
  price={product.price} thumbnail={product.thumbnail} />  
))}
```

4. Rajouter le composant **<Products/>** dans le render :

```
ReactDOM.render(<Products/>, document.getElementById("app"));
```

Utiliser les images des ordinateurs dans le répertoire qui accompagne ce TP.