Cheat Sheet

Create React App:

```
npx create-react-app my-app
cd my-app
npm start
```

Routing in React:

```
import React from 'react';
import './App.css';
import Header from './components/Header/Header';
import Shop from './components/Shop/Shop';
import { Routes, Route, Link, BrowserRouter } from "react-router-dom";
import Review from './components/Review/Review';
import Inventory from './components/Inventory/Inventory';
import NotFound from './components/NotFound/NotFound';
import ProductDetail from './components/ProductDetail/ProductDetail';
import Info from './components/Info/Info';

function App() {
  return (
    <div>
      <Header></Header>
      <BrowserRouter>
        <Routes>
          <Route path="/" element={<Shop></Shop>}></Route>
          <Route path="/shop" element={<Shop></Shop>}></Route>
          <Route path="/review" element={<Review></Review>}></Route>
          <Route path = "/inventory" element = {<Inventory></Inventory>}></Route>
          <Route path = "/info" element = {<Info></Info>}></Route>


          <Route path = "/product/:productKey" element = {<ProductDetail></ProductDetail>}></Route>

          <Route path="*" element={<NotFound></NotFound>}></Route>

        </Routes>
      </BrowserRouter>



    </div>
  );
}

export default App;
```

- var and let are block scope.

For loop in Javascript:

```
for (let i = 0; i < 5; i++) {
  text += "The number is " + i + "<br>";
}
```

Array function in Javascript:

```
const formatNumber = (num) => {
    const precision = num.toFixed(2);
    return Number(precision);
}
```

Button hover:

```
.mainButton:hover{
    background-color: darkgoldenrod
}
```

Form:

```
<form onSubmit={handleSubmit}>
    <input type="text" name="Name" onBlur={handleBlur} placeholder="Name" required />
    <br />
    <input type="text" name="Email" onBlur={handleBlur} placeholder="Email" required />
    <br />
    <input style={{ width: "500px" }} name="Address" type="text" onBlur={handleBlur} placeholder="Address" required />
    <br />
    <br />
    <input type="submit" value="Submit" />
</form>
```

Conditional Execution:

```
{
    orderSuccess && <h4>Order Successful!</h4>

}
```

Use useEffect to perform side effects:

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

function Example() {
  const [count, setCount] = useState(0);

  // Similar to componentDidMount and componentDidUpdate:
  useEffect(() => {
    // Update the document title using the browser API
    document.title = `You clicked ${count} times`;
  });

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

Use style within a tag:

```
<div style={{ marginLeft: "100px" }}>
```

Use of useState hook:

```
const [cart, setCart] = useState([]);
const [userInfo, setUserInfo] = useState([]);
const [orderSuccess, setOrderSuccess] = useState(false);
```

How to get data from props:

```
const { img, name, seller, price, stock, key } = props.product;
```

useParams to get the dynamic value of url:

```
const { productKey } = useParams();
```

Array.find():

```
const array1 = [5, 12, 8, 130, 44];

const found = array1.find(element => element > 10);

console.log(found);
// expected output: 12

```

Children of a component:

```
<Cart cart={cart}>
    <Link to = "/info">
    <button className="mainButton">Place Order</button>
    </Link>
</Cart>
```

Array.slice():

```
const animals = ['ant', 'bison', 'camel', 'duck', 'elephant'];

console.log(animals.slice(2));
// expected output: Array ["camel", "duck", "elephant"]

console.log(animals.slice(2, 4));
// expected output: Array ["camel", "duck"]
```

Passing data as props:

```
products.map(product => <Product
    key={product.key}
    showAddToCart={true}
    handleAddProduct={handleAddProduct}
    product={product}>
</Product>)
```

Array.reduce():

```
const array1 = [1, 2, 3, 4];
const reducer = (previousValue, currentValue) => previousValue + currentValue;

// 1 + 2 + 3 + 4
console.log(array1.reduce(reducer));
// expected output: 10

// 5 + 1 + 2 + 3 + 4
console.log(array1.reduce(reducer, 5));
// expected output: 15
```

How to use <nav></nav> tag:

```
<nav>
    <a href="/shop">Shop</a>
    <a href="/review">Order Review</a>
    <a href="/inventory">Manage Inventory</a>
</nav>
```

- **e.preventDefault() to stop reloading of page while submitting a form**

Destructuring array of objects:

```
const handleBlur = (e) => {
    const newInfo = { name: e.target.name, value: e.target.value };
    const addedUserInfo = [...userInfo, newInfo];
    setUserInfo(addedUserInfo);
}
```