React Js

By Codek Academy

What is React?

React is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications.

Why Learn React?

- Popularity and Community: React is one of the most popular JavaScript libraries, with a large community and extensive documentation.
- Reusable Components: React's component-based architecture allows for reusable components, making your code more maintainable and scalable.
- Performance: React is known for its high performance due to the virtual DOM.
- Versatility: You can use React for web applications, mobile applications (with React Native), and even for building desktop applications.

Getting Started

- To get started with React, you need to have Node.js and npm (Node Package Manager) installed on your computer. You can download them from nodejs.org.

- You can create a new React application using the Create React App CLI tool. Run the following command in your terminal:

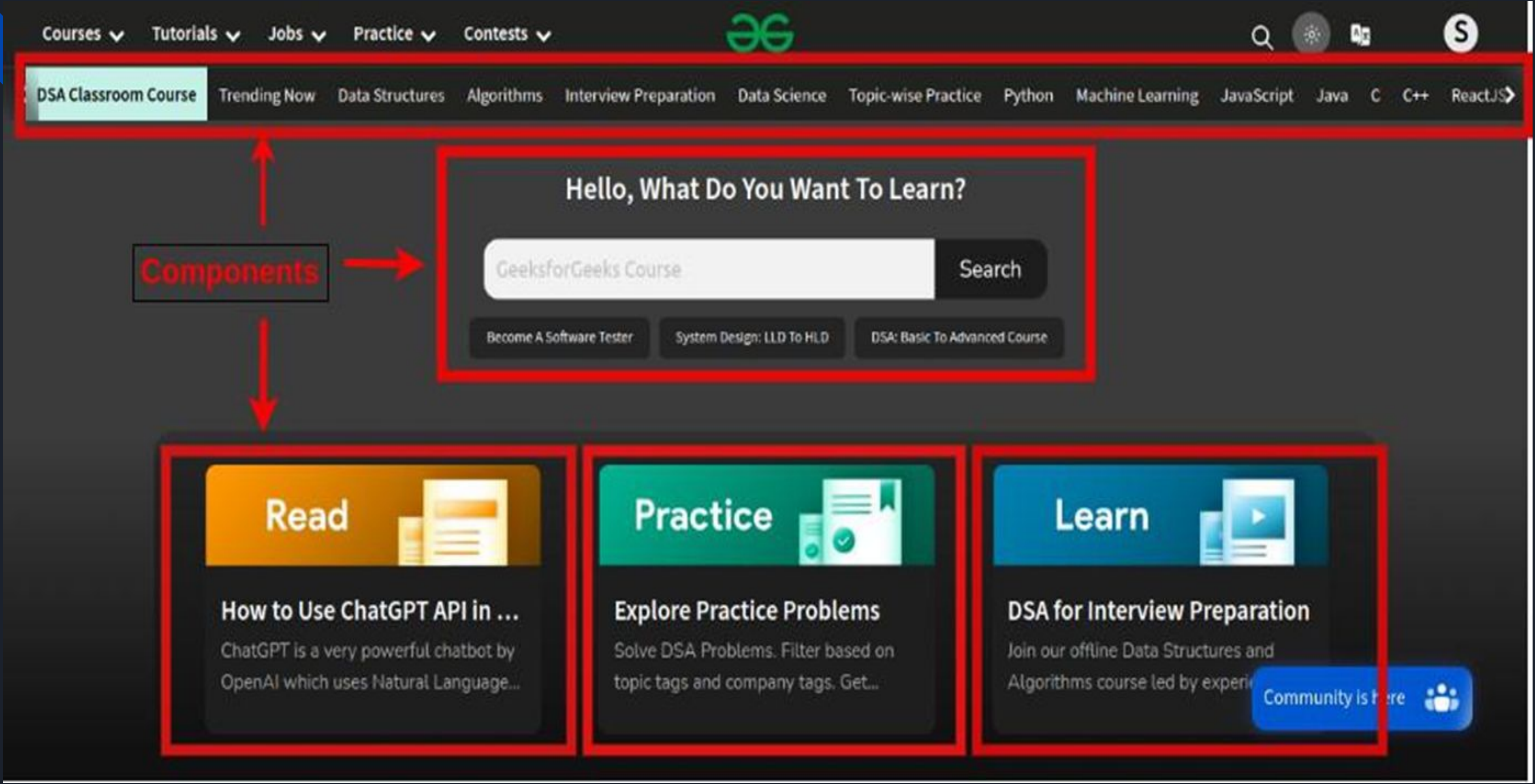
```
npm create vite@latest my-app

cd my-app

npm install

npm run dev
```

Components

What are components?

Components are the fundamental building blocks in React. They are reusable pieces of code that can be used to build elements on the page

- ❖ Reusable piece of code that can be used to build elements on the page
- ❖ Components can get props passed in and can hold own state
- ❖ Allows us to break down complex Uis, which make them easier to maintain and scale
- ❖ Components can be nested within other components, allowing you to build hierarchical and well-structured UIs.



Courses
Tutorials
Jobs
Practice
Contests
DSA Classroom Course
Trending Now
Data Structures
Algorithms
Interview Preparation
Data Science
Topic-wise Practice
Python
Machine Learning
JavaScript
Java
C
C++
ReactJS
Components
Hello, What Do You Want To Learn?
GeeksforGeeks Course
Search
Become A Software Tester
System Design: LLD To HLD
DSA: Basic To Advanced Course
Read
How to Use ChatGPT API in ...
ChatGPT is a very powerful chatbot by OpenAI which uses Natural Language...
Practice
Explore Practice Problems
Solve DSA Problems. Filter based on topic tags and company tags. Get...
Learn
DSA for Interview Preparation
Join our offline Data Structures and Algorithms course led by experi
Community is here

```jsx
// App.jsx ;

function App() {

  return (
   <>
     <Header name="Omar" />
   </>
  )
}

 export default App
```

```jsx
// Header.jsx ;

const Header = ({name}) => {

  return (
   <div>
     <h1>Hello {name}</h1>
   </div>
  );
};

 export default Header;
```

State

What is State

- State is specific to a component and not shared directly with other components. This promotes isolation and prevents unintended side effects.
- This could be form input data, fetched data, UI-related data like if a modal is open/close
- State allows you to create dynamic, data-driven UIs

useState

```
import React, { useState } from "react";

 const Counter = () => {

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

   const handleClick = () => {
    setCount(count + 1);
  };

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

 export default Counter;
```

onChange

```
import React, { useState } from "react";

 const NameInput = () => {
  const [name, setName] = useState(""); // Initialize state with an empty string

   const handleChange = (event) => {
    setName(event.target.value); // Update state with the input value
  };

   return (
    <div>
      <label>Enter your name:</label>
      <input
        type="text"
        value={name} // Set the input value to the current state
        onChange={handleChange} // Call handleChange on input change
      />
      <p>Hello, {name}!</p>
    </div>
  );
};

 export default NameInput;
```

Toggle state

```
import React, { useState } from "react";

 const Toggle = () => {
  const [isOn, setIsOn] = useState(false);

   const handleClick = () => {
    setIsOn(!isOn);
  };

   return (
    <button
      style={{ backgroundColor: isOn ? "green" : "red" }}
      onClick={handleClick}
    >
      {isOn ? "ON" : "OFF"}
    </button>
  );
};

 export default Toggle;
```

Let’s build our first app

App.jsx

```jsx
function App() {
  const [books, setBooks] = useState([
    { id: 1, title: "Clean Code", author: "Robert Cecil Martin" },
    { id: 2, title: "Design Patterns", author: "Erich Gamma" },
  ]);

   return (
    <>
      <Book books={books} />
    </>
  );
}
```

Book.jsx

```jsx
const Book = ({ books }) => {
  return (
    <div>
      {books.map((book) => (
        <div key={book.id}>
          <h1>{book.title}</h1>
          <h4>{book.author}</h4>
        </div>
      ))}
    </div>
  );
};

 export default Book;
```

Add title props

```
<Book books={books} title="All Books Here" />
```

Reusing Component

```
function App() {
  const [books, setBooks] = useState([
     { id: 1, title: "Clean Code", author: "Robert Cecil Martin" },
     { id: 2, title: "Design Patterns", author: "Erich Gamma" },
     { id: 3, title: "Contributing To Eclipse", author: "Erich Gamma" },
     { id: 4, title: "Secrets of the JavaScript Ninja", author: "John
Resig" },
  ]);

   return (
     <div className="App">
     <Book books={books} title="All Books Here" />
     <Book books={books.filter((book) => book.author == "Erich Gamma")}
     title="Erich Gamma Books Here"
     />
     </div>
  );
}
export default App;
```

Function props

Delete function

```
import Book from "./components/Book";

function App() {

  const [books, setBooks] = useState([
      { id: 1, title: "Clean Code", author: "Robert Cecil Martin" },
      { id: 2, title: "Design Patterns", author: "Erich Gamma" },
      { id: 3, title: "Contributing To Eclipse", author: "Erich Gamma" },
      { id: 4, title: "Secrets of the JavaScript Ninja", author: "John Resig"
},]);

  const handleDelete = (id) => {
      const newBooks = books.filter((book) => book.id !== id);
      setBooks(newBooks);};

  return (
      <div className="App">
      <Book books={books} title="All Books Here" handleDelete={handleDelete} />
      </div>
  );}

export default App;
```

```
// Book.jsx component


function Book({handleDelete}) {
  const books = props.books;
  return (
     <div>
     <h1>{props.title}</h1>
     {books.map((book) => (
     <div key={book.id}>
          <h2>{book.title}</h2>
          <h4>{book.author}</h4>
          <button onClick={() =>
handleDelete(book.id)}>Delete</button>
     </div>))}
     </div>
)}
export default Book;
```

Add function

```
 // in App.jsx we added addBook function and the UI to add a new Book :
const [title, setTitle] = useState("");
const [author, setAuthor] = useState("");

const addBook = () => {
  const newBook = {
    id: books.length + 1,
    title: title,
    author: author,
  };
  setBooks([...books, newBook]);
};
return (
  <>
    <div>
      <h2>Add a New Book</h2>
      <label>Title:</label>
      <input type="text" value={title} onChange={(e) => setTitle(e.target.value)}/>
      <label>Author:</label>
      <input type="text" value={author} onChange={(e) => setAuthor(e.target.value)}/>
      <button onClick={addBook}>Add Book</button>
    </div>
    <Book books={books} title="All Books Here" handleDelete={handleDelete} />
  </>
);
```

Update function

```jsx
// App.jsx

  const [Id, setId] = useState("");
  const [error, setError] = useState("");
  const [isEdit, setIsEdit] = useState(false);

  const handleEditClick = (book) => {
    setIsEdit(true);
    setTitle(book.title);
    setAuthor(book.author);
    setId(book.id);
  };

  const handleSave = () => {
    const updatedBook = {
      id: Id,
      title: title,
      author: author,
    };

    const updatedBooks = books.map((book) =>
      book.id === Id ? updatedBook : book
    );
    setBooks(updatedBooks);
  };
```

```
return (
  <>
    <div>
      {isEdit ? (
        <>
          <h2>Update a Book</h2>
          <label>Title:</label>
          <input
            type="text"
            value={title}
            onChange={(e) => setTitle(e.target.value)}
          />
          <label>Author:</label>
          <input
            type="text"
            value={author}
            onChange={(e) => setAuthor(e.target.value)}
          />
          <button onClick={() => handleSave()}>Save</button>
        </>
      ) :
```

```
(
          <div>
            <h2>Add a New Book</h2>
            <label>Title:</label>
            <input
              type="text"
              value={title}
              onChange={(e) => setTitle(e.target.value)}
            />
            <label>Author:</label>
            <input
              type="text"
              value={author}
              onChange={(e) => setAuthor(e.target.value)}
            />
            <button onClick={addBook}>Add Book</button>
            <p style={{ color: "red" }}>{error}</p>
          </div>
        )}
      </div>

      <Book
        books={books}
        title="All Books Here"
        handleDelete={handleDelete}
        handleEditClick={handleEditClick}
        isEdit={isEdit}
      />
    </>
  );
```

```
const Book = ({ books, title, handleDelete, handleEditClick }) => {

  return (
    <div className="bookContainer">
      <h1 className="bookTitle">{title}</h1>
      <ul className="bookList">
        {books.map((book) => (
          <li key={book.id} className="bookItem">
            <h2 className="bookItemTitle">{book.title}</h2>
            <h4 className="bookItemAuthor">{book.author}</h4>
            <button onClick={() => handleDelete(book.id)}>Delete</button>
            <button onClick={() => handleEditClick(book)}>Edit</button>
          </li>
        ))}
      </ul>
    </div>
  );
};

export default Book;
```

Higher Order Function

```
// map() :
```

- The map() method creates a new array by applying a function to each element of the original array.
- The map() method loop through each item it the array and every single time it does that it returns a value
- map() does not change the original array
- `array.map((currentValue, index)=>{... return value })`

```
// Exemple 1

const numbers = [1, 2, 3, 4, 5];

const doubledNumbers = numbers.map((num) => {
  return (
      num * 2
   );
}); // Double each number in the array
console.log(doubledNumbers); // Output: [2, 4, 6, 8, 10]

// Simplified
const doubledNumbers = numbers.map(num =>  num * 2);
```

```javascript
// Convert an Array of Strings to Uppercase

const words = ['hello', 'world', 'javascript', 'map'];

const uppercasedWords = words.map(word => word.toUpperCase());
// Output: ['HELLO', 'WORLD', 'JAVASCRIPT', 'MAP']


// Extract a Specific Property from an Array of Objects

const people = [
  { name: "Alice", age: 25 },
  { name: "Bob", age: 30 },
  { name: "Charlie", age: 35 },
];

const names = people.map((person) => person.name);
console.log(names); // Output: ['Alice', 'Bob', 'Charlie']
```

// Book.js

```
  const [books, setBooks] = useState([
    { id: 1, title: "Clean Code", author: "Robert Cecil Martin" },
    { id: 2, title: "Design Patterns", author: "Erich Gamma" },
    { id: 3, title: "Contributing To Eclipse", author: "Erich Gamma" },
    { id: 4, title: "Secrets of the JavaScript Ninja", author: "John Resig" },
  ]);


{books.map((book) => (
  <li key={book.id} className="bookItem">
   <h2 className="bookItemTitle">{book.title}</h2>
   <h4 className="bookItemAuthor">{book.author}</h4>
   <button onClick={() => handleDelete(book.id)}>Delete</button>
   <button onClick={() => handleEditClick(book)}>Edit</button>
  </li>
))}
```

```
const handleSave = () => {
  const updatedBook = {
    id: Id,
    title: title,
    author: author,
  };

  const updatedBooks = books.map((book) =>
    book.id === Id ? updatedBook : book
  );
  setBooks(updatedBooks);
};
```

// filter()

- the filter method will filter out the elements of an array based on the specified test condition
- The filter() method creates a new array filled with elements that pass a test provided by a function.
- The filter() method does not change the original array.
- `array.map((currentValue, index)=>{... return value })`

```
const words = ["Avocado", "Banana", "Apple", "Mango", "Kiwi", "Orange"];

const result = words.filter((word) => {
  return (
          word.length > 5
       )
});

console.log(result);

// Clean

const words = ["Avocado", "Banana", "Apple", "Mango", "Kiwi", "Orange"];

const result = words.filter(word => word.length > 5);

console.log(result);
```

```javascript
const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

const evenNumbers = numbers.filter((number) => number % 2 === 0);

console.log(evenNumbers); // Output: [2, 4, 6, 8, 10]

// Object filter

const people = [
  { name: "Alice", age: 25 },
  { name: "Bob", age: 17 },
  { name: "Charlie", age: 19 },
];

const adults = people.filter((person) => person.age >= 18);

console.log(adults); // Output: [{ name: 'Alice', age: 25 }, { name: 'Charlie', age:
19 }]
```

```javascript
const products = [
  { name: 'Laptop', category: 'Electronics' },
  { name: 'Shirt', category: 'Clothing' },
  { name: 'Phone', category: 'Electronics' }
];

const electronics = products.filter(product => product.category === 'Electronics');

console.log(electronics); // Output: [{ name: 'Laptop', category: 'Electronics' }, { name:
'Phone', category: 'Electronics' }]
```

```
const [books, setBooks] = useState([
  { id: 1, title: "Clean Code", author: "Robert Cecil Martin" },
  { id: 2, title: "Design Patterns", author: "Erich Gamma" },
  { id: 3, title: "Contributing To Eclipse", author: "Erich Gamma" },
  { id: 4, title: "Secrets of the JavaScript Ninja", author: "John Resig" },
]);


const handleDelete = (id) => {
  const newBooks = books.filter((book) => book.id !== id);
  setBooks(newBooks);
};
```

Conditional Rendering

```
function Greeting() {
  const isLoggedIn = true;
  return (
    <div>{isLoggedIn ? <h1>Welcome back!</h1> : <h1>Please sign up.</h1>}</div>
  );
}

export default Greeting;
```

```
function Mailbox() {
  const unreadMessages = 5;
  return (
    <div>
      {unreadMessages > 0 && (
        <h2>You have {unreadMessages} unread messages.</h2>
      )}
    </div>
  );
}

export default Mailbox;
```

```jsx
{isEdit ? (
    <>
      <h2>Update a Book</h2>
      <label>Title:</label>
      <input
        type="text"
        value={title}
        onChange={(e) => setTitle(e.target.value)}
      />
      <label>Author:</label>
      <input
        type="text"
        value={author}
        onChange={(e) => setAuthor(e.target.value)}
      />
      <button onClick={() => handleSave()}>Save</button>
    </>
  ) : (
    <div>
      <h2>Add a New Book</h2>
      <label>Title:</label>
      <input
        type="text"
        value={title}
        onChange={(e) => setTitle(e.target.value)}
      />
      <label>Author:</label>
      <input
        type="text"
        value={author}
        onChange={(e) => setAuthor(e.target.value)}
      />
      <button onClick={addBook}>Add Book</button>
      <p style={{ color: "red" }}>{error}</p>
    </div>
  )}
```

useEffect

```
import { useState, useEffect } from "react";

 function App() {
  const [count, setCount] = useState(0);
  const increment = () => {
    setCount(count + 1);
  };
  const decrement = () => {
    setCount(count - 1);
  };

   useEffect(() => {
    console.log("use effect run");
  });

   return (
    <div className="App">
      <div>
        <h2>Counter: {count}</h2>
        <button onClick={increment}>Increment</button>
        <button onClick={decrement}>Decrement</button>
      </div>
    </div>
  );
}

 export default App;
```

```
import { useState, useEffect } from "react";

const UseEffectHook = () => {
  const [count, setCount] = useState(0);
  const [name, setName] = useState("");

  const increment = () => {
    setCount(count + 1);
  };
  const decrement = () => {
    setCount(count - 1);
  };

  const handleChange = (event) => {
    setName(event.target.value);
  };

  useEffect(() => {
    console.log("use effect run");
  }, [count]);

  return (
    <div className="App">
      <div>
        <h2>Counter: {count}</h2>
        <button onClick={increment}>Increment</button>
        <button onClick={decrement}>Decrement</button>
      </div>
      <label>Enter your name:</label>
      <input type="text" value={name} onChange={handleChange} />
      <p>Hello, {name}!</p>
    </div>
  );
};

export default UseEffectHook;
```

```
useEffect(() => {
  console.log("use effect run");
}, []);
useEffect(() => {
  console.log("count change");
}, [count]);
```

Primitives vs References

```javascript
// Primitives : number, String, Boolean, Null, Undefined

const x = 5;
const y = 5;

console.log(x === y); // output : true

const a = "React";
const b = "React";

console.log(a === b); // output : true

// Reference : Array, Object, Function

const a = [1, 2, 3];
const b = [1, 2, 3];

console.log(a === b); // output : false

const a = { value: 5 };
const b = { value: 5 };

console.log(a === b); // output : false
```

```jsx
import { useState, useEffect } from "react";
function App() {
  const [count, setCount] = useState(0);

  const a = 1;
  const b = [1,2,3];

  useEffect(() => {
    console.log("The useEffect hook runs when 'a' change occurs");
  }, [a]);

 useEffect(() => {
    console.log("The useEffect hook runs when 'b' change occurs");
  }, [b]);

  return (
    <div>
      <div>
        <p>
          Count: <span id="count">{count}</span>
        </p>

        <button onClick={() => setCount(count + 1)}>Increment</button>
      </div>
    </div>
  );
}
export default App;
```

axios

https://freetestapi.com/apis/books

/books	Get	Fetch all books
/books/{id}	Get	Fetch a single book
/books	Post	Add a new book
/books/{id}	Delete	Delete a book

```
const Book = () => {
  const [books, setBooks] = useState([]);

  useEffect(() => {
    axios
      .get("https://freetestapi.com/api/v1/books")
      .then(({ data }) => {
        console.log(data);
      })
      .catch((error) => {
        console.log(error);
      });
  }, []);

  useEffect(() => {
    async function getBooks() {
      const response = await axios.get("https://freetestapi.com/api/v1/books");
      console.log(response.data);
      setBooks(response.data);
    }
    getBooks();
  }, []);
```

Thanks for your attention

Don’t forget to follow us please

Click here : [Codek Academy]