

React.js

A JavaScript library to build user interfaces.

React does that combining small amount of HTML, CSS, JS as a single component.
each components have the own styling and functionality.

e.g. html. JSX.

```
<body>  
  <div id='root'></div>  
</body>
```

index.js

```
import React from 'react'  
import ReactDOM from 'react-dom'  
  
ReactDOM.render(<h1>Hello</h1>, document.getElementById('root'))
```

→ Hello

(render will take minimum 2 inputs first one what to show,
second where to show)

If want you give more than one element wrap them using `<div>` `</div>`

```
const name = 'dhinesh';           → code for the current year.  
const year = new Date.getFullYear();
```

```
ReactDOM.render(<div>  
  <p>created by {name} </p>  
  <p>copyright {year} </p>  
  </div>,  
  document.getElementById('root'))
```

we can use values using the curly brackets.

bacon
created by dhinesh
copyright 2023

JSX attributes & styling react elements.

```
import React from 'react';
import ReactDOM from 'react-dom';

ReactDOM.render(
  <div>
    <h1> Foods </h1>
    <ul>
      <li> bacon </li>
      <li> noodle </li>
    </ul>
  </div> ,
```

```
document.getElementById('root');
```

index.js
ReactDOM.render(<h1 className='heading'> hello </h1> ,
document.getElementById('root'));

styles.css

.heading { color: red; }

(The codes print h1 tag in red color, Note that in normal
html we use class but here className in javascript.)

→ <h1, className='heading' contentEditable='True'>
hello </h1>

(using the contentEditable JSX attribute we can
change and write text in the interface in website)

JSX attributes (for more JSX attributes refer html global attribute
in W3 Schools doc).

inline styling in JSX.

```
const customStyle = [
  color: 'red',
  fontSize: '20px',
  border: '1px solid black'
};

customStyle.color = 'blue';

ReactDOM.render(
  <h1 style={customStyle}>Hello </h1>,
  document.getElementById('root')
);
```

React Components.

Heading.jsx

```
import React from 'react';

function Heading() {
  return <h1> display </h1>;
}

export default Heading;
```

index.js

```
import Heading from './Heading.jsx'
```

```
ReactDOM.render(
```

```
<Heading></Heading>,
```

```
document.getElementById('root'));
```

display

using react components we can write reusable codes.

the extension is
optional ↑

React Properties

```
function Wish(props) { return <h1>hello {props.name} </h1> }
```

```
ReactDOM.render(<Wish name="dhinu"></Wish>,  
document.getElementById('root'));
```

React dev tools.

contacts.js

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

```
const contacts = [ {id: 1, name: 'dhinesh', phone: '7339'},  
                  {id: 2, name: 'kumar', phone: '4752'} ]
```

```
export default contacts;
```

card.jsx

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

```
function Card(props) {
```

```
    return <div>
```

```
        <h1>{props.name}</h1>
```

```
        <h1>{props.phone}</h1>
```

```
    </div>;
```

```
}
```

```
export default Card;
```

APP.jsx.

```
import React from 'react';
import Card from './Card';
import Contacts from '../contacts';

function createCard(contacts) {
    return <Card
        name={contacts.name}
        phone={contacts.phone}>
        </Card>;
}

function App() {
    return <div>
        {Contacts.map(createCard)}
        </div>;
}

export default App;
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './Components/App';

ReactDOM.render(<App/>, document.getElementById('root'));
```

(These code perfectly working fine, but it produce the following warning in console.

Warning: Each child in a list should have a unique 'key' prop).

So we need to set unique value. in our case we have the unique id.

So we can change like the following,

APP.jsx

```
function CreateCard(contact) {
  return <card key={contact.id}>
    name={contact.name}
    phone={contact.phone} </card>;
}
```