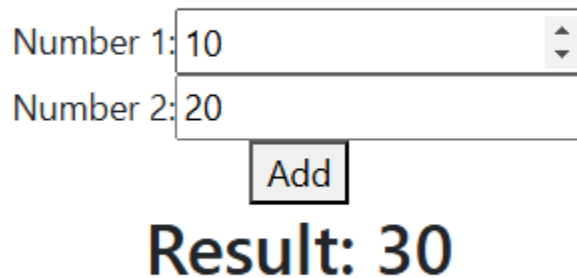


## Addition of 2 Number in React Js :-

# Add Two Numbers



Number 1: 10

Number 2: 20

Add

**Result: 30**

To create a simple React form that adds two numbers, you can follow the steps below.

### Step 1: Set up a React App

If you don't have a React app set up, you can create a new one using:

```
Cmd (terminal):-  
npx create-react-app add-numbers  
cd add-numbers  
npm start
```

### Step 2: Create a Form to Add Two Numbers

Here is a simple React component that provides a form to add two numbers:

```
Jsx code  
import React, { useState } from 'react';  
  
function AddNumbers() {  
  const [num1, setNum1] = useState('');  
  const [num2, setNum2] = useState('');  
  const [result, setResult] = useState(null);  
  
  const handleSubmit = (e) => {  
    e.preventDefault();  
    // Parse numbers and add them  
    const number1 = parseFloat(num1);  
    const number2 = parseFloat(num2);
```

```

    if (!isNaN(number1) && !isNaN(number2)) {
      setResult(number1 + number2);
    } else {
      alert("Please enter valid numbers.");
    }
  };

return (
  <div>
    <h1>Add Two Numbers</h1>
    <form onSubmit={handleSubmit}>
      <div>
        <label>
          Number 1:
          <input
            type="number"
            value={num1}
            onChange={(e) => setNum1(e.target.value)}
            placeholder="Enter first number"
          />
        </label>
      </div>
      <div>
        <label>
          Number 2:
          <input
            type="number"
            value={num2}
            onChange={(e) => setNum2(e.target.value)}
            placeholder="Enter second number"
          />
        </label>
      </div>
      <button type="submit">Add</button>
    </form>
    {result !== null && (
      <div>
        <h2>Result: {result}</h2>
      </div>
    )}
  </div>
);
}

export default AddNumbers;

```

### Step 3: Use the `AddNumbers` Component in `App.js`

Replace the content of `src/App.js` with:

#### Jsx code

```
import React from 'react';
import AddNumbers from './AddNumbers'; // Import the AddNumbers component

function App() {
  return (
    <div className="App">
      <AddNumbers />
    </div>
  );
}

export default App;
```

#### Explanation:

##### 1. State Management:

- We use `useState` to store the values of `num1`, `num2`, and the `result`.

##### 2. Form Handling:

- When the form is submitted, the `handleSubmit` function parses the input values to floats and adds them.

##### 3. Rendering:

- If the result is available (i.e., it's not `null`), it is displayed under the form.

### Step 4: Styling (Optional)

You can style your form using simple CSS. Here's an example of how you could style the form in `App.css`:

#### css code:-

```
div {
  font-family: Arial, sans-serif;
  text-align: center;
  padding: 20px;
}

input {
  padding: 5px;
  margin: 10px;
}

button {
  padding: 5px 10px;
  background-color: #4CAF50;
  color: white;
  border: none;
  cursor: pointer;
}
```

```
button:hover {  
  background-color: #45a049;  
}  
  
h2 {  
  color: blue;  
}
```

## Step 5: Run the Application

Now, you should be able to run the application with:

```
cmd (terminal):-  
npm start
```

This will launch the React app in your browser, where you can input two numbers and see their sum after submitting the form.