React Guide

Input Fields in React (Controlled Components)

In React, input fields should be **controlled components**, meaning React manages their state. Here's how to properly handle inputs:

Basic Text Input

```
import { useState } from 'react';

function TextInput() {
  const [value, setValue] = useState('');

  const handleChange = (e) => {
    setValue(e.target.value);
  };

  return (
    <input
      type="text"
      value={value}
      onChange={handleChange}
      placeholder="Type something..."
    />
  );
}
```

Key Concepts

1. **Controlled Input** - React controls the input value via state
2. `value` **prop** - Ties input to React state
3. `onChange` **handler** - Updates state when user types

Common Input Types

1. Password Input

```
const [password, setPassword] = useState('');
```

```
<input
  type="password"
  value={password}
  onChange={(e) => setPassword(e.target.value)}
/>
```

2. Checkbox

jsx

Copy

Download

```
const [isChecked, setIsChecked] = useState(false);

<input
  type="checkbox"
  checked={isChecked}
  onChange={(e) => setIsChecked(e.target.checked)}
/>
```

3. Radio Buttons

```
const [selectedOption, setSelectedOption] = useState('option1');

<div>
  <input
    type="radio"
    value="option1"
    checked={selectedOption === 'option1'}
    onChange={(e) => setSelectedOption(e.target.value)}
  /> Option 1

  <input
    type="radio"
    value="option2"
    checked={selectedOption === 'option2'}
    onChange={(e) => setSelectedOption(e.target.value)}
  /> Option 2
</div>
```

4. Select Dropdown

```
const [selectedFruit, setSelectedFruit] = useState('apple');

<select
  value={selectedFruit}
  onChange={(e) => setSelectedFruit(e.target.value)}
>
  <option value="apple">Apple</option>
  <option value="banana">Banana</option>
  <option value="orange">Orange</option>
</select>
```

5. File Input

```
const [file, setFile] = useState(null);

<input
  type="file"
  onChange={(e) => setFile(e.target.files[0])}
/>
```

Form Submission

```
function MyForm() {
  const [name, setName] = useState('');

  const handleSubmit = (e) => {
    e.preventDefault();
    alert(`Submitted: ${name}`);
  };

  return (
    <form onSubmit={handleSubmit}>
      <input
        type="text"
        value={name}
        onChange={(e) => setName(e.target.value)}
      />
      <button type="submit">Submit</button>
    </form>
  );
}
```

Example 1

```
import React from "react";
import "./App.css";
import { useState } from 'react';

function TextInput() {
  const [value, setValue] = useState(0); //
Initialize with number 0

  const increment = () => {
    setValue(prevValue => prevValue + 1); //
Use functional update
  };

  const decrement = () => {
    setValue(prevValue => prevValue - 1); //
Use functional update
  };

  const handleChange = (e) => {
    const numValue = parseInt(e.target.value)
|| 0; // Convert to number
    setValue(numValue);
  };

  return (
    <>
```

```
      <input
        type="number"
        value={value}
        onChange={handleChange}
        placeholder="Enter a number"
      />
      <br />
      <button type="button"
onClick={increment}>Increment</button>
      <button type="button"
onClick={decrement}>Decrement</button>
    </>
  );
}

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

export default App;
```

Output

Example 2 Calculator

```
import React from "react";
import "./App.css";
import { useState } from 'react';

function Calculator() {
  const [number1, setNumber1] = useState(0);
  const [number2, setNumber2] = useState(0);
  const [result, setResult] = useState(0);

  const handleNumber1 = (e) => {
    const numValue = parseFloat(e.target.value)
|| 0;
    setNumber1(numValue);
  };

  const handleNumber2 = (e) => {
    const numValue = parseFloat(e.target.value)
|| 0;
    setNumber2(numValue);
  };
```

```
  const addition = () => {
    setResult(number1 + number2);
  };

  const subtraction = () => {
    setResult(number1 - number2);
  };

  const multiplication = () => {
    setResult(number1 * number2);
  };

  const division = () => {
    setResult(number2 !== 0 ? number1 / number2
: "Cannot divide by zero");
  };

  return (
    <div className="calculator">
      <h2>Simple Calculator</h2>
      <div className="input-group">
        <input
          type="number"
          value={number1}
          onChange={handleNumber1}
          placeholder="First number"
```

```
        />
        <br/><br/>
        <input
          type="number"
          value={number2}
          onChange={handleNumber2}
          placeholder="Second number"
        />
      </div>
      <div className="result">
        <p>Result: {result}</p>
      </div>
      <div className="buttons">
        <button type="button" onClick={addition}
style={{margin:20}}>+</button>
        <button type="button"
onClick={subtraction}style={{margin:20}}>-
</button>
        <button type="button"
onClick={multiplication}style={{margin:20}}>*</b
utton>
        <button type="button"
onClick={division}style={{margin:20}}>/</button>
      </div>
    </div>
  );
```

```
}

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

export default App;
```

Output