

# Two Way Binding

---

- two-way binding, which simply means that for inputs we don't just listen to changes, but we can also pass a new value back into the input. So that we can reset or change the input programmatically.
- Two-way binding is very useful when you're working with forms because it allows you to gather user input, but then also change it.

## How to do it?

1. We have to add a **value attribute**, which is default attribute in input element. It will set the internal value property, which every input element has.

```
return (  
  <form onSubmit={handleSubmit}>  
    <div className="new-expense__controls">  
      <div className="new-expense__control">  
        <label>Title</label>  
        <input type="text"  
          value={entitledTitle}  
          onChange={titleChangeHandler} />  
      </div>  
    </div>  
  </form>  
)
```

2. and then we can bind it to set new value. Here we are setting as empty bc we want to clear the inputs once form is submitted.

```
setChangedTitle(''),
```

## Whole program after binding.

```
import React, { useState } from "react";  
  
import "./ExpenseForm.css";  
  
const ExpenseForm = () => {  
  const [entitledTitle, setChangedTitle] = useState("");  
  const [entitledAmount, setChangedAmount] = useState("");  
  const [entitledDate, setChangedDate] = useState("");  
  
  const titleChangeHandler = (event) => {  
    setChangedTitle(event.target.value);  
  };  
  
  const amountChangeHandler = (event) => {  
    setChangedAmount(event.target.value);  
  };  
}
```

```
const dateChangeHandler = (event) => {
  setChangedDate(event.target.value);
};

const submitHandler = (event) => {
  event.preventDefault();

  const expenseData = {
    title: entitledTitle,
    amount: entitledAmount,
    date: new Date(entitledDate),
  };

  setChangedTitle(''),
  setChangedAmount(''),
  setChangedDate('')
};

return (
  <form onSubmit={submitHandler}>
    <div className="new-expense__controls">
      <div className="new-expense__control">
        <label>Title</label>
        <input type="text"
          value={entitledTitle}
          onChange={titleChangeHandler} />
      </div>
      <div className="new-expense__control">
        <label>Amount</label>
        <input
          type="number"
          min="0.01"
          step="0.01"
          value={entitledAmount}
          onChange={amountChangeHandler}
        />
      </div>
      <div className="new-expense__control">
        <label>Date</label>
        <input
          type="date"
          min="2019-01-01"
          max="2022-12-31"
          value={entitledDate}
          onChange={dateChangeHandler}
        />
      </div>
      <div className="new-expense__actions">
        <button type="submit">Add Expense</button>
      </div>
    </form>
  );
```

```
};  
  
export default ExpenseForm;
```