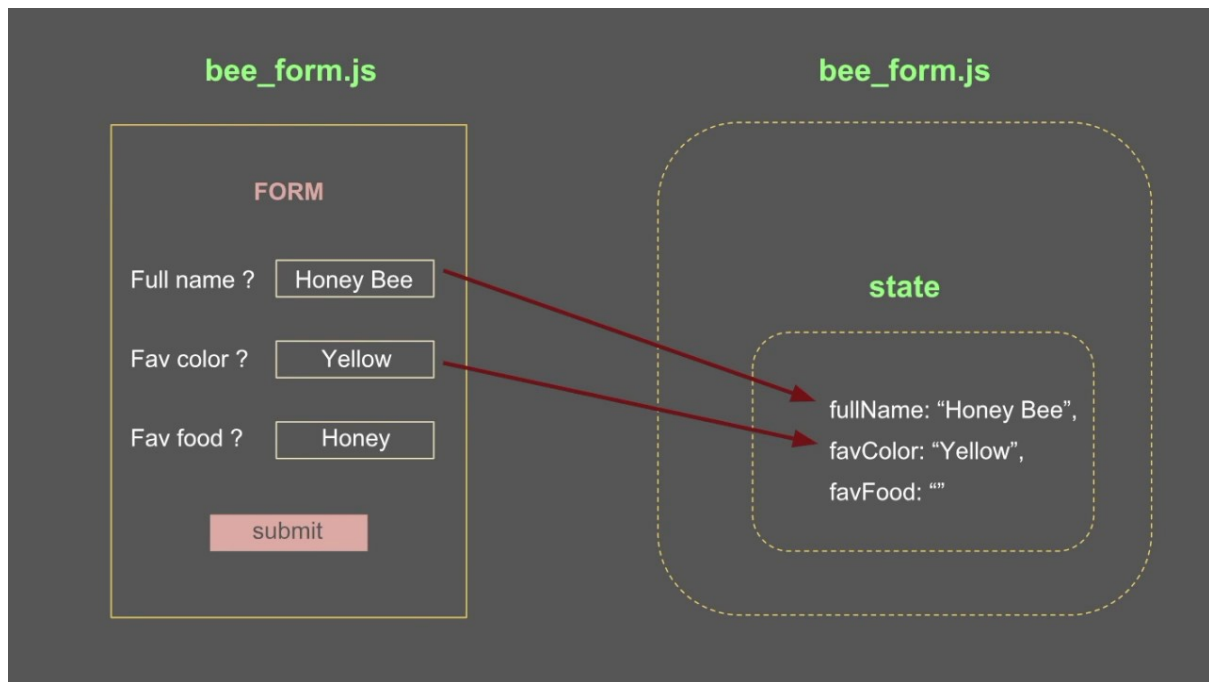


# Multiple Input Fields

---



- You can control the values of more than one input field by adding a name attribute to each element.
- We will initialize our state with an empty object.
- To access the fields in the event handler use the `event.target.name` and `event.target.value` syntax.
- To update the state, use square brackets [bracket notation] around the property name.

## Example:

- Write a form with **two input fields**:

```
import { useState } from "react";
import ReactDOM from "react-dom";

function MyForm() {
  const [inputs, setInputs] = useState({});

  const handleChange = (event) => {
    const name = event.target.name;
    const value = event.target.value;
    setInputs(values => ({...values, [name]: value}))
  }

  const handleSubmit = (event) => {
    event.preventDefault();
    alert(inputs);
  }

  return (
    <form onSubmit={handleSubmit}>
      <label>Enter your name:
      <input
        type="text"
        name="username"
        value={inputs.username || ""}
        onChange={handleChange}
      />
```

```
    </label>
    <label>Enter your age:
      <input
        type="number"
        name="age"
        value={inputs.age || ""}
        onChange={handleChange}
      />
    </label>
    <input type="submit" />
  </form>
)
}
```

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



localhost:3000

Enter your name:  Enter your age: