React useState Examples - Beginner to Advanced

Basic Counter

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

function Counter() {
  const [count, setCount] = useState(0);

  return (
    <div>
      <h1>Count: {count}</h1>
      <button onClick={() => setCount(count + 1)}>Increment</button>
      <button onClick={() => setCount(count - 1)}>Decrement</button>
      <button onClick={() => setCount(0)}>Reset</button>
    </div>
  );
}

export default Counter;
```

Input State

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

function InputExample() {
  const [text, setText] = useState('');
  return (
    <div>
      <input
        type="text"
        value={text}
        onChange={(e) => setText(e.target.value)}
      />
      <p>You typed: {text}</p>
    </div>
  );
}

export default InputExample;
```

Toggle State

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

function ToggleExample() {
  const [isOn, setIsOn] = useState(false);

  return (
    <div>
      <button onClick={() => setIsOn(!isOn)}>
        {isOn ? 'ON' : 'OFF'}
      </button>
    </div>
  );
}

export default ToggleExample;
```

Array State

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

function ArrayExample() {
  const [items, setItems] = useState([]);

  const addItem = () => {
    setItems([...items, `Item ${items.length + 1}`]);
  };

  return (
    <div>
      <button onClick={addItem}>Add Item</button>
      <ul>
        {items.map((item, index) => (
          <li key={index}>{item}</li>
        ))}
      </ul>
    </div>
  );
}

export default ArrayExample;
```

Object State

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

function ObjectExample() {
  const [user, setUser] = useState({ name: '', age: '' });

  return (
    <div>
      <input
        type="text"
        placeholder="Name"
        value={user.name}
        onChange={(e) => setUser({ ...user, name: e.target.value })}
      />
      <input
        type="number"
        placeholder="Age"
        value={user.age}
        onChange={(e) => setUser({ ...user, age: e.target.value })}
      />
      <p>Name: {user.name}</p>
      <p>Age: {user.age}</p>
    </div>

  );

}



export default ObjectExample;
```

Conditional Rendering

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



function ConditionalExample() {

  const [isLoggedIn, setIsLoggedIn] = useState(false);



  return (

    <div>

      {isLoggedIn ? (

        <p>Welcome, User!</p>

      ) : (

        <p>Please log in.</p>

      )}

      <button onClick={() => setIsLoggedIn(!isLoggedIn)}>

        {isLoggedIn ? 'Logout' : 'Login'}

      </button>

    </div>

  );

}
```

```
export default ConditionalExample;
```

State with Multiple Counters

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

function MultiCounter() {
  const [counters, setCounters] = useState([0, 0, 0]);

  const incrementCounter = (index) => {
    const newCounters = counters.map((count, i) =>
      i === index ? count + 1 : count
    );
    setCounters(newCounters);
  };

  return (
    <div>
      {counters.map((count, index) => (
        <div key={index}>
          <h2>Counter {index + 1}: {count}</h2>
          <button onClick={() => incrementCounter(index)}>Increment</button>
        </div>
      ))}
    </div>
  );
}
```

```
export default MultiCounter;
```

Dynamic Form Fields

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

function DynamicForm() {
  const [fields, setFields] = useState([{ name: '' }]);

  const addField = () => {
    setFields([...fields, { name: '' }]);
  };

  const updateField = (index, value) => {
    const newFields = [...fields];
    newFields[index].name = value;
    setFields(newFields);
  };

  return (
    <div>
      {fields.map((field, index) => (
        <input
          key={index}
          value={field.name}
          onChange={(e) => updateField(index, e.target.value)}
          placeholder={`Field ${index + 1}`}
        />
      ))}
      <button onClick={addField}>Add Field</button>
    </div>
  );
}

export default DynamicForm;
```

State with API Call Example

```
import React, { useState, useEffect } from 'react';

function FetchExample() {
  const [data, setData] = useState([]);
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    fetch('https://jsonplaceholder.typicode.com/posts')
      .then((response) => response.json())
      .then((data) => {
        setData(data);
        setLoading(false);
      });
  }, []);
  if (loading) return <p>Loading...</p>;

  return (
    <div>
      <h1>Posts</h1>
      <ul>
        {data.slice(0, 5).map((post) => (
          <li key={post.id}>{post.title}</li>
        ))}
      </ul>
    </div>
  );
}

export default FetchExample;
```

Debounced Input Example

```
import React, { useState, useEffect } from 'react';

function DebouncedInput() {
  const [text, setText] = useState('');
  const [debouncedText, setDebouncedText] = useState('');

  useEffect(() => {
    const handler = setTimeout(() => {
      setDebouncedText(text);
    }, 500);

    return () => clearTimeout(handler);
  }, [text]);

  return (
    <div>
      <input
        type="text"
        value={text}
        onChange={(e) => setText(e.target.value)}
      />
      <p>Debounced Text: {debouncedText}</p>
    </div>
  );
}

export default DebouncedInput;
```