ChatGPT

React useState + Arrow vs Normal Function Notes

🔍 What is `useState`?

- A React Hook that lets you add state (data that changes) to functional components.

```
const [state, setState] = useState(initialValue);
```

Example:

```
const [ingredients, setIngredients] = useState([]);
```

👍 Prevent Form Reloading

```
event.preventDefault();
```

- Stops the page from refreshing when a form is submitted.

Getting Form Data

```
const formData = new FormData(event.currentTarget);
const newIngredient = formData.get("ingredient");
```

- `event.currentTarget` refers to **the form that triggered the submit event**.
- `formData.get("ingredient")` fetches input value by the `name="ingredient"` attribute.

Updating State with Previous Value (Functional Way)

Arrow Function:

```
setIngredients((prev) => [...prev, newIngredient]);
```

Normal Function:

```
setIngredients(function (prev) {
  return [...prev, newIngredient];
});
```

- Both do the same thing: take old state, add new item, return new array.

Why use spread `...`?

```
[...prev, newIngredient]
```

- **Creates a new array**, doesn't change the old one (immutability).
- Spread operator works in **JavaScript**, not just React.
- Clean, readable, modern syntax.

⚖️ `.concat()` vs `...spread`

- `array.concat(item)` also returns a new array
- `spread` is more flexible and readable:

```
[...arr, item1, item2]
```

⚛ Arrow Function vs Normal Function

Feature	Arrow Function	Normal Function
Implicit return	Yes (if no `{}` used)	No
Needs `return`?	Only with `{}` block	Always
Cleaner syntax		More verbose

Examples:

```
// Arrow function - implicit return
const add = (a, b) => a + b;

// Arrow with return
const add = (a, b) => {
  return a + b;
};

// Normal function
```

```
function add(a, b) {
  return a + b;
}
```

React map example with arrow:

```
const ingredientElements = ingredients.map((item) => (
  <li key={item}>{item}</li>
));
```

🪶 Summary

- `useState` gives state to React components
- Always use **spread** or `concat()` to update arrays without mutating
- Arrow functions are shorter but normal functions are easier to read at first
- `prev` is just a **parameter name for previous state**, React gives it
- Using forms with `FormData` and `event.currentTarget` works even if multiple forms exist

You're on fire, Raj! Keep this handy for future projects.