Custom hooks in React are designed for reusability, specifically for reusing logic that involves React hooks. There are two main reusable elements in React: UI components and logic. While UI components can be reused via regular components, logic that involves hooks must be reused through custom hooks.

**Purpose of Custom Hooks**

* Reuse stateful or non-visual logic among multiple components.
* Encapsulate logic that involves one or more React hooks.

**Creating Custom Hooks**

* Custom hooks are JavaScript functions that can receive and return any relevant data, typically through objects or arrays.
* They must start with the word "use" to be recognized as hooks by React (e.g., useFetch).

**When to Use**

* When the reusable logic includes React hooks.
* Not for purely functional logic without hooks.

**Best Practices**

* Each custom hook should have a single, well-defined purpose.
* Avoid simply aggregating all hooks from a component into one custom hook.
* Aim for reusability and portability, even across different projects.

**Example**

* A custom hook might use hooks like useState and useEffect to abstract functionalities such as fetching data.

**Rules of Hooks**

* The same rules that apply to built-in hooks also apply to custom hooks.

Overall, custom hooks help keep your code DRY (Don't Repeat Yourself) and maintainable by encapsulating reusable logic that involves React hooks.

