REACT

COMPONENT FUNCTIONS

Usage:

```
function Name(props) {
  return elements;
}
```

Example:

```
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}
```

RENDERING A COMPONENT

Usage:

```
const element = <Component key="value" />;
ReactDOM.render(
  element,
  target
);
```

Example:

```
const element = <Welcome name="Sara" />;
ReactDOM.render(
  element,
  document.getElementById('root')
);
```

JSX

JavaScript expressions	Wrap in curly braces `Hello, {formatName(user)}!`
Attribute names	Camel case, using JS standards `<div className="hidden"></div>;`

Attribute values	String literals in quotes; expressions without quote
	`<div tabIndex="0"></div>;`
	`<img src={user.avatarUrl}></img>;`

Empty tags	Close with space and slash `<img src={user.avatarUrl} />;`

BUILDING A LIST

Usage:

```
const name = array.map((element) =>
  <li>{element}</li>
);
return (
  <ul>{name}</ul>
);
```

Example:

```
const listItems = numbers.map((number) =>
  <li>{number}</li>
);
return (
  <ul>{listItems}</ul>
);
```

CREATING A COMPONENT HIERARCHY

1. Start with a mock
2. Draw a box around each item that does one thing (including groups)
3. Name the boxes (semantically)
4. Make a hierarchy

- FilterableProductTable
  - ‣ SearchBar
  - ‣ ProductTable
    - » ProductCategoryRow
    - » ProductRow