Instagram Feed Using ReactJS

Key Topics

- BEM (Block Element Modifier)
- Intro to ReactJS
- Intro to Redux

DEMO

●●●

BEM

What

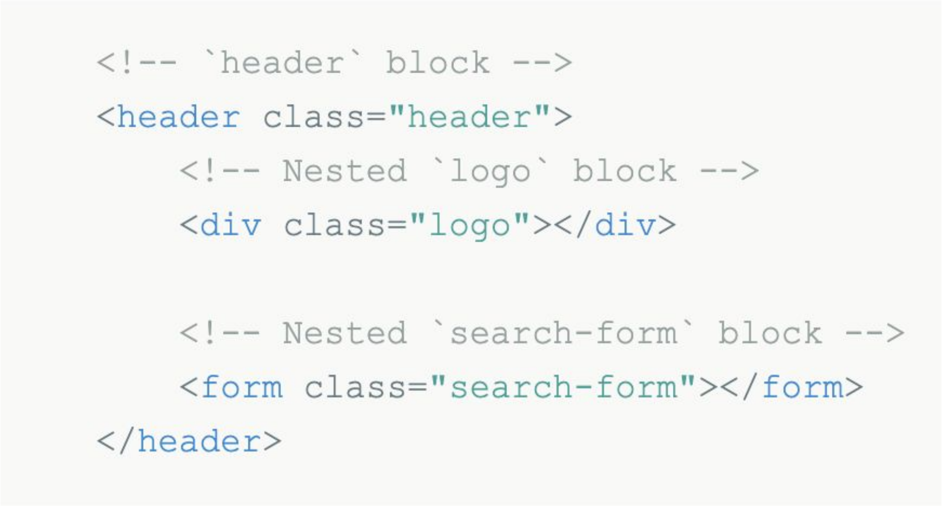
- Enforces naming conventions
- Helps you to create reusable components

Why

- Ensures that everyone who participates in the development of a website works with a single codebase and speaks the same language.
- Will better prepare you for design changes made to your website

Block

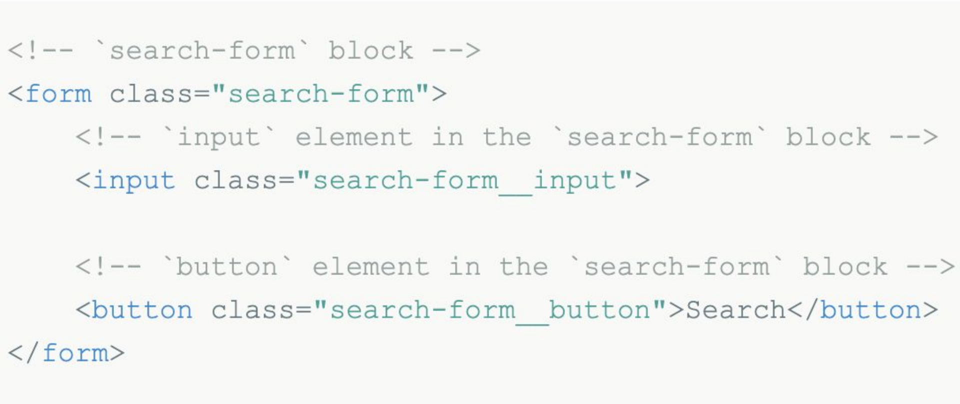
- The block name describes its purpose ("What is it?" — menu or button), not its state ("What does it look like?" — red or big).
- Blocks can be nested in each other.
- You can have any number of nesting levels.



```html
<!-- `header` block -->
<header class="header">
    <!-- Nested `logo` block -->
    <div class="logo"></div>

    <!-- Nested `search-form` block -->
    <form class="search-form"></form>
</header>
```

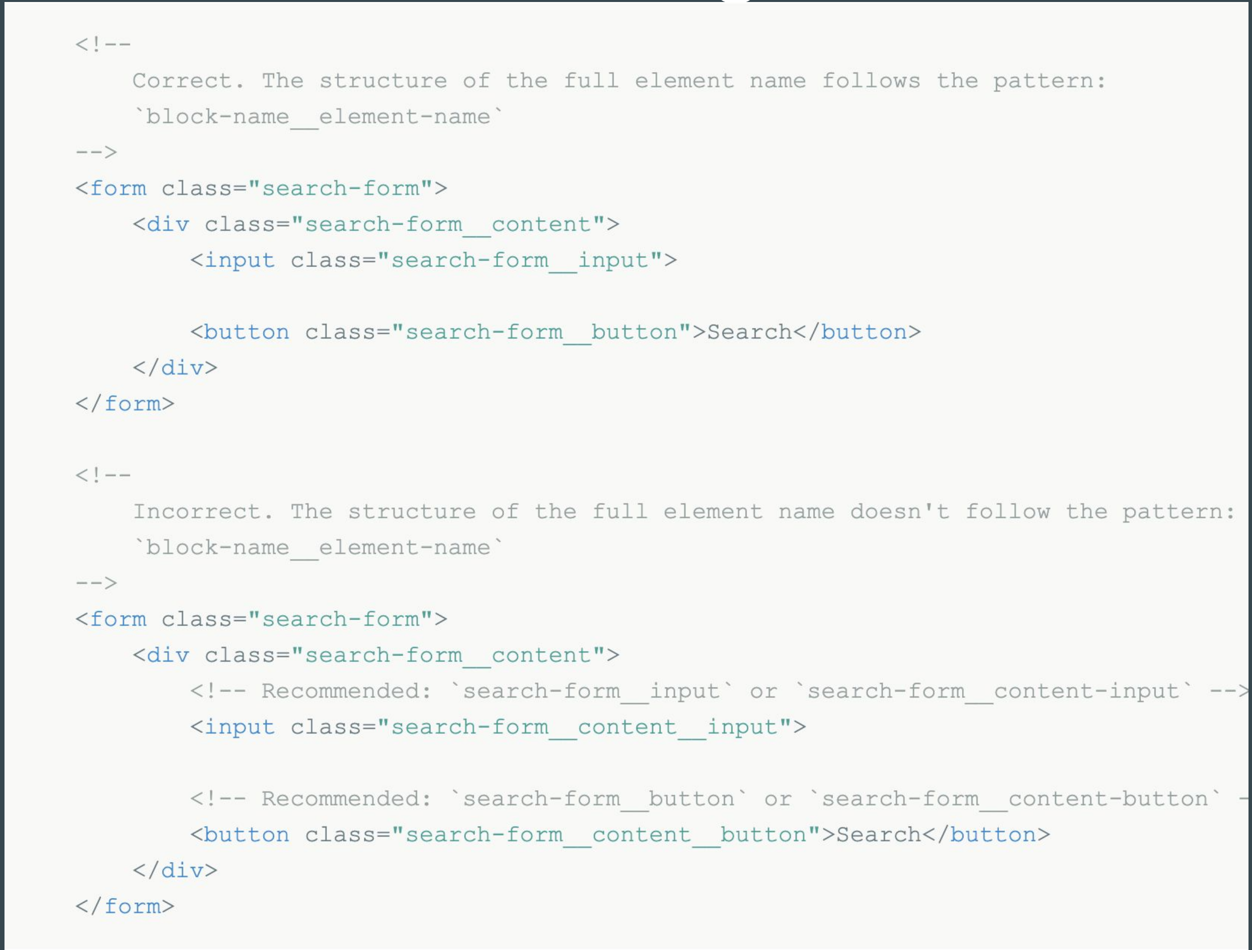
Element

- The element name describes its purpose ("What is this?" — item, text, etc.), not its state ("What type, or what does it look like?" — red, big, etc.).
- A composite part of a block that can't be used separately from it.
- Named as block-name__element-name.



```html
<!-- `search-form` block -->
<form class="search-form">
    <!-- `input` element in the `search-form` block -->
    <input class="search-form__input">

    <!-- `button` element in the `search-form` block -->
    <button class="search-form__button">Search</button>
</form>
```

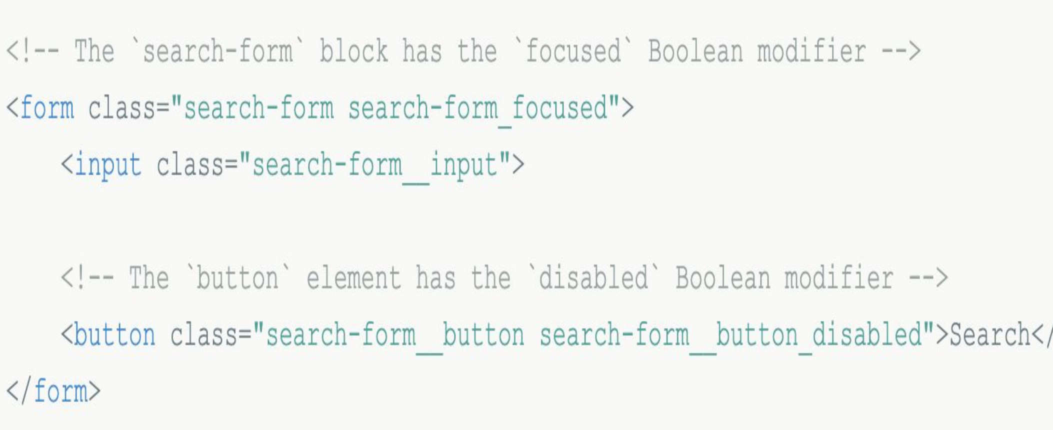
Nesting



```
<!--
    Correct. The structure of the full element name follows the pattern:
    `block-name__element-name`
-->
<form class="search-form">
    <div class="search-form__content">
        <input class="search-form__input">

        <button class="search-form__button">Search</button>
    </div>
</form>

<!--
    Incorrect. The structure of the full element name doesn't follow the pattern:
    `block-name__element-name`
-->
<form class="search-form">
    <div class="search-form__content">
        <!-- Recommended: `search-form__input` or `search-form__content-input` -->
        <input class="search-form__content__input">

        <!-- Recommended: `search-form__button` or `search-form__content-button` -
        <button class="search-form__content__button">Search</button>
    </div>
</form>
```

Modifier

- The modifier name describes its appearance, its state, and its behavior.
- An entity that defines the appearance, state, or behavior of a block or element.
- Named as element-name_modifier or element-name--modifier.



```html
<!-- The `search-form` block has the `focused` Boolean modifier -->
<form class="search-form search-form_focused">
    <input class="search-form__input">

    <!-- The `button` element has the `disabled` Boolean modifier -->
    <button class="search-form__button search-form__button_disabled">Search</
</form>
```

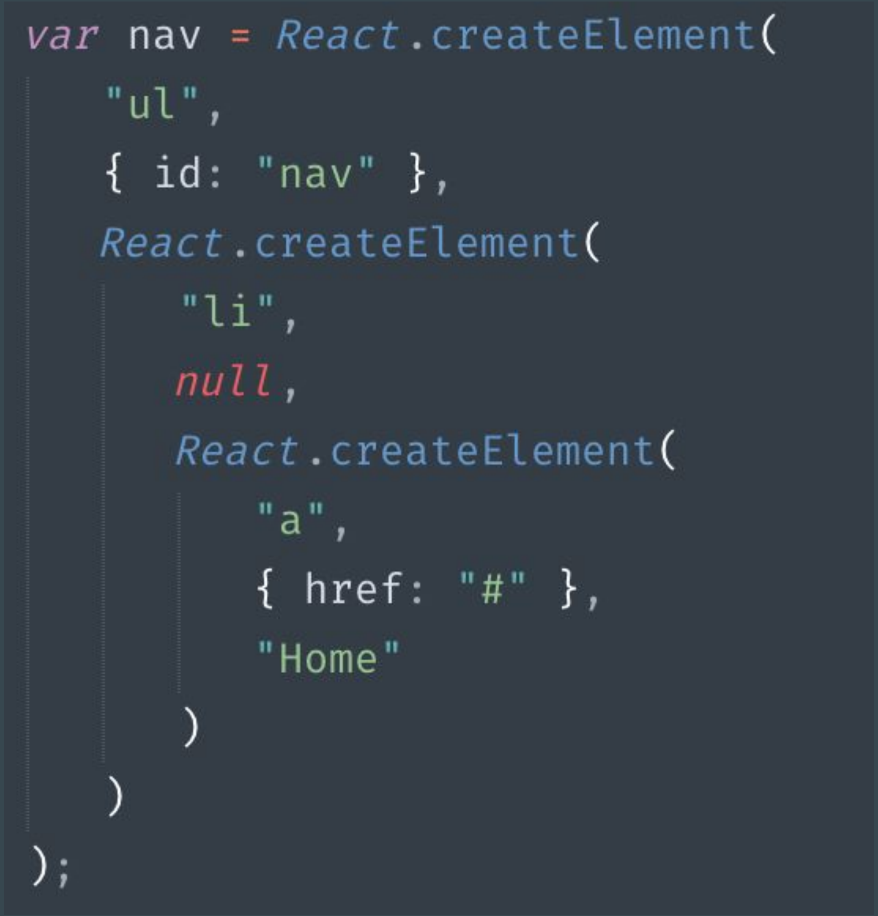
https://en.bem.info/

ReactJS

A javascript library that helps building modular UI easily with better performance, with the help of concepts like components and virtual dom, etc.

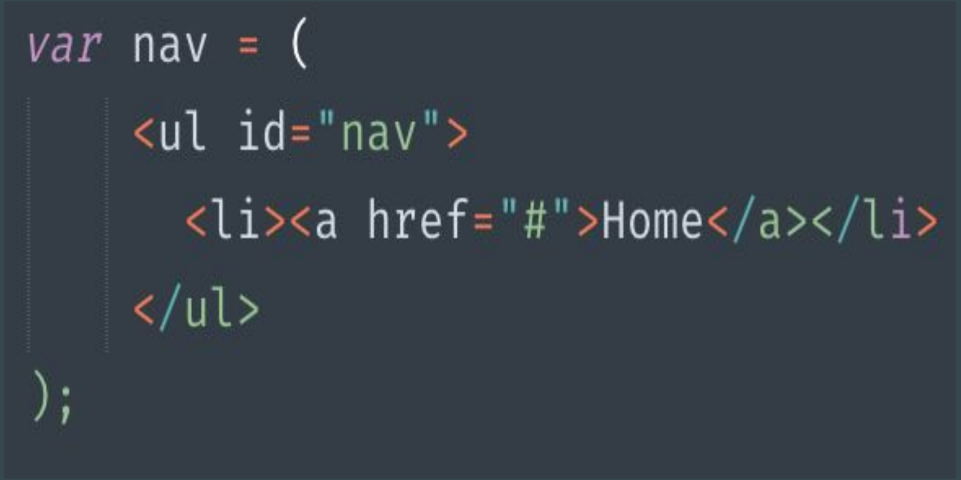
- JSX - javascript syntax extension used for creating react elements easily
- Components
- Props
- States
- API calls
- Event handling

Non-JSX



```
var nav = React.createElement(
    "ul",
    { id: "nav" },
    React.createElement(
        "li",
        null,
        React.createElement(
            "a",
            { href: "#" },
            "Home"
        )
    )
);
```

Using JSX



```
var nav = (
    <ul id="nav">
      <li><a href="#">Home</a></li>
    </ul>
);
```

Components

- Components let you split the UI into independent, reusable pieces, and think about each piece in isolation
- Components are like JavaScript functions.
- There can be two types of components - functional and class based

Props
(are Read-Only)

- Behaves just like function argument.
- Used for passing data from parent to child component(s).

State

- Stores data that belongs to the component.
- To re render any component the state needs to be updated.
- Helps in making a component truly reusable.

API call

- Use `componentDidMount` lifecycle method (gets hooked when rendering of component is finished).
- Fetch data using native Fetch API.
- Do whatever with the data.

Event Handling

Flashback

- Created an app using create-react-app boilerplate
- Created few components while making use of props as well as state.
- Used a lifecycle method to make an API call.

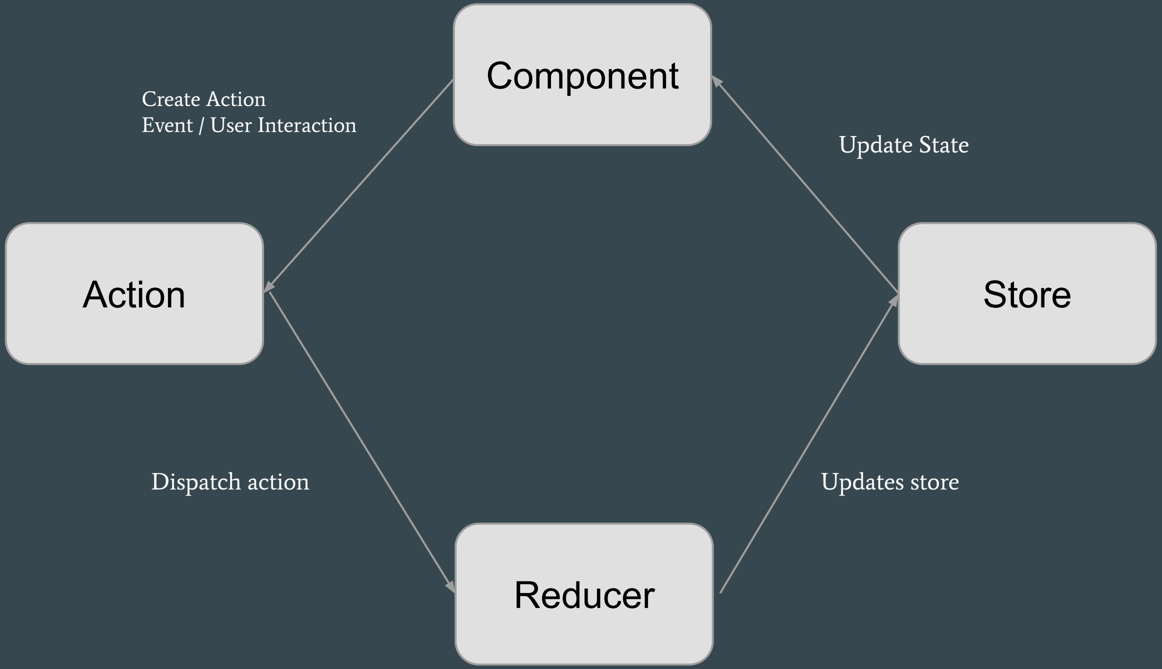
Q/A

REDUX

REDUX

Maintains the state of an entire application in a single immutable state tree (object), which can’t be changed directly. When something changes, a new object is created (using actions and reducers).

REDUX data flow



Create Action
Event / User Interaction

Update State

Application global store

Dispatch action

Updates store

Creates and returns new
State

Q/A

THANK YOU

•••