

# Behind The Scenes

---

## Understanding & Optimizing React

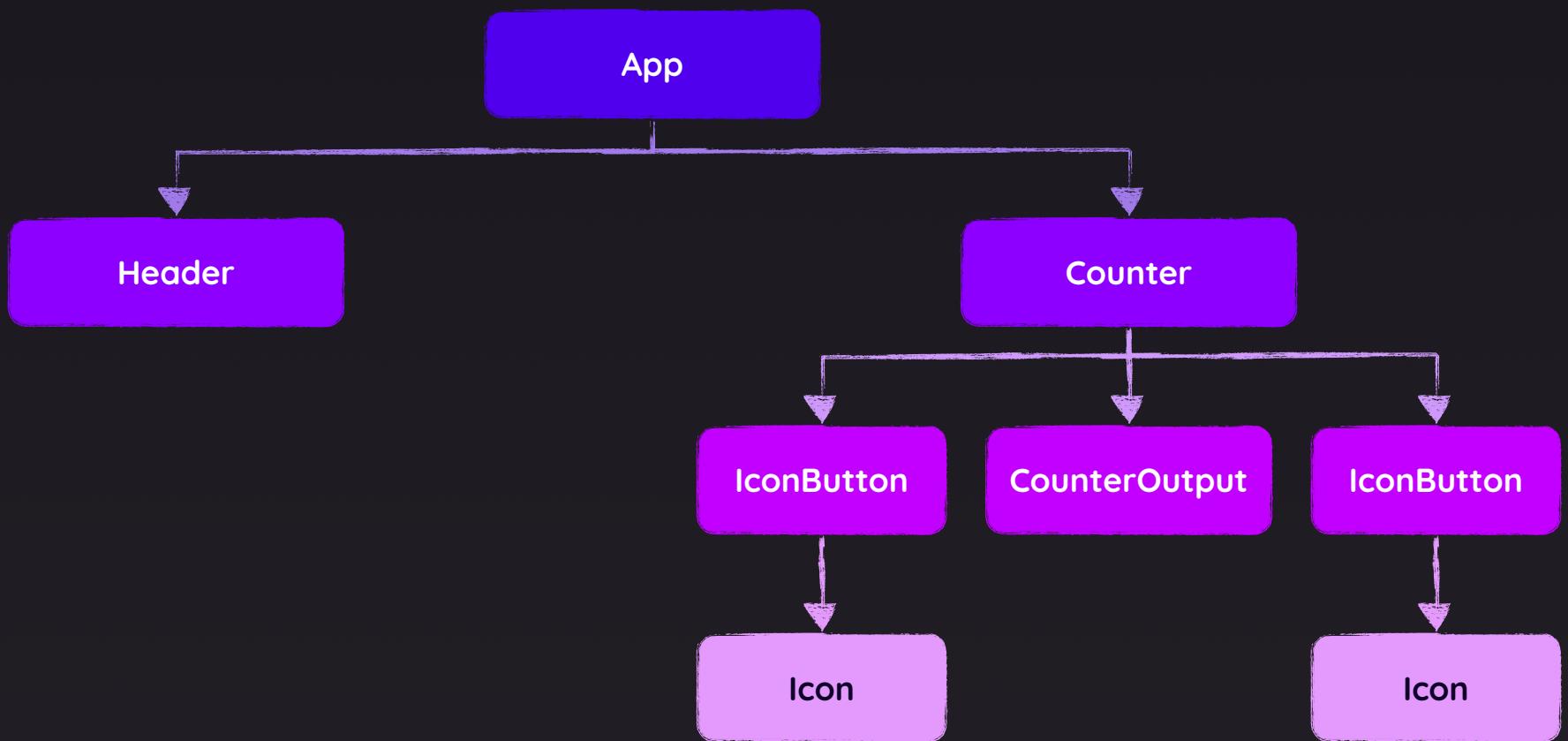
- ▶ How React **Updates The DOM**
- ▶ **Avoiding** Unnecessary Updates
- ▶ A Closer Look At **Keys**
- ▶ State **Scheduling** & State **Batching**

# How Does React Update The DOM?

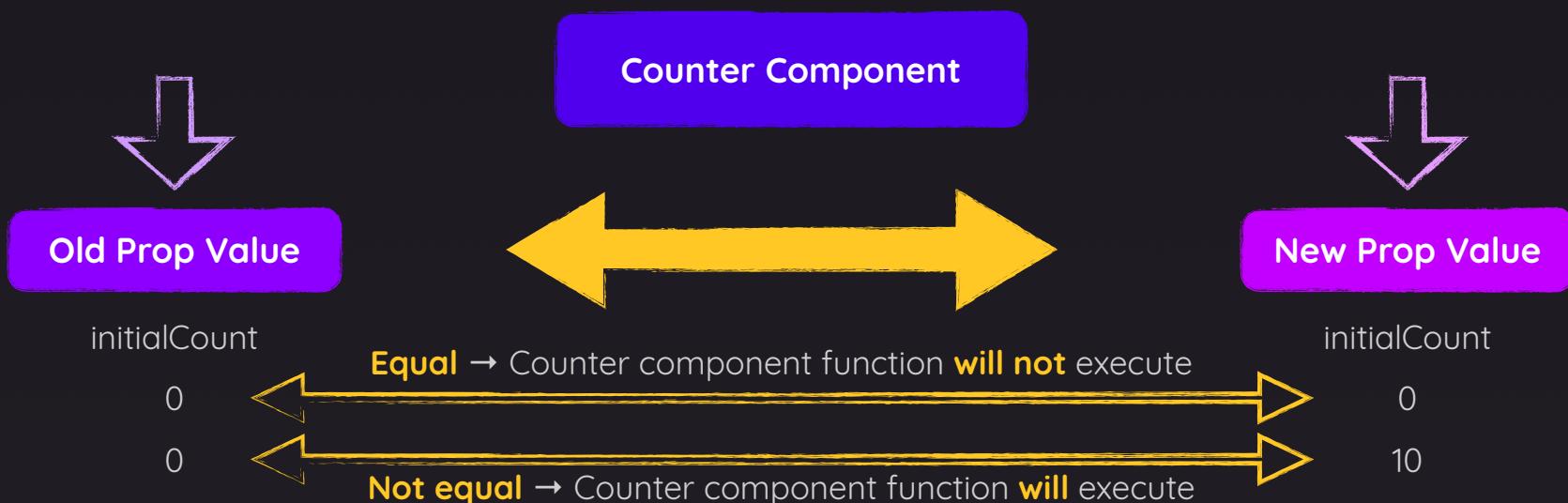
And how are component  
functions executed?

# React Builds A Component Tree

The **relation between components** is internally modelled as a tree structure



# memo() compares prop values



# Don't overuse memo()!

Use it **as high up in the component tree as possible**

→ blocking a component execution there will also block all child component executions

Checking props with memo() **costs performance!**

→ don't wrap it around all your components — it will just add a lot of unnecessary checks

**Don't** use it on components where **props will change frequently**

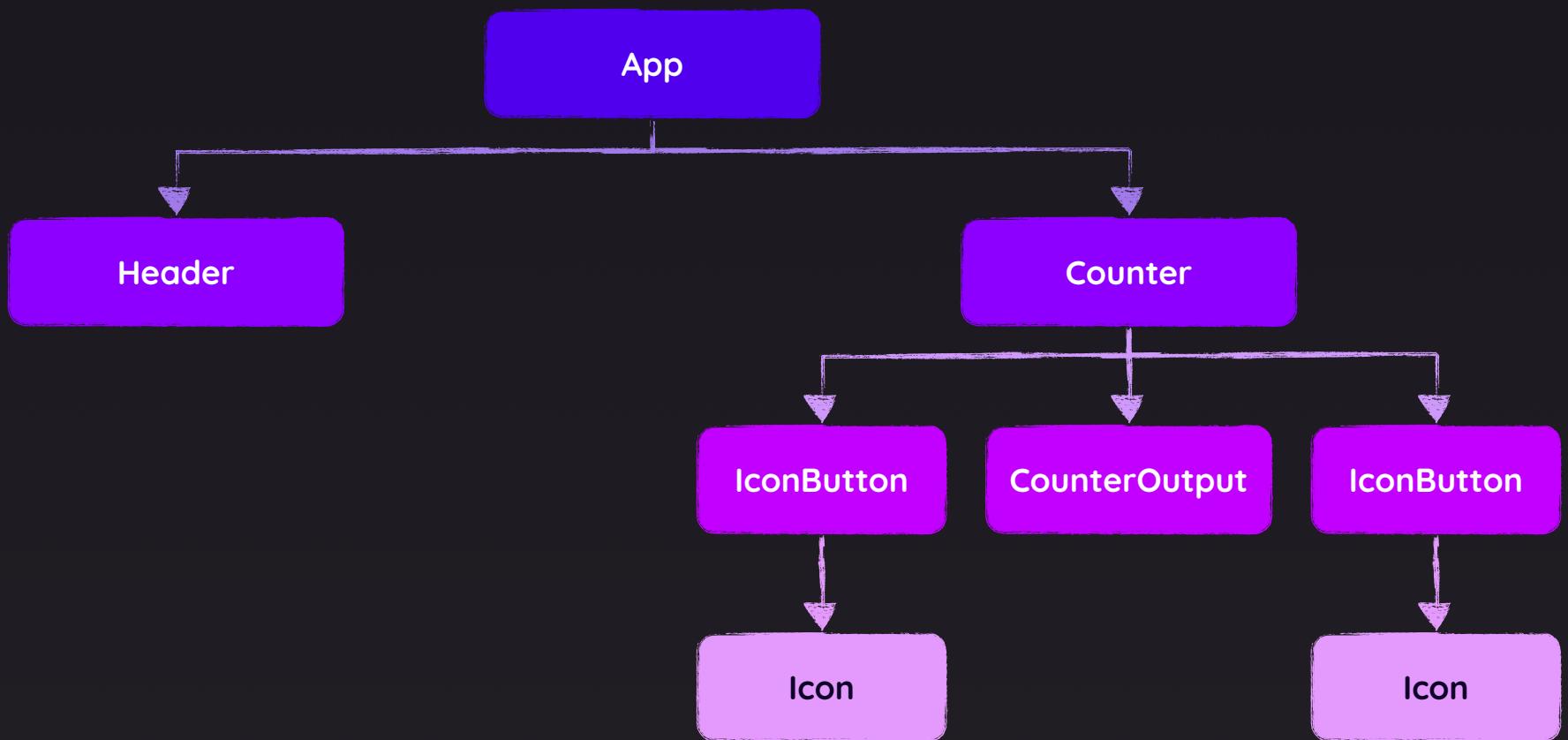
→ memo() would just perform a meaningless check in such cases (which costs performance)

# React checks for necessary DOM updates via a “Virtual DOM”

It creates & compares virtual DOM snapshots to find out which parts of the rendered UI need to be updated

# How React Updates The Website UI

Step 1: Creating a Component Tree



# How React Updates The Website UI

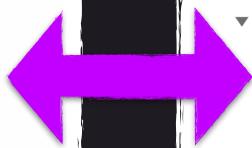
## Step 2: Creating a Virtual Snapshot of the Target HTML Code

```
▼< <header id="main-header">
    
    <h1>React – Behind The Scenes</h1>
</header>
▼< <main>
    <section id="configure-counter"> flex
        <h2>Set Counter</h2>
        <input type="number" value="0">
        <button>Set</button>
    </section>
    <section class="counter">
        <p class="counter-info">
            "The initial counter value was "
            <strong>0</strong>
            ". It"
            ▶<strong>...</strong>
            " prime number."
        </p>
        <p> flex
            ▶<button class="button">...</button> flex
            <span class="counter-output">1</span>
            ▶<button class="button">...</button> flex
        </p>
    </section>
</main>
```

# How React Updates The Website UI

Step 3: Compare New Virtual DOM Snapshot to Previous (Old) Virtual DOM Snapshot

Old Snapshot



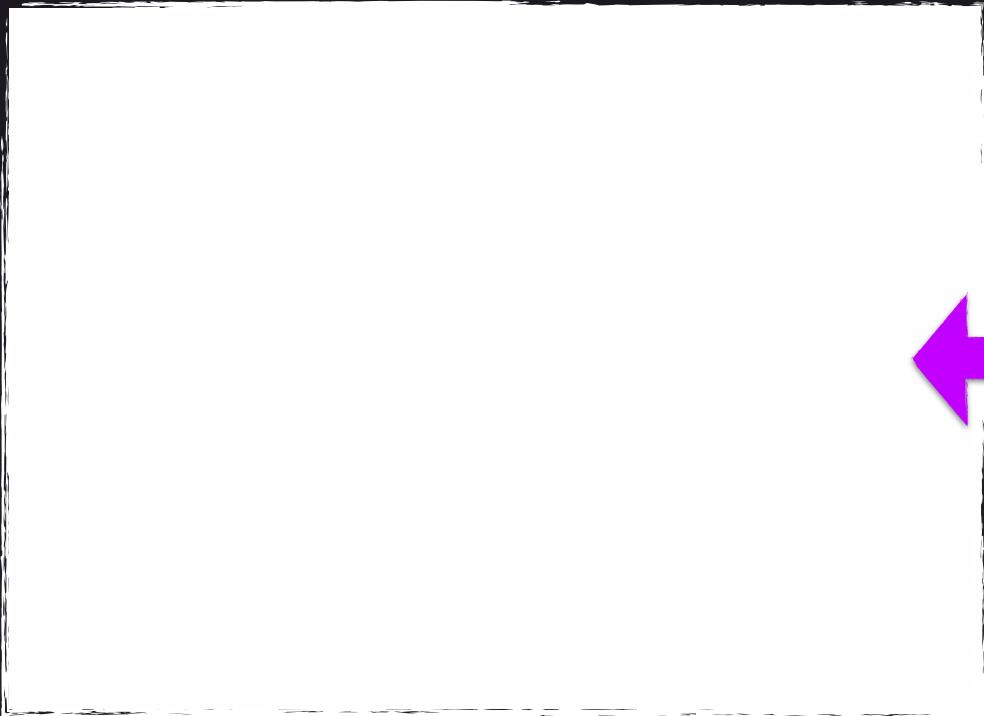
New Snapshot

```
<header id="main-header">
  
  <h1>React - Behind The Scenes</h1>
</header>
<main>
  <section id="configure-counter"> (flex)
    <h2>Set Counter</h2>
    <input type="number" value="0">
    <button>Set</button>
  </section>
  <section class="counter">
    <p class="counter-info">
      "The initial counter value was "
      <strong>0</strong>
      ". It"
      ><strong>...</strong>
      " prime number."
    </p>
    <p> (flex)
      ><button class="button">...</button> (flex)
        <span class="counter-output">0</span>
      ><button class="button">...</button> (flex)
    </p>
  </section>
</main>
```

# How React Updates The Website UI

Step 4: Identify & Apply Changes to the “Real DOM”

Old Snapshot

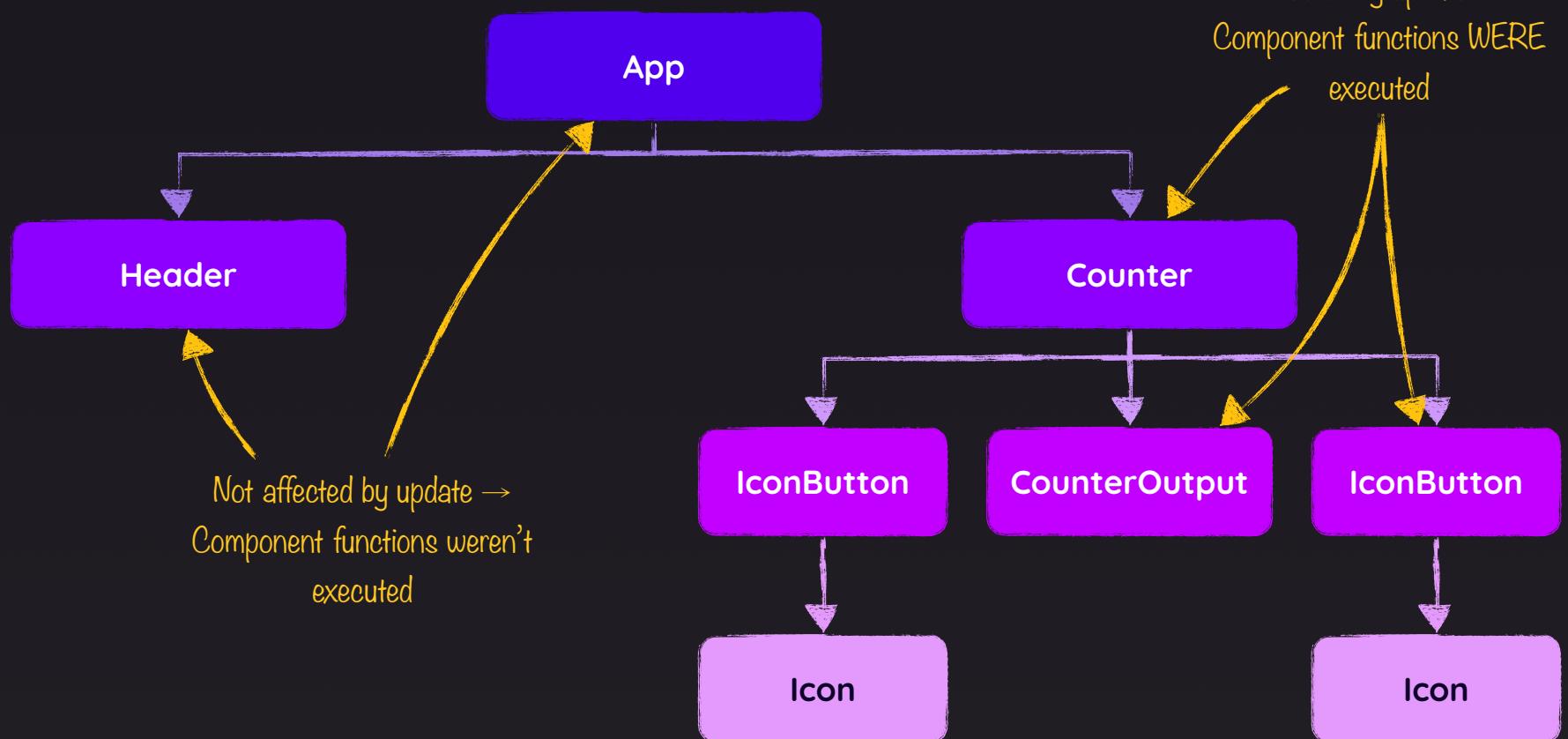


New Snapshot

```
<header id="main-header">
  
  <h1>React - Behind The Scenes</h1>
</header>
<main>
  <section id="configure-counter"> (flex)
    <h2>Set Counter</h2>
    <input type="number" value="0">
    <button>Set</button>
  </section>
  <section class="counter">
    <p class="counter-info">
      "The initial counter value was "
      <strong>0</strong>
      ". It"
      ><strong>...</strong>
      " prime number."
    </p>
    <p> (flex)
      ><button class="button">...</button> (flex)
        <span class="counter-output">0</span>
      ><button class="button">...</button> (flex)
    </p>
  </section>
</main>
```

# Updating The Website UI After A Button Click

Step 1: Creating a Component Tree



# How React Updates The Website UI

## Step 2: Creating a Virtual Snapshot of the Target HTML Code

```
▼ <header id="main-header">
  
  <h1>React – Behind The Scenes</h1>
</header>
▼ <main>
  ▼ <section id="configure-counter"> flex
    <h2>Set Counter</h2>
    <input type="number" value="0">
    <button>Set</button>
  </section>
  ▼ <section class="counter">
    ▼ <p class="counter-info">
      "The initial counter value was "
      <strong>0</strong>
      ". It"
      ▶ <strong>...</strong>
      " prime number."
    </p>
    ▼ <p> flex
      ▶ <button class="button">...</button> flex
      <span class="counter-output">1</span>
      ▶ <button class="button">...</button> flex
    </p>
  </section>
</main>
```

# How React Updates The Website UI

Step 3: Compare New Virtual DOM Snapshot to Previous (Old) Virtual DOM Snapshot

## Old Snapshot

```
▼<header id="main-header">
  
  <h1>React - Behind The Scenes</h1>
</header>
▼<main>
  ▼<section id="configure-counter"> [flex]
    <h2>Set Counter</h2>
    <input type="number" value="0">
    <button>Set</button>
  </section>
  ▼<section class="counter">
    ▼<p class="counter-info">
      "The initial counter value was "
      <strong>0</strong>
      ". It"
      ▶<strong>...</strong>
      " prime number."
    </p>
    ▼<p> [flex]
      ▶<button class="button">...</button> [flex]
      <span class="counter-output">0</span>
      ▶<button class="button">...</button> [flex]
    </p>
  </section>
</main>
```

## New Snapshot

```
▼<header id="main-header">
  
  <h1>React - Behind The Scenes</h1>
</header>
▼<main>
  ▼<section id="configure-counter"> [flex]
    <h2>Set Counter</h2>
    <input type="number" value="0">
    <button>Set</button>
  </section>
  ▼<section class="counter">
    ▼<p class="counter-info">
      "The initial counter value was "
      <strong>0</strong>
      ". It"
      ▶<strong>...</strong>
      " prime number."
    </p>
    ▼<p> [flex]
      ▶<button class="button">...</button> [flex]
      <span class="counter-output">1</span>
      ▶<button class="button">...</button> [flex]
    </p>
  </section>
</main>
```



# How React Updates The Website UI

Step 4: Identify & Apply Changes to the “Real DOM”

## Old Snapshot

```
▼<header id="main-header">
  
  <h1>React - Behind The Scenes</h1>
</header>
▼<main>
  ▼<section id="configure-counter"> [flex]
    <h2>Set Counter</h2>
    <input type="number" value="0">
    <button>Set</button>
  </section>
  ▼<section class="counter">
    ▼<p class="counter-info">
      "The initial counter value was "
      <strong>0</strong>
      ". It"
      ▶<strong>...</strong>
      " prime number."
    </p>
    ▼<p> [flex]
      ▶<button class="button">...</button> [flex]
      <span class="counter-output">0</span>
      ▶<button class="button">...</button> [flex]
    </p>
  </section>
</main>
```

## New Snapshot

```
▼<header id="main-header">
  
  <h1>React - Behind The Scenes</h1>
</header>
▼<main>
  ▼<section id="configure-counter"> [flex]
    <h2>Set Counter</h2>
    <input type="number" value="0">
    <button>Set</button>
  </section>
  ▼<section class="counter">
    ▼<p class="counter-info">
      "The initial counter value was "
      <strong>0</strong>
      ". It"
      ▶<strong>...</strong>
      " prime number."
    </p>
    ▼<p> [flex]
      ▶<button class="button">...</button> [flex]
      <span class="counter-output">1</span>
      ▶<button class="button">...</button> [flex]
    </p>
  </section>
</main>
```

