

How Does React Js Works

Here, in this article, we will learn how exactly react works ~~behind the scenes.~~

Which concepts are simplified by React? **Do you need to refresh the page after you like a post on Facebook?** It's no. Right? The page gets updated **simultaneously** as soon as it is being clicked. This is where react comes into the picture. The whole application in react is divided into several **components**. Each component acts independently and gets updated on its own if it encounters any change.

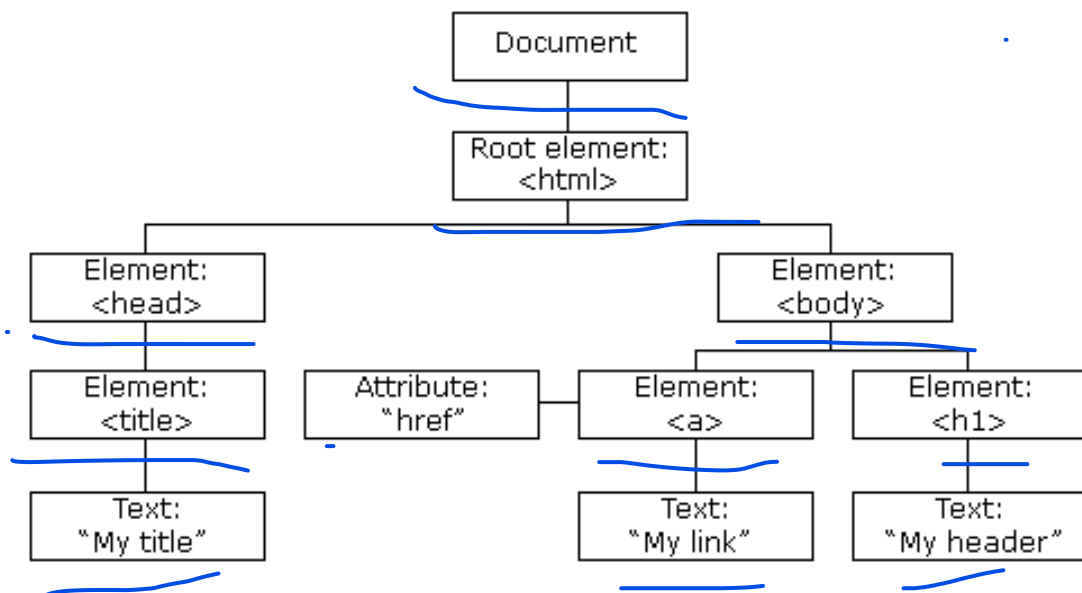


DOM (Document Object Model)



DOM

The **Document Object Model** (DOM) is a programming interface for **HTML** and **XML** (Extensible markup language) documents. It defines the logical structure of documents and the way a document is accessed and manipulated. When a webpage is loaded, the browser creates the **DOM** of the page. It represents the documents as **nodes** or **objects**. A basic DOM looks like this:



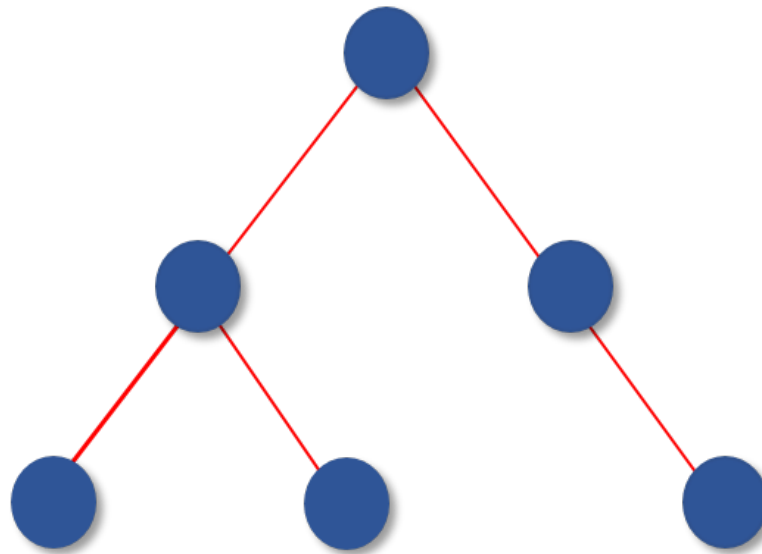
(Basic DOM View)

Slower HTML

DOM creation and **update** are a **slower** process. To speed up the process, Facebook developed ReactJS. ReactJS uses the concept of **virtual DOM** which makes the **loading** faster. A virtual DOM is just a **lightweight javascript object** which is just the copy of a real DOM.

3. After the final calculations are made, the real DOM will only be updated with those elements which have actually changed.

Real DOM (updated)



How does the diffing algorithm work?

While diffing any 2 trees, react first compare the **root** of both the **trees**. It runs differently depending on the root.

Whenever the root element is of a different type, the entire tree is **destroyed** and a new tree is built from **scratch**. i.e. if we compare **<p>** with **<div>** or **<article>** with **<a>**, a **completely** new **DOM** is **constructed**.

Whenever props and state is accurate then ui is re-rendered compare them virtual dom vs real dom real dom is updated so changes reflected in real dom and render ui.

Component will Unmount While Destroying



While destroying the old tree, **componentWillUnmount()** function is executed, which performs all the necessary cleanup within the method, such as **invalidating** timers, canceling network requests, or cleaning up any subscriptions that were created during **componentDidMount()**.

Again, while constructing the new tree, **componentWillMount()** function is executed first (which is the only lifecycle method that is called on the server-side when you use server-side rendering), followed by **componentDidMount()** (which is executed once in the lifecycle of a component and after the first render).

When the root elements are of the same type, then react compares the attributes of the element. If it encounters a change, it updates that part only.

Code in the file

```
<div className="initial-div" title="facebook" />
<div className="new-div" title="facebook" />
```

In the above snippet, upon comparison, react knows only to modify the **className attribute** of the **underlying node**.

className is the only property