

dotnotes

REACT BASICS

A Javascript library for building reusable, flexible UI components.

JSX ⇒

<MARKUP/> + Logic();

JSX is a supercharged template language that combines structure (UI) & function (rendering).

A SYNTAX EXTENSION TO JAVASCRIPT

COMPONENTS

Components are the building blocks of React.

They generate reusable pieces of UI, created in JSX, that are injected into the DOM.

Components come in two flavors:

function Class

Data can be passed into components via props or maintained internally via state.

Must be capitalized!

Accepts a "props" object argument.

DOCUMENT OBJECT MODEL

EXAMPLE

```
function Cactus (props) {
  return <h1>{props.name}</h1>
}
```

Returns a React element.

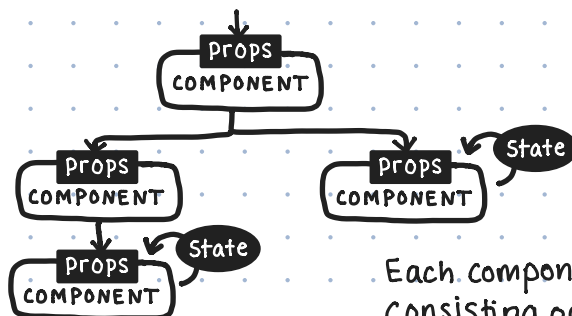
DOM



The DOM is the browser's internal programmatic representation of a web page.

Why components?

- ✓ Reusability
- ✓ Organization
- ✓ Blended JS & HTML
- ✓ Encapsulation of state



Each component has a lifecycle consisting of three different phases:



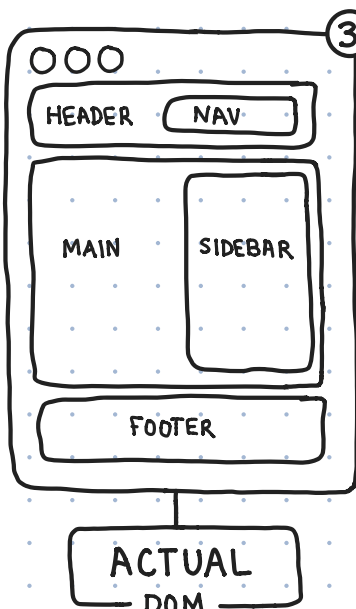
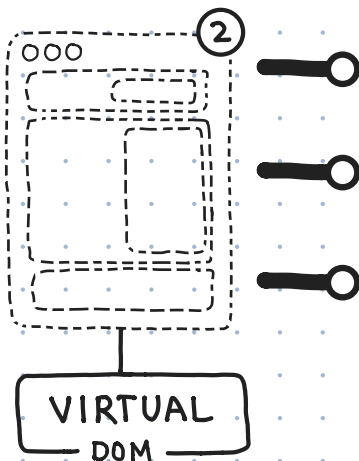
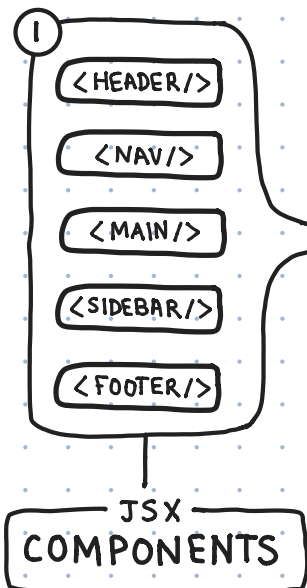
HOW REACT WORKS

① Components generate React elements.

② Updates are passed to the virtual DOM, which selectively re-renders only changed elements.

③ React syncs application state with the DOM in a process called reconciliation.

render()



- The render method is called whenever the state or props of a component is changed.
- This triggers the creation of a new virtual DOM.



The Diffing Algorithm figures out how to efficiently update the UI so that the virtual & actual DOM trees match.

Time complexity: O(n)

Components allow you to think about each piece in isolation.

The virtual DOM is an interface that sits between <React/> code & the actual DOM.