**REACT WITH ZERO TO MASTERY**

If before we asked the server for updates for every page, now we take the whole app code and each web app acts more and more as a desktop app.

With every year, people rely less and less on HTML and more and more on JS.

The DOM (Document Object Model) is what the browser uses to display a website or a webapp.

It is the tree representation of the page.

All js does is manipulating the DOM.

With an imperative approach, you just modify the DOM directly with js. That is a very expensive operation everytime. (For example, every 5 seconds display a new add, show the icon of an user if it is logged , update the feed, etc).

React uses the declarative approach, where react handles the DOM.

React says: “I’ll find the best way for me to change the DOM, just declare to me how your app looks like.”

So you declare a js object about how the app should look like and react is going to hold that blueprint and it will manipulate the DOM on the best way.

So the REACT principles would be:  
  
1. Don’t touch the DOM. I’ll do it.

2. Build websites like lego blocks (it is designed with the concept of reusable components).

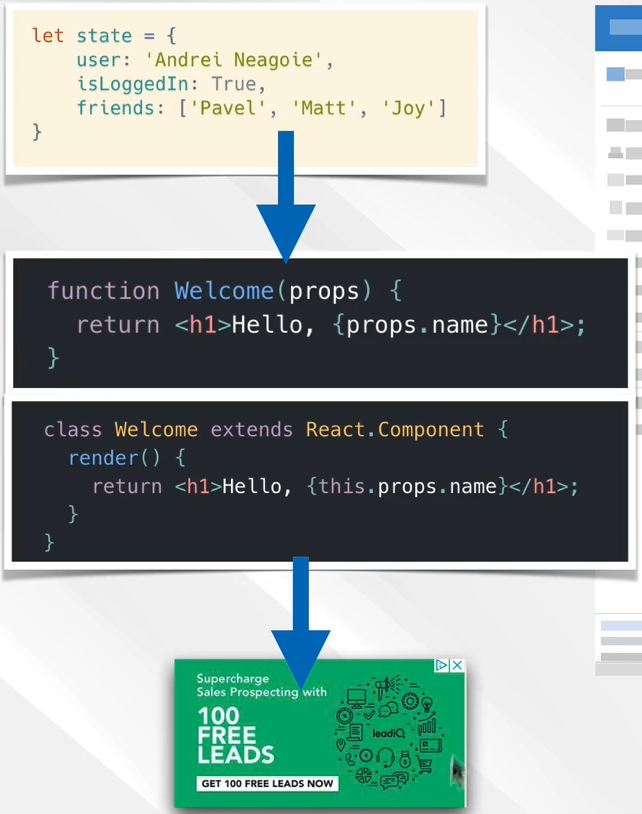
So we have app data. Then components are created based on that data.

So! There are only 3 things we have to think about when we are developing with react:

1-.Decide on components. (What is a component, how much do we want to break down a component, how do we build reusable components, how to break the app into components)

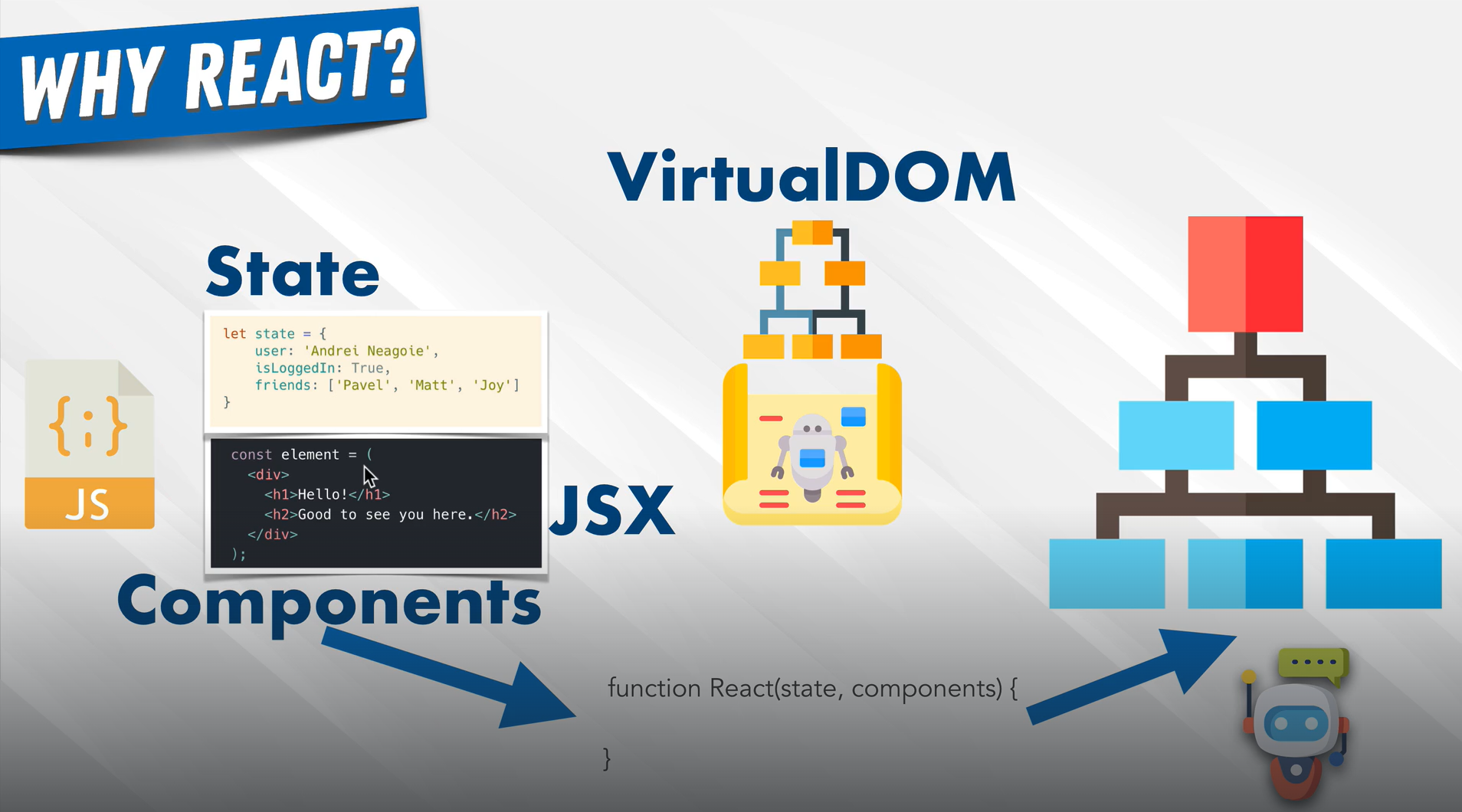
2. Decide the State and where it lives.

3. What changes when state changes



3-.Third concept: unidirectional dataform.

From the previous points, react creates a VirtualDOM. That is a js representation of the DOM.



The process is unidirectional. As soon as some information changes, it is going to change everything. Data can never move up

**CREATE NEW APP**

Meta provides a tool for creating the application, npx create-react-app my-app

npm install create-react-app -> Installs on one folder

npm install -g create-react-app -> installs globally

npm list -g cowsay -> shows globally the versions of that packages we have installed

npx installs the packages when you call one, executes it inmediatly and then delets it from the disk.

Package.json tells all the packages that are needed in order to run an app.

Anyway, create-react-app is fucked up so we use

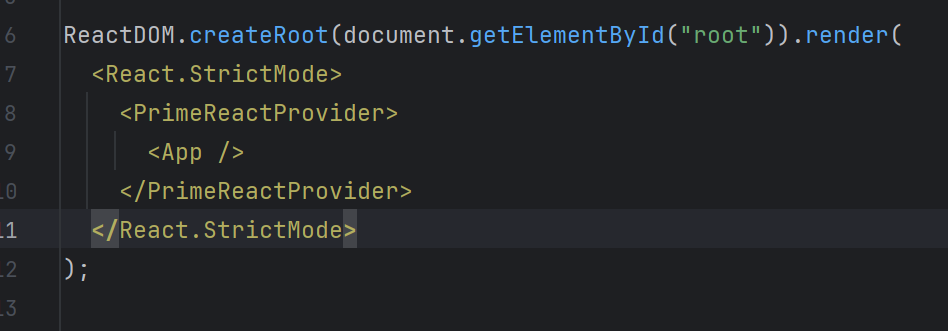
npm create vite@latest tornado2 -- --template react

The useful code is under th src folder.

Think about the App.jsx as about the whole application.

<React.StrictMode>

Lo que esté dentro nos va a dar avisos y errores sobre las versiones.



This looks for the element with the root id and randers the react app inside of it.

Everything lives inside of what the StrictMode is.