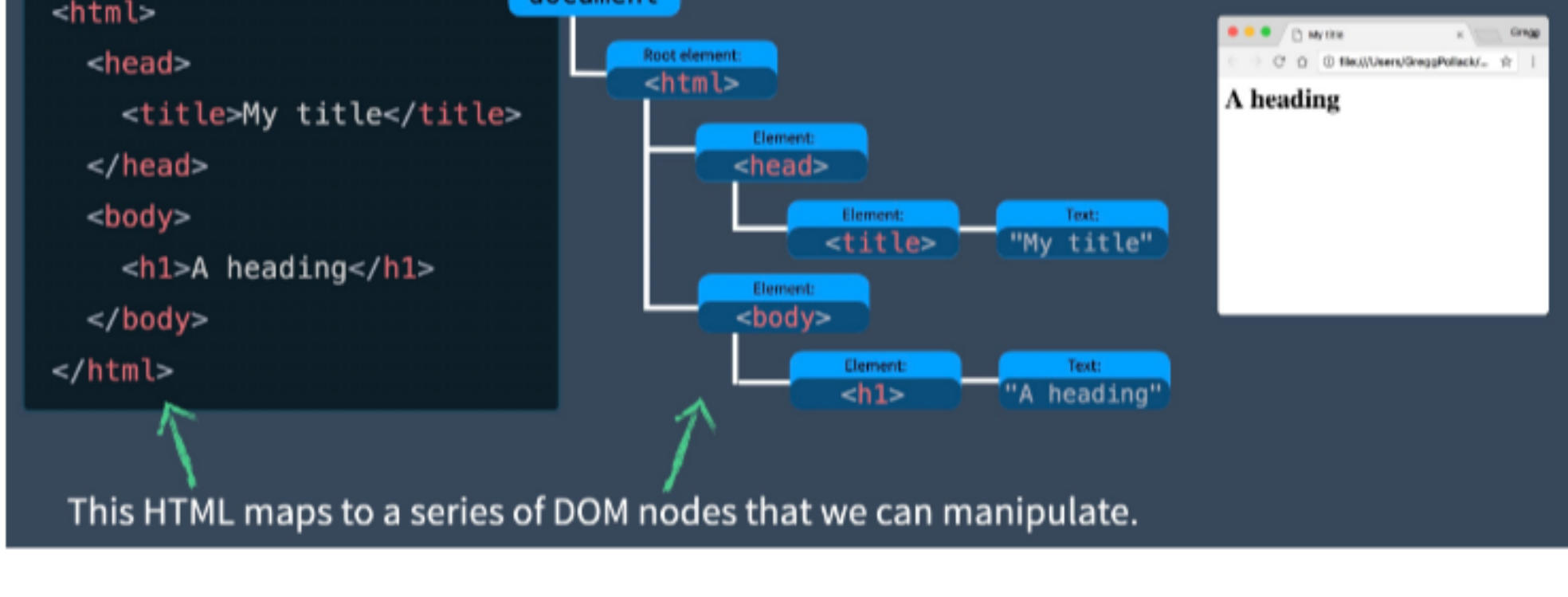


# Vue 3 Overview

This Vue 3 course with Evan You will help you learn how Vue is put together, so you can more effectively build and scale Vue applications. The majority of it is taught by Evan, but before I hand over the teaching to him, it's important that you understand some core Vue concepts.

## What is the DOM?

Dom stands for Document Object Model, and it's the our browser's interface (API) to change what is displayed in the browser.



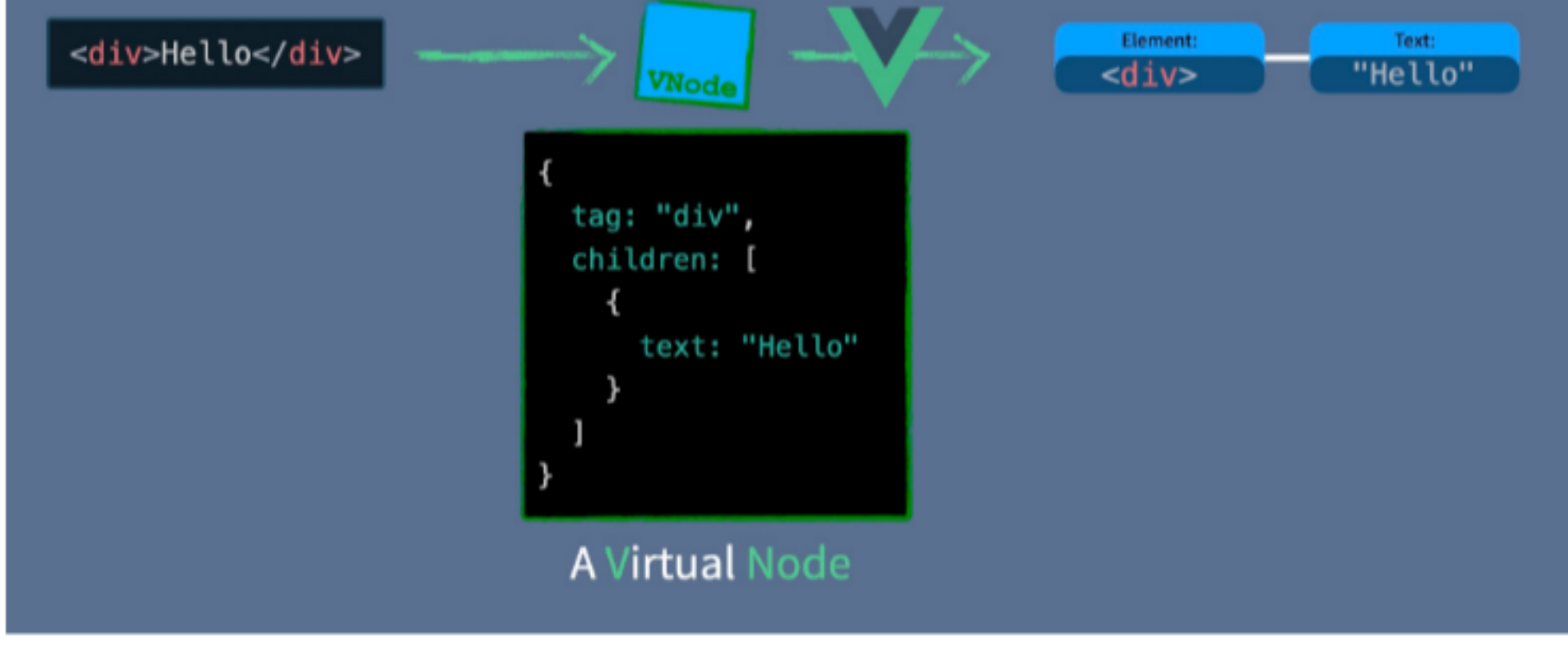
We use JavaScript to manipulate DOM, like so:

```
let item = document.getElementsByTagName("h1")[0];
item.textContent = "New Heading";
```

This would update our h1 heading.

## The Problem with the DOM

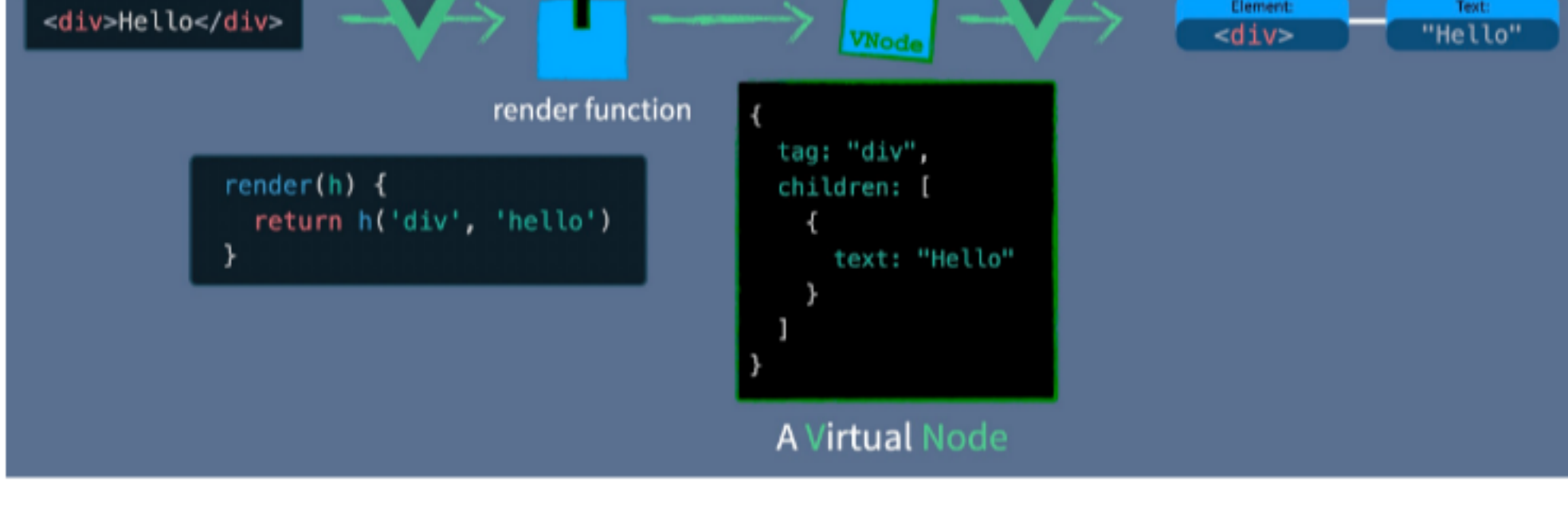
The underlying problem with using and manipulating the DOM is that some pages have thousands of nodes. This is why some frameworks (like Vue) have something called the **Virtual DOM**. The Virtual DOM is a way of representing the actual DOM with JavaScript Objects.



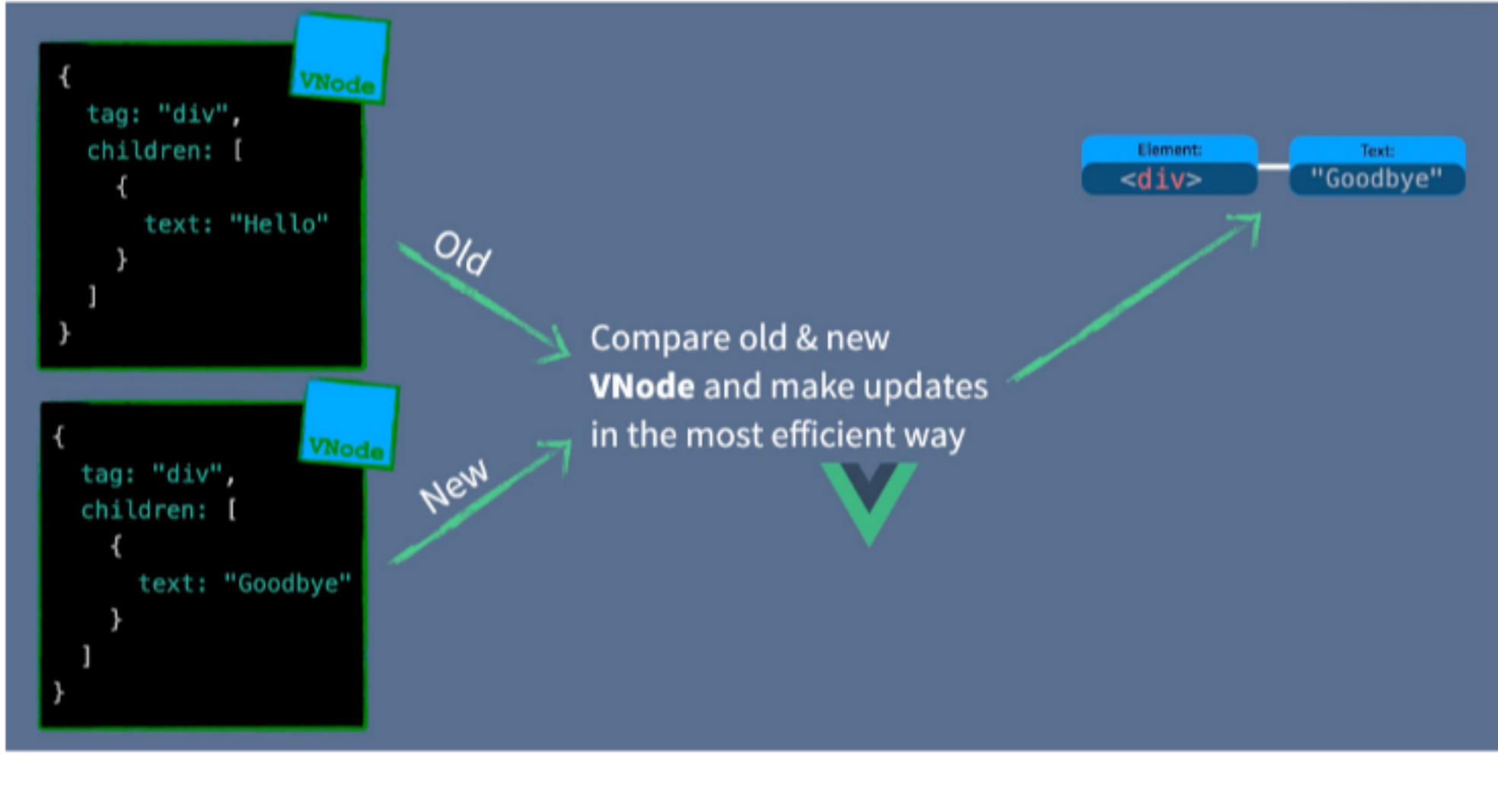
Above you can see how we can express a `div` as a JavaScript object. Vue then knows how to take this Virtual Node and make the appropriate JavaScript calls to create an actual DOM Node in the browser. There's actually an extra step in this lifecycle, called render functions.

## What is a Render Function?

When Vue receives a template, before creating a Virtual Node, it first compiles it into a render function. You can see this below:



The render function is what creates the virtual node, which gets sent to Vue to update the DOM. Later, if the data used by the render function changes, the render function will get run again producing a new Virtual DOM Node. Then Vue takes the old and the new node, compares the two, and makes the appropriate DOM calls to change the webpage.



## An Analogy of the Virtual DOM

A good way of understanding the Virtual DOM vs the Actual DOM, is blueprints (Virtual DOM) vs the actual building (Actual DOM). If I make changes to the 29th floor of the building, such as changing the layout of the furniture and adding new kitchen cabinets, I can make updates to the building in two ways:

1. I can demolish everything on the 29th floor and start from scratch.
2. I can create new blueprints, compare the differences, and make updates to the minimal amount of work.

Obviously, the second option is going to be faster.

## What is the Anatomy of Vue 3?

Vue 3's core engine is made up of multiple parts.

**Reactivity Module.** This allows for creating JavaScript reactive objects that can be watched for changes. When code which use these objects are run, they are tracked to be run later if the reactive object changes. If you haven't played through my [Vue 3 Reactivity course](#), I highly recommend you take it before moving on.

**Compiler Module.** This knows how to take HTML templates and compile them into render functions. This can happen in the browser at runtime, but more often happens when a Vue project is built, so that a browser only ever receives render functions.

**Renderer Module.** The renderer contains the code for 3 different phases of rendering out a component onto a webpage.

- **Render Phase.** When the render function is called and returns a representation of the actual DOM called the Virtual DOM. The Virtual DOM is a JavaScript object representation of what will be rendered into the browser.
- **Mount (or Create) Phase:** The renderer takes the Virtual DOM object and makes actual DOM JavaScript calls to create a webpage.
- **Patch (or Update) Phase:** The renderer takes the two Virtual DOM objects, an old and a new one, and update only the parts of the webpage that have changed using DOM JavaScript calls.

## Putting Vue 3 Together

Let's think about a simple component with a template which uses a reactive object. When Vue loads up:

1. The **Compiler Module** changes the HTML into a render function.
2. The reactive objects are initialized by the **Reactive Module**.
3. The **Renderer Module Render Phase** invokes the render function which creates a Virtual DOM object, or VNode. Since our render function references a reactive object, we track the render function to be run again if the reactive object changes.
4. In the **Mounting Phase** the **mount** function is called, using the VNode to create the webpage.
5. When the reactive object is changed, the **Patch Phase** invokes the render function again, and this time calls **patch** with the old VNode and the new VNode, which updates only the parts of the page which have changed.

With this knowledge you are all set to jump into the instruction with Evan in the next lesson!



# Virtual DOM & Render Functions

In this video Evan explains the benefits of the Virtual DOM and specifically some of the improvements made to it in Vue 3. He also covers render functions and how they've been improved as well.