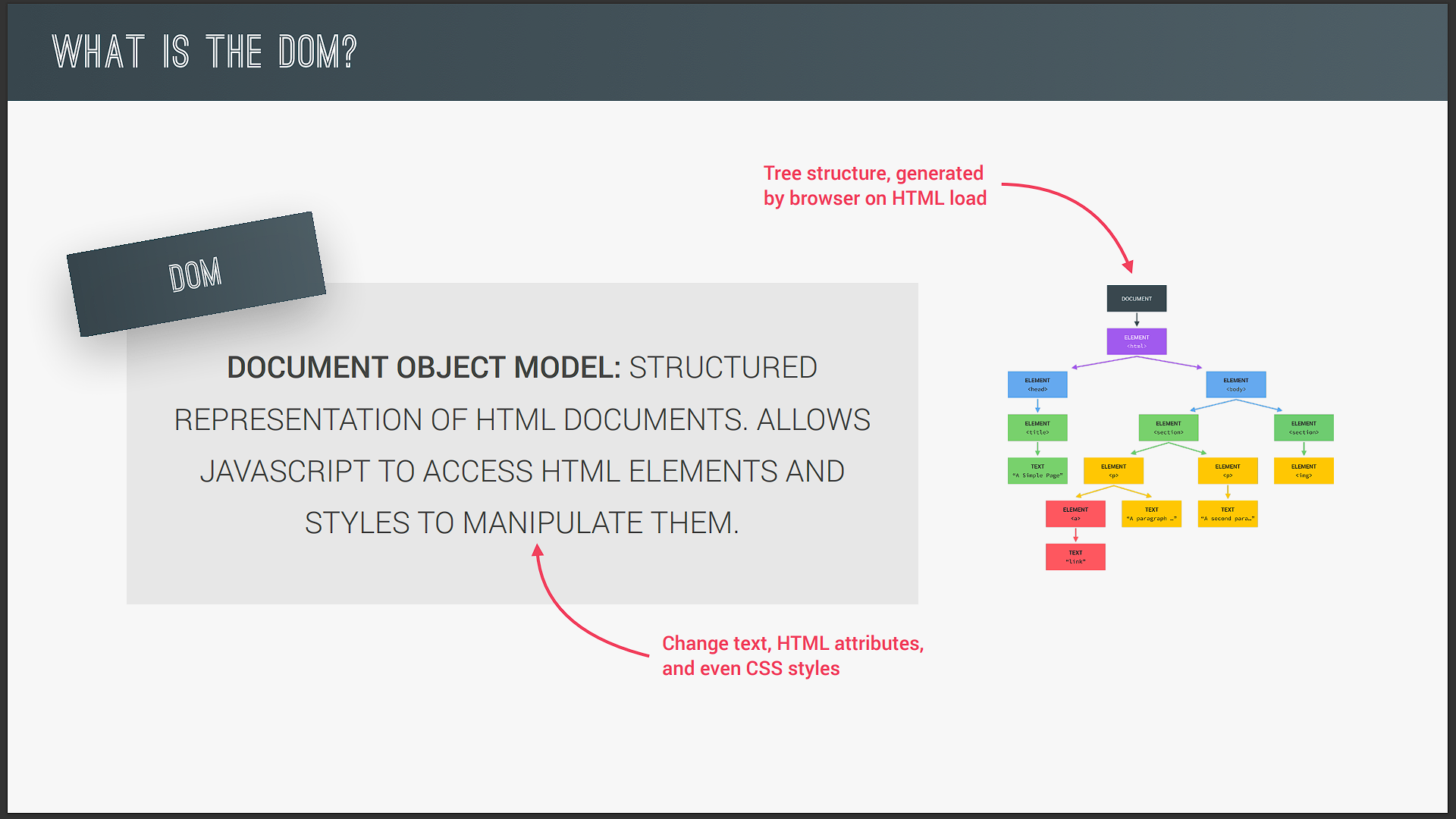
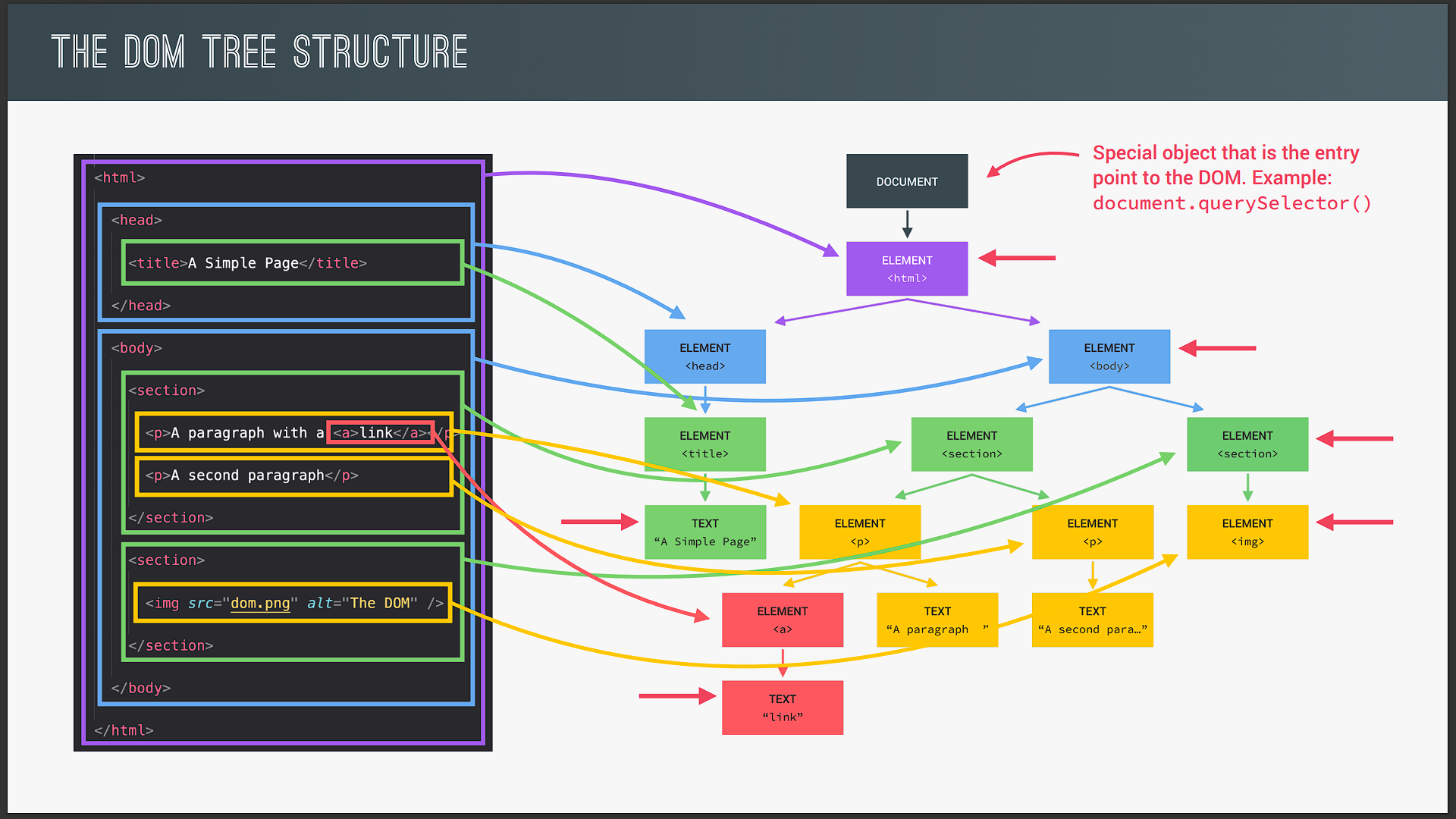
DOM stands for Document Object Model and it is basically a structured representation of HTML documents. The DOM allows us to use JavaScript to access HTML elements and styles in order to manipulate them. For example we will be able to change text to change HTML attributes and also to change CSS styles from our JavaScript. So we can say that DOM is basically a connection point between HTML documents and JavaScript code. Now the DOM is automatically created by the browser as soon as the HTML page loads. And it's stored in a tree structure like this one. In this tree each HTML element is one object. And so let's now take a look at this DOM structure in a little bit more detail.



And to illustrate this here is a very simple HTML document and here is what a DOM tree corresponding to this HTML looks like. So as I already mentioned this is a tree structure which looks a bit like a family tree right? And actually we used the terms child element parent element sibling element and so on when we talk about the DOM tree and DOM manipulation. Anyway as you can see for each element in the HTML there is one element node and the DOM tree and we can access and interact with each of these nodes using JavaScript.

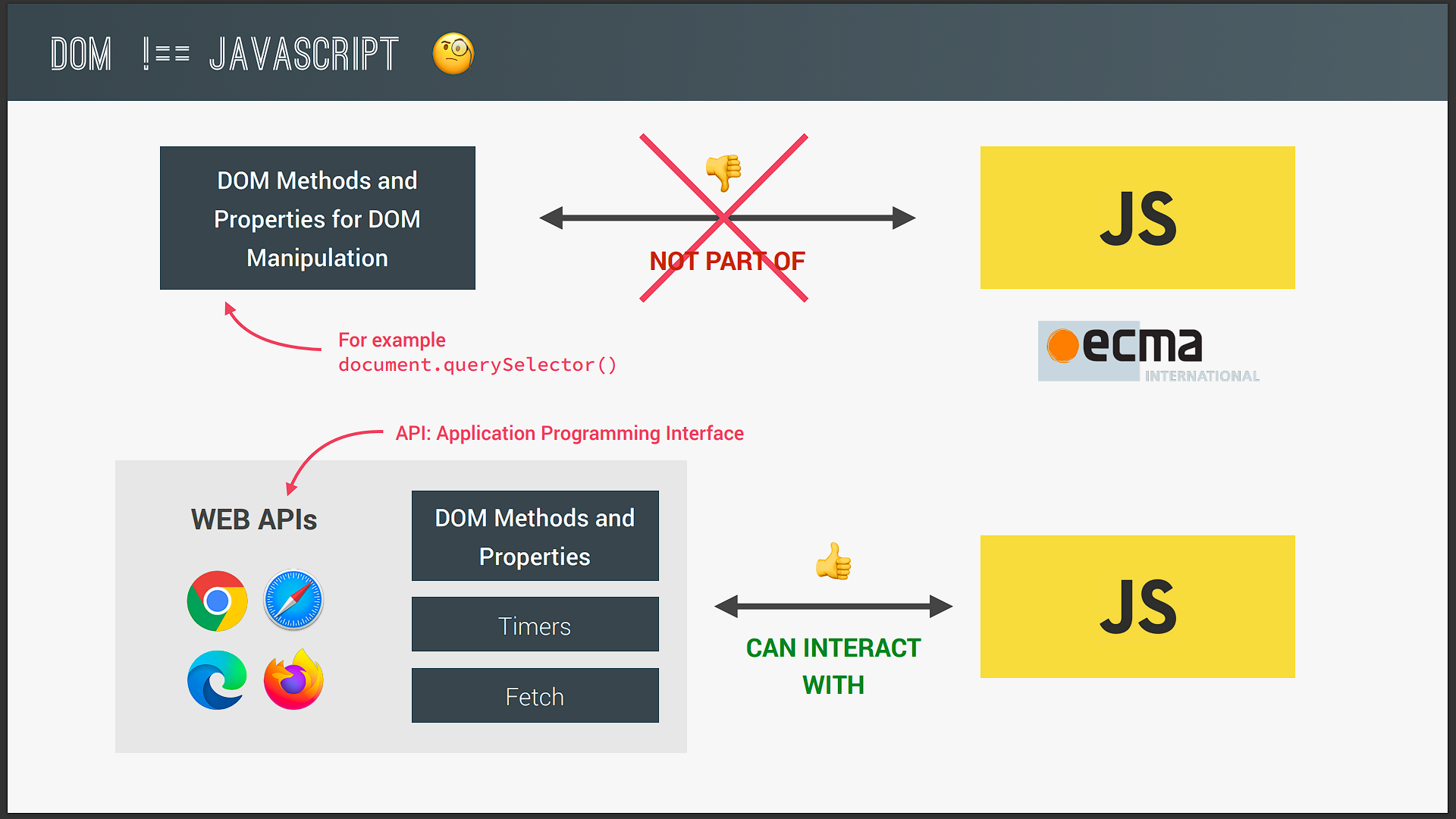


Okay so the DOM always starts with the document object right at the very top. And document is a special object that we have access to in JavaScript. And this object serves as an entry point into the DOM. Remember how you used document dot query selector in the last lecture to select an element. So that means that the query selector method is available on the document object. And so that's why we say that document is the entry point to the DOM because we need it to start selecting elements.

All right then the first child element of document is usually the HTML element because that's usually the root element in all HTML documents. Next HTML usually has two child elements head and body. And so of course you can also find them here in the DOM tree. In the HTML they are adjacent elements and so they are siblings in our DOM as well. Then as we keep going deeper into the nested HTML structure we keep adding more and more children to the DOM tree. So inside head and body you have more child elements and the two sections and the body even have child elements themselves. And from there it goes even deeper because the first paragraph also has a child which is this link element here. And with that finally we have all our HTML elements in the DOM tree.

But a Dom tree actually has more than just element nodes. It also has nodes for all the text itself comments and other stuff because basically the rule is that whatever is in the HTML document also has to be in the DOM. And so as you see the DOM really is a complete representation of the HTML document so that we can manipulate it in complex ways. And with this you should now have a good overview of how the DOM works and what it looks like. But before we finish I need to clarify something that many beginners find confusing. So many people believe that the DOM and all the methods and properties that we can use to manipulate the DOM such as documented or the query selector and lots of other stuff are actually part of JavaScript. However this is not the case. Remember that JavaScript is actually just a dialect of the ECMAScript specification and all this DOM related stuff is simply not in there. So up until this point we have only used the JavaScript language itself.

But starting in this section we will also use JavaScript in a browser. I mean sure we have used Google Chrome to run our code in the developer console but that's not what I mean here. What I mean is to manipulate webpages that are actually displayed and rendered in the browser just like any normal website that you know. Okay. But now you might ask "If the DOM is not a part of the JavaScript language then how does this all work?" Well the DOM and DOM methods line% are actually part of something called the web APIs. line% So the web API APIs are like libraries that browsers implement line% and that we can access from our JavaScript code. And API stands for Application Programming Interface. But more about that later.



For now what you need to know is that a web APIs are basically libraries that are also written in JavaScript and that are automatically available for us to use. So all this happens behind the scenes we don't have to import or do anything okay? And there is actually an official DOM specification that browsers implement which is the reason why DOM manipulation works the same in all browsers. Now besides the DOM there are actually a ton more web APIs such as timers the fetch API and many more. Again we will learn a lot more about this later on. For now let's finally start our project and do some DOM manipulation in practice.