**Day 35**



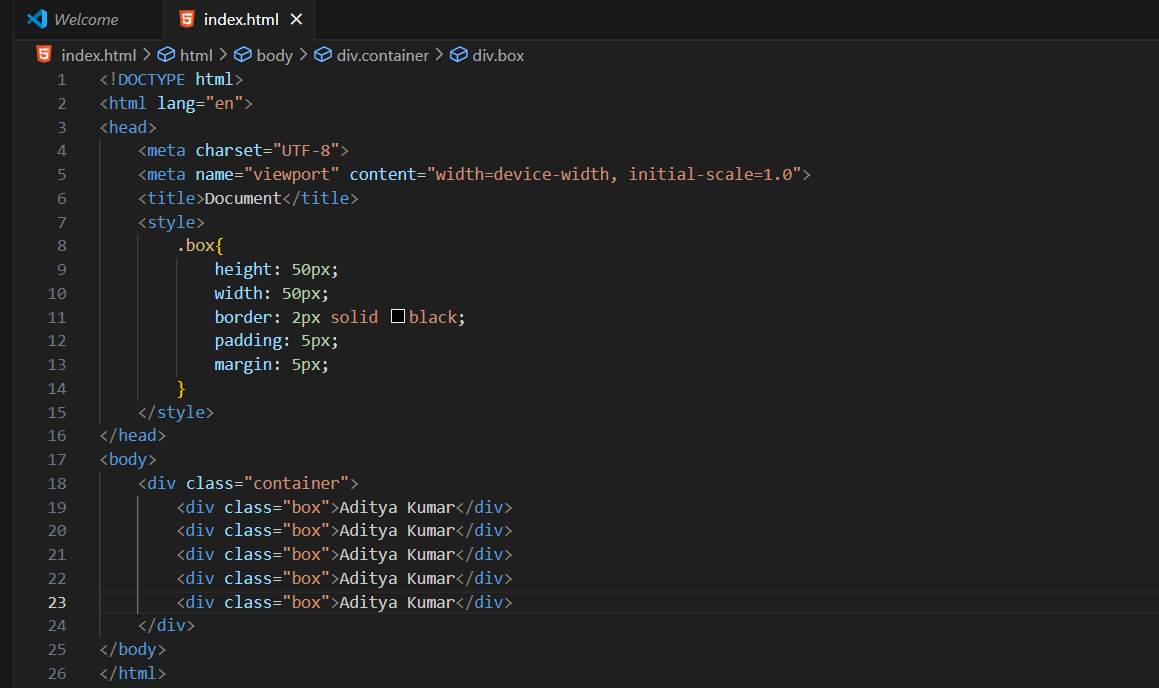


**“Web Development + Security”**

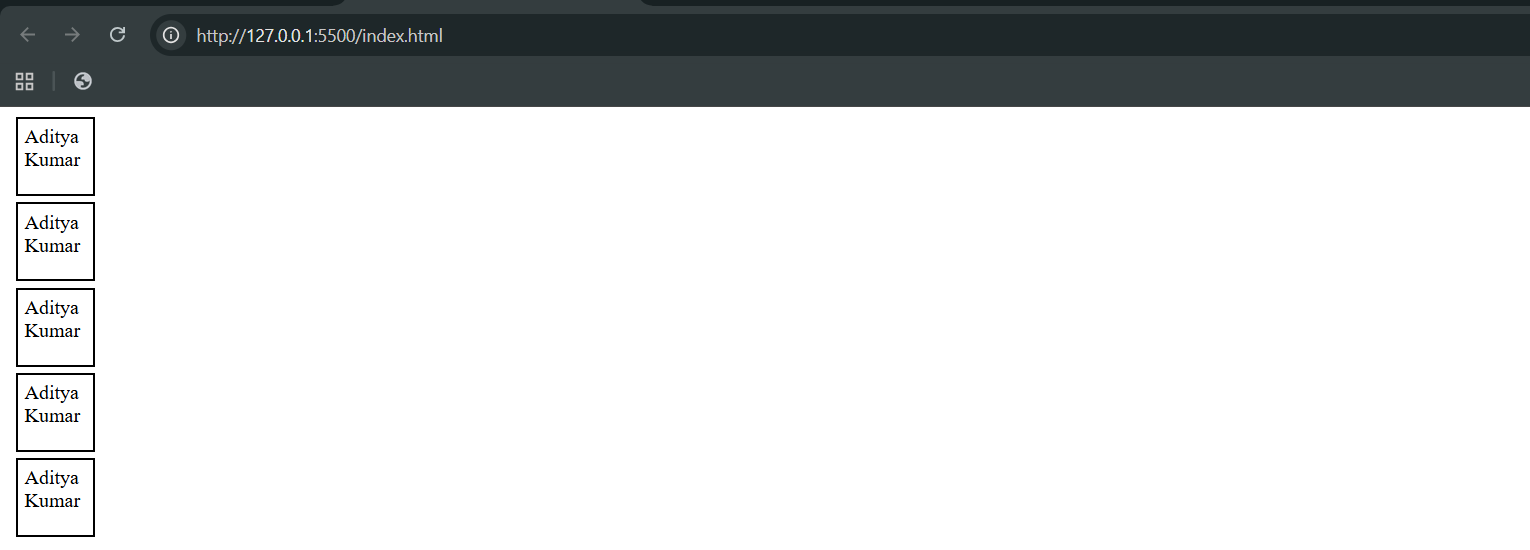
**JavaScript Selecting by Ids, Classes, and More:**

Creating a very basic page to understand from basics: without any DOM

Code:

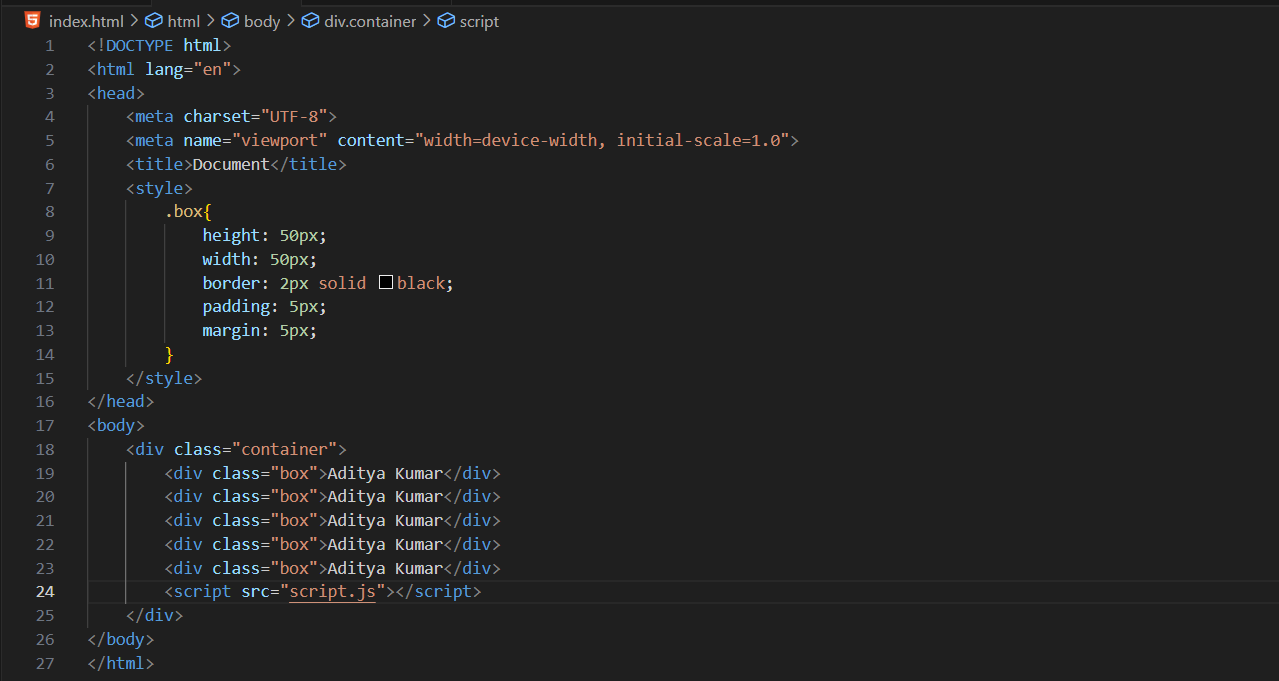


Output:

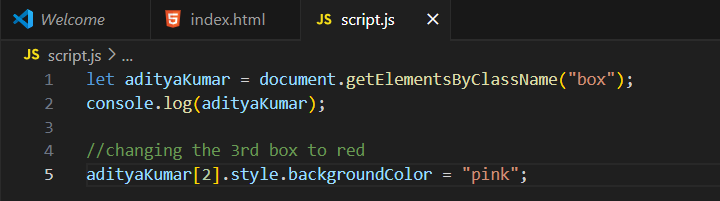


Now, I want to change the bg-color of the 3rd box using the class name, then for it we will include the .js in the .html, as shown below:

Index.html:



Script.js:



Output:

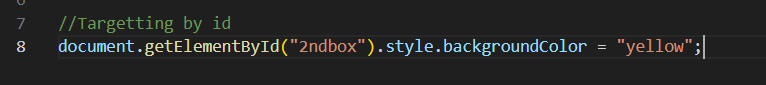


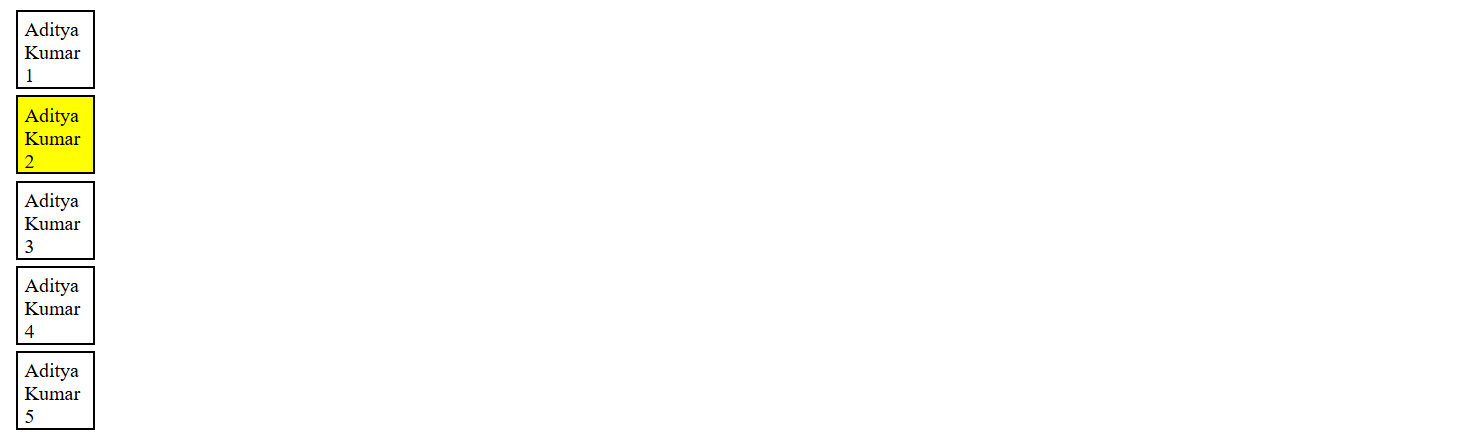
Now, targeting by the id name:

Index.html:



Script.js:

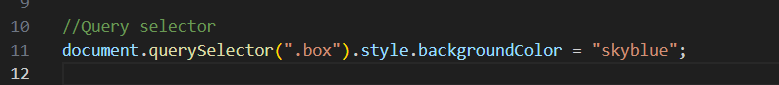


Output:  


Now, using the query selector: we can target the 1st class

Index.html: same as above

Script.js:

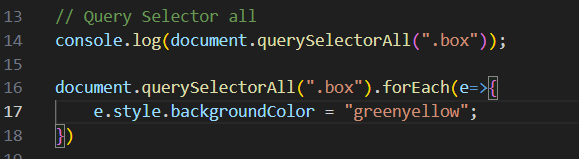


Output: 

Now, queryselectorAll is bit differnet, it allows us to target all the classes which matched the name .box, but since it returns the Nodelist, we shall have to loop to change the properties of them:

Index.html: same

Script.js:



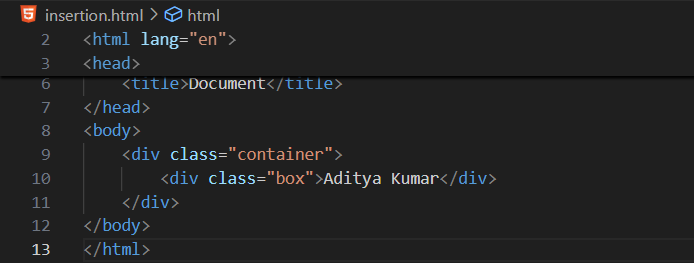
Output:



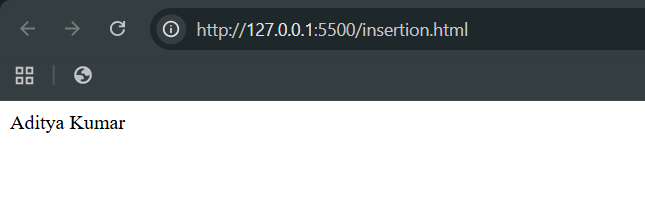
**Inserting and Removing Elements using JavaScript:**

Basic code, no <style> or <script>:

Code:

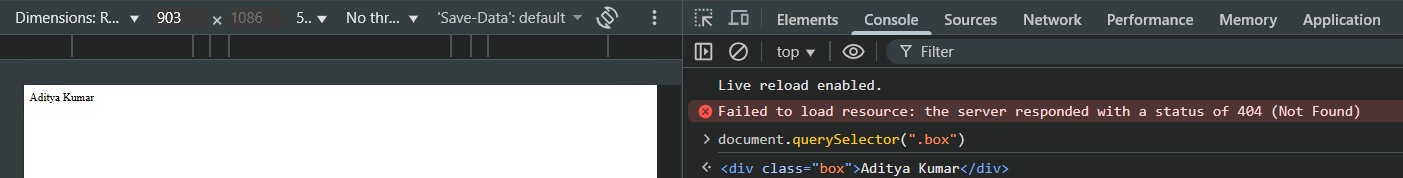


Output:

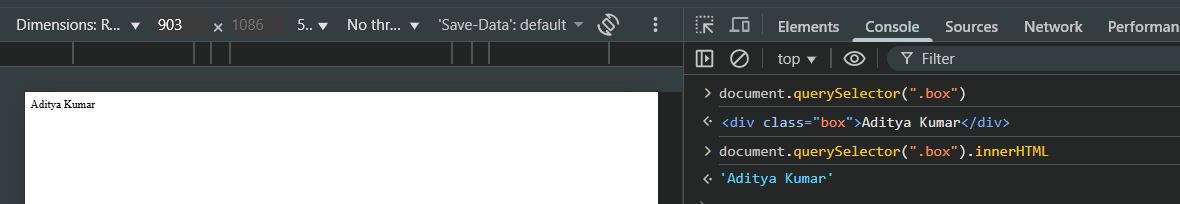


Now, we will be using JavaScript DOM in the browser:

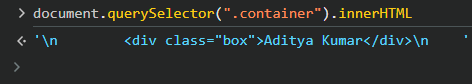
First, we will be using the query selector to get the first <div> whose class is box:



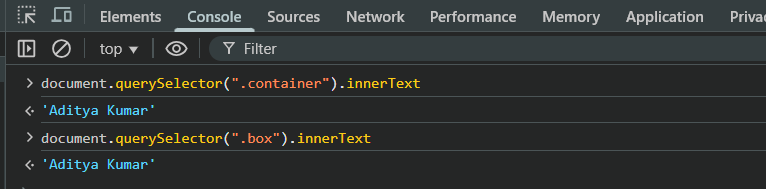
Now, we will be using .innerHTML in order to get the content of that <div>:



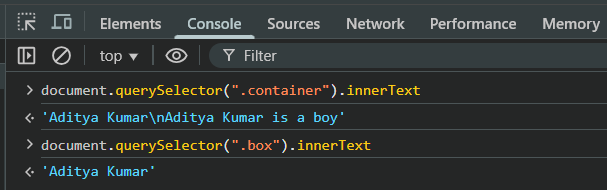
Similarly, we can get the inner content of .container class:

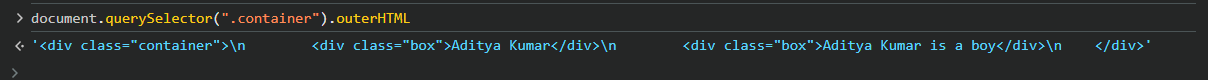


While, if we just want the text which is there inside the container or box class, we will use .innerText:

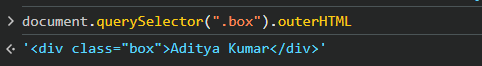


In case we have two <div> inside the .container class, we will see something like this:

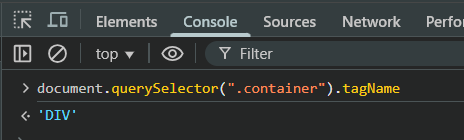


Now, If we uses the .outerHTML, then we will get the container itself and the HTML inside that class:  


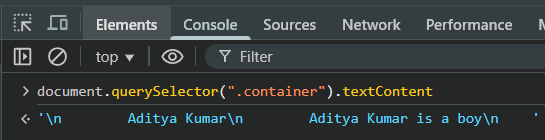
Similarly,



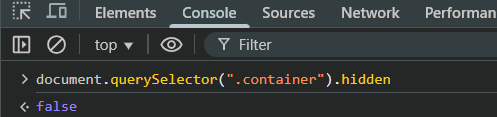
Now, to get the tag name of a class, we will use, .tagName:



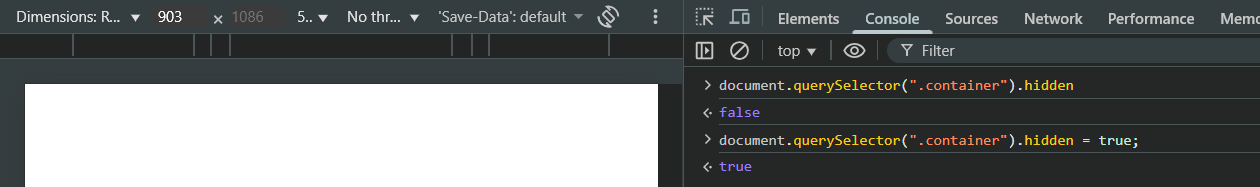
Now, using the .textContent we can get the text content which be there inside the container class:



Now, to check if that element is hidden or visible: we uses .hidden. Clearly, it returned false, it means that class whose name is container is not hidden

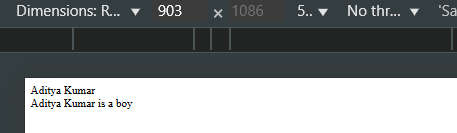


We can actually change it, and can hide it as well: clearly, the content inside it get removed.

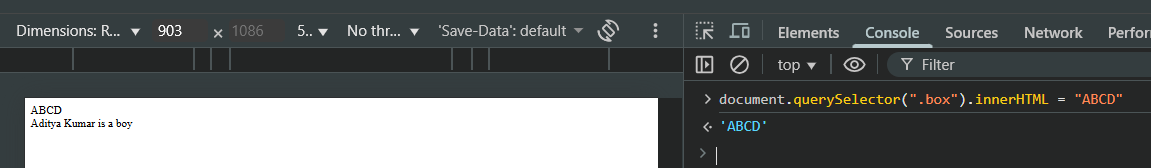


Now, we can also edit the content of the first <div> whose class is .box: we will use .innerHTML

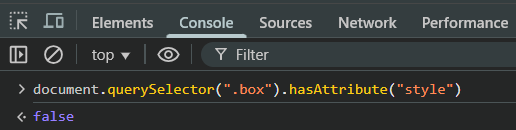
Before:



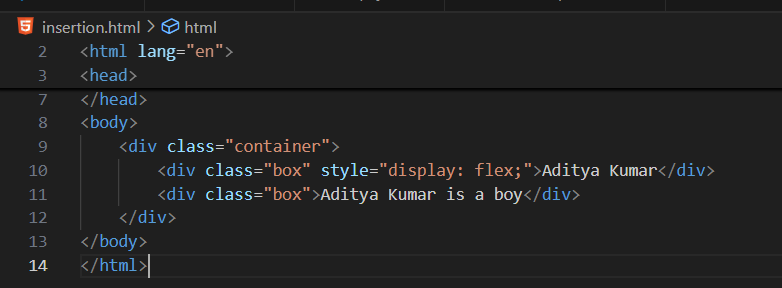
After:



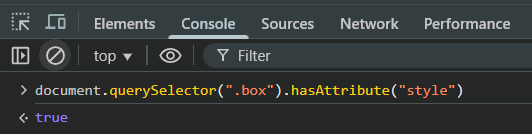
We can also check if a certain class has a certain attribute or not: using .hasAttribute



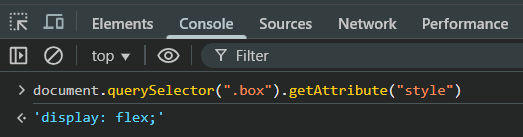
Now, we have added the style attribute to the first <div> whose class is box:



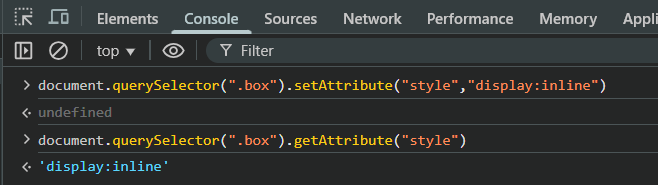
At console:



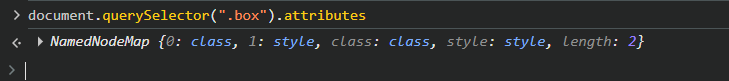
Now, in case the attribute be present, then we can also see what attribute is there actually: using the getAttribute:



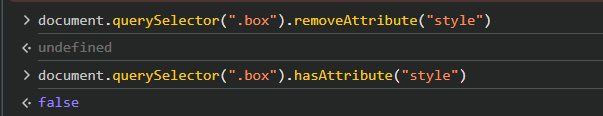
Now, we can also change the attribute value using the .setAttribute: we just changed the attribute form flex to inline.



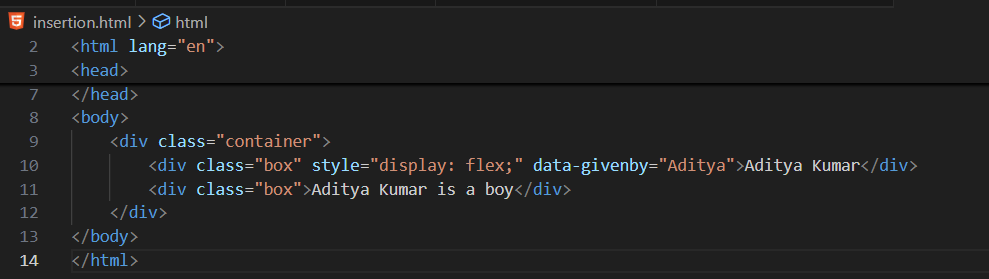
Now, using the .attributes we can get all the attributes of an element whose class is box:



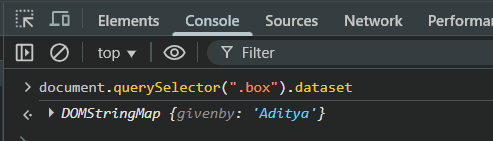
Now, we can also remove the attributes: using the .removeAttribute



Moreover, we can store information using the data-attribute:



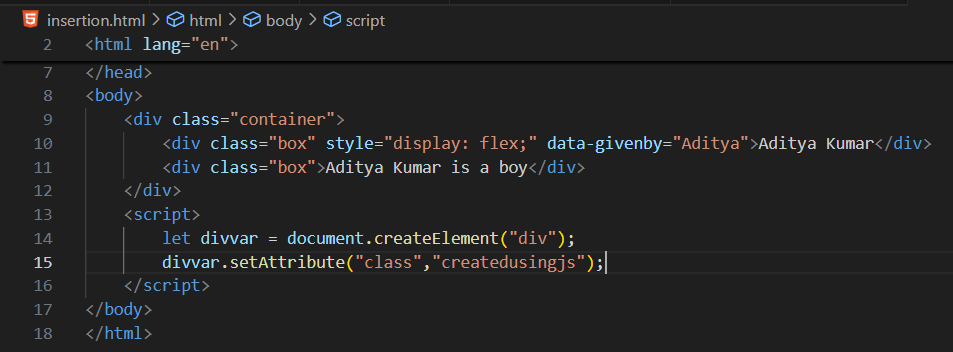
If in case you want to access them: then we will use .dataset



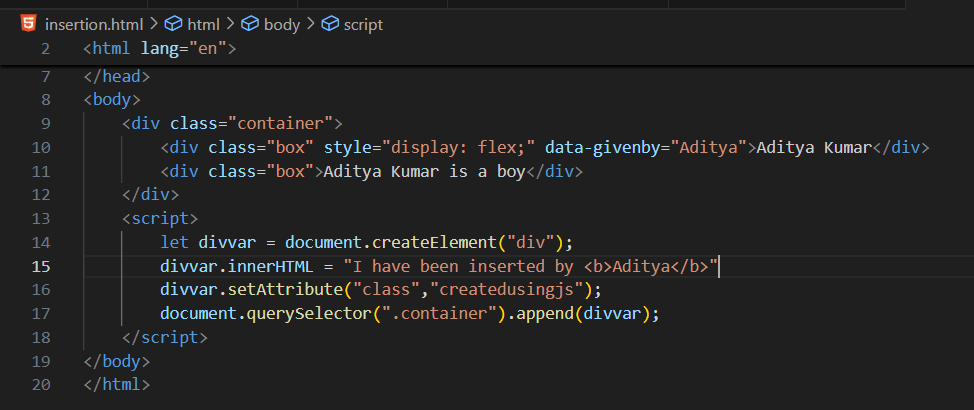
Now, in case we want to create the element using the JavaScript:



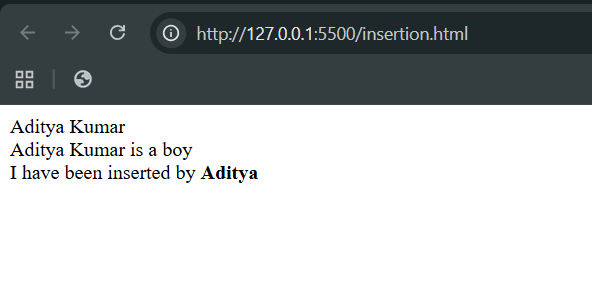
Now, in case we want to set attribute for this: here we are creating the class named createdusingjs



Now, suppose we want to append this new <div> create using JS to the .container class: we will use querySelector as well as .append to do so:

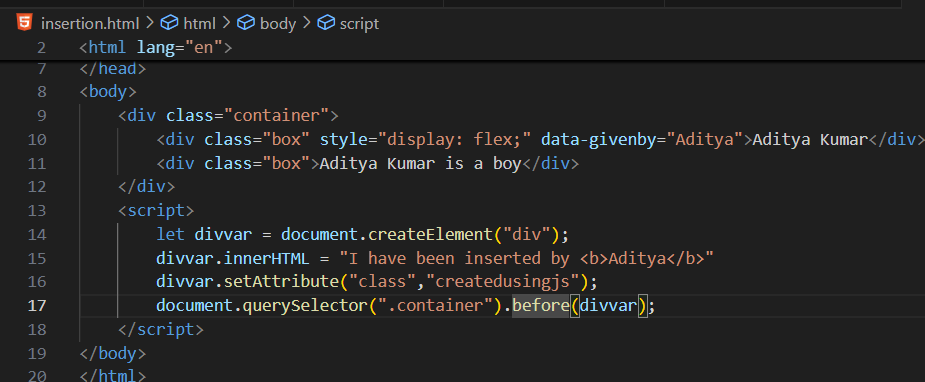
Code: 

Output:

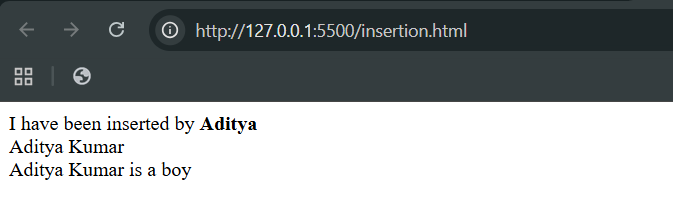


Similarly, we have .before and .after:

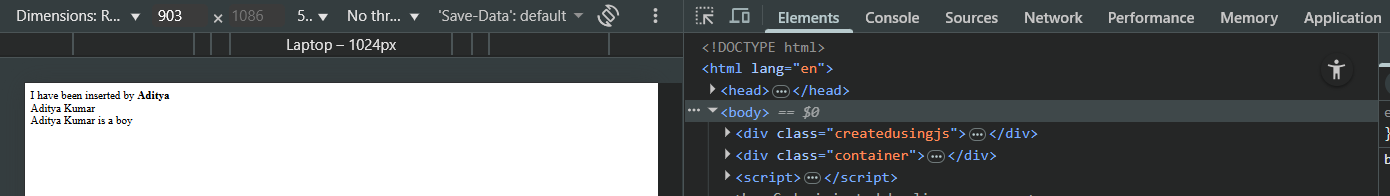
Code:



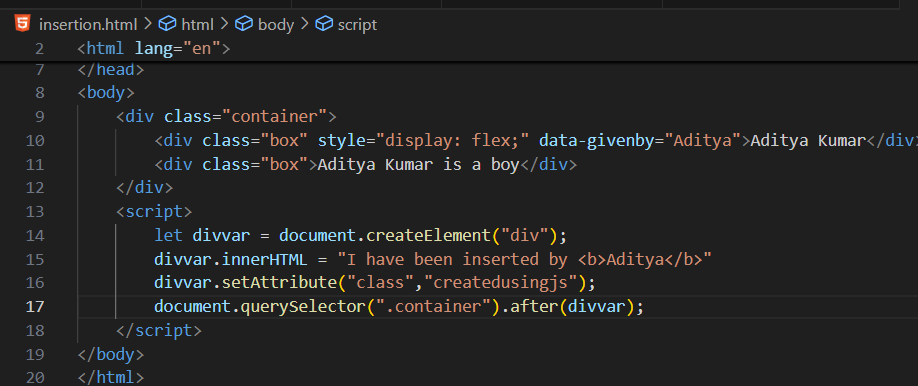
Output:



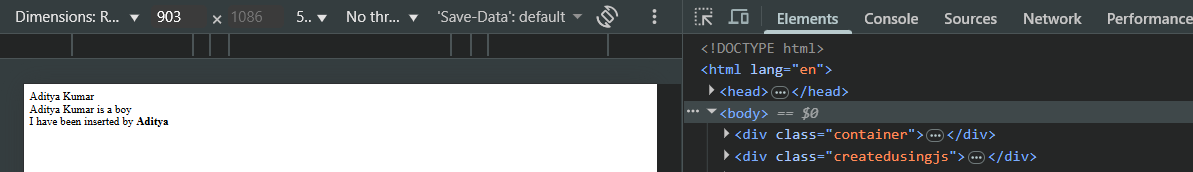
For clarity:



Similarly, we can use the .after:



Output:

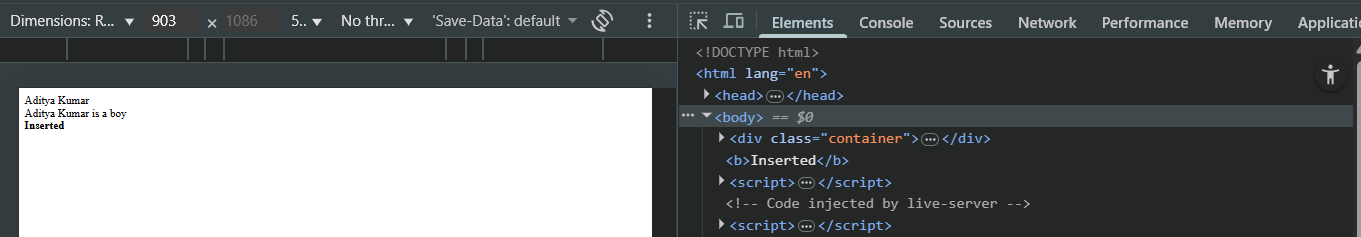


Now, suppose we want to insert an HTML code: we will use insertAdjacentHTML

Code: using afterend



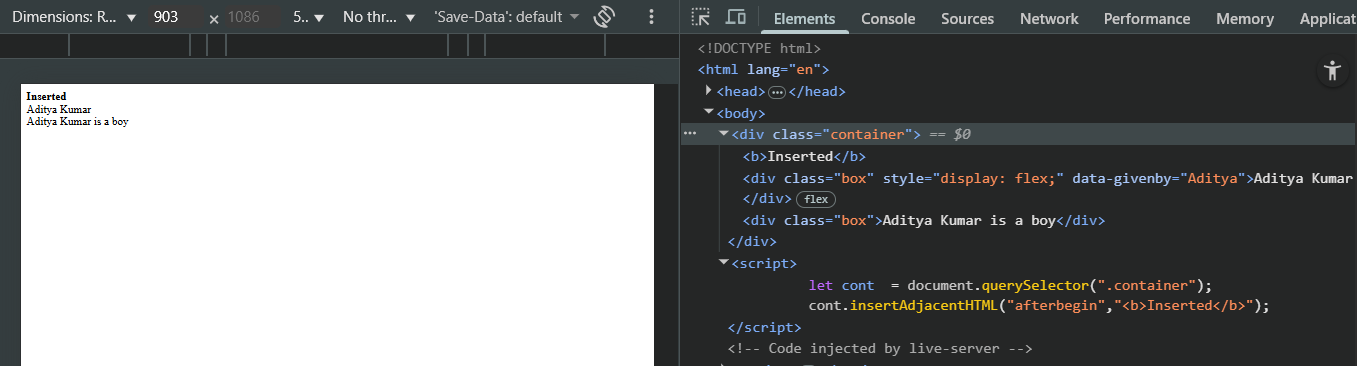
Output:



Code: using afterbegin



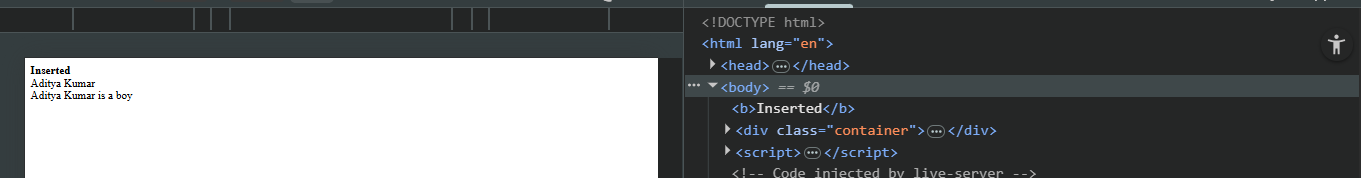
Output:



Code: using beforebegin

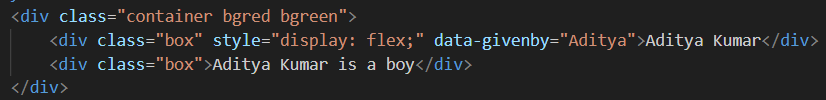


Output:

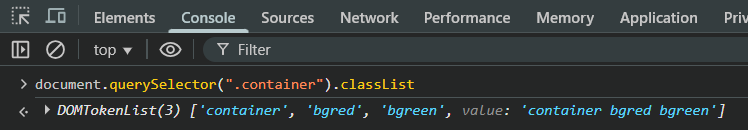


Now, to get the class list, we will use .classList:

Code:



Console:



Now, to get the class name, we will use .className:

Code: same as above

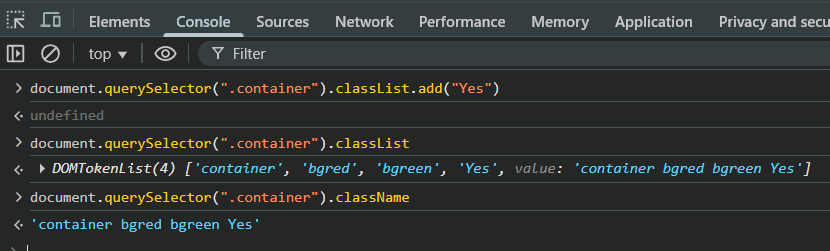
Console:



Now, in case we want to add a class from our end: then we will use .classList as well as .add :

Code: same as above

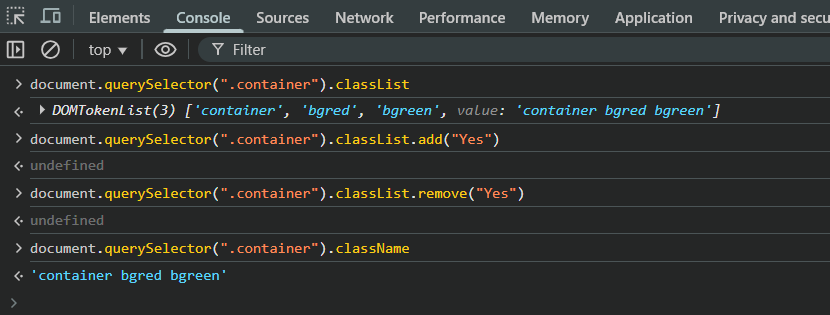
Console:



Now, in case we want to remove a class from our end: then we will use .classList as well as .remove :

Code: same as above

Console:



--The End--