

Day 35

“Web Development + Security”

JavaScript Selecting by Ids, Classes, and More:

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

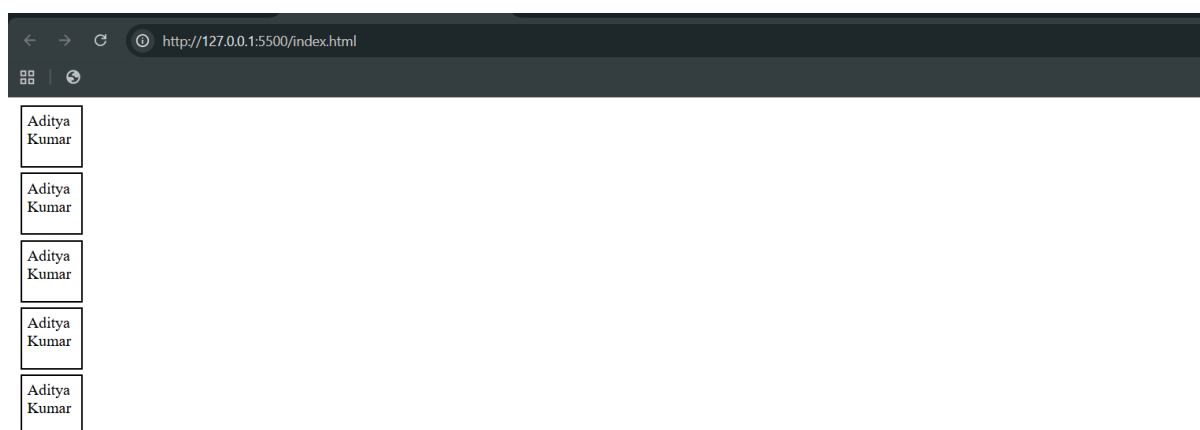
Code:

```

Welcome index.html x
index.html > html > body > div.container > div.box
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     .box{
9       height: 50px;
10      width: 50px;
11      border: 2px solid black;
12      padding: 5px;
13      margin: 5px;
14    }
15  </style>
16 </head>
17 <body>
18   <div class="container">
19     <div class="box">Aditya Kumar</div>
20     <div class="box">Aditya Kumar</div>
21     <div class="box">Aditya Kumar</div>
22     <div class="box">Aditya Kumar</div>
23     <div class="box">Aditya Kumar</div>
24   </div>
25 </body>
26 </html>

```

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:

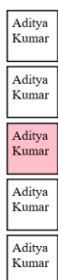
Index.html:

```
index.html > html > body > div.container > script
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Document</title>
7    <style>
8      .box{
9        height: 50px;
10       width: 50px;
11       border: 2px solid black;
12       padding: 5px;
13       margin: 5px;
14     }
15   </style>
16 </head>
17 <body>
18   <div class="container">
19     <div class="box">Aditya Kumar</div>
20     <div class="box">Aditya Kumar</div>
21     <div class="box">Aditya Kumar</div>
22     <div class="box">Aditya Kumar</div>
23     <div class="box">Aditya Kumar</div>
24     <script src="script.js"></script>
25   </div>
26 </body>
27 </html>
```

Script.js:

```
Welcome  index.html  JS script.js  X
JS script.js > ...
1  let adityaKumar = document.getElementsByClassName("box");
2  console.log(adityaKumar);
3
4  //changing the 3rd box to red
5  adityaKumar[2].style.backgroundColor = "pink";
```

Output:



Now, targeting by the id name:

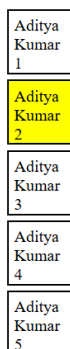
Index.html:

```
index.html > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     .box{
9       height: 50px;
10      width: 50px;
11      border: 2px solid black;
12      padding: 5px;
13      margin: 5px;
14    }
15  </style>
16 </head>
17 <body>
18   <div class="container">
19     <div class="box">Aditya Kumar 1</div>
20     <div id="2ndbox" class="box">Aditya Kumar 2</div>
21     <div class="box">Aditya Kumar 3</div>
22     <div class="box">Aditya Kumar 4</div>
23     <div class="box">Aditya Kumar 5</div>
24     <script src="script.js"></script>
25   </div>
26 </body>
27 </html>
```

Script.js:

```
7 //Targetting by id
8 document.getElementById("2ndbox").style.backgroundColor = "yellow";
```

Output:



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

Index.html: same as above

Script.js:

```
9
10 //Query selector
11 document.querySelector(".box").style.backgroundColor = "skyblue";
12
```

Output:

Aditya Kumar 1
Aditya Kumar 2
Aditya Kumar 3
Aditya Kumar 4
Aditya Kumar 5

Now, `querySelectorAll` is bit different, 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:

```
13 // Query Selector all
14 console.log(document.querySelectorAll(".box"));
15
16 document.querySelectorAll(".box").forEach(e=>{
17     e.style.backgroundColor = "greenyellow";
18 })
```

Output:

Aditya Kumar 1
Aditya Kumar 2
Aditya Kumar 3
Aditya Kumar 4
Aditya Kumar 5

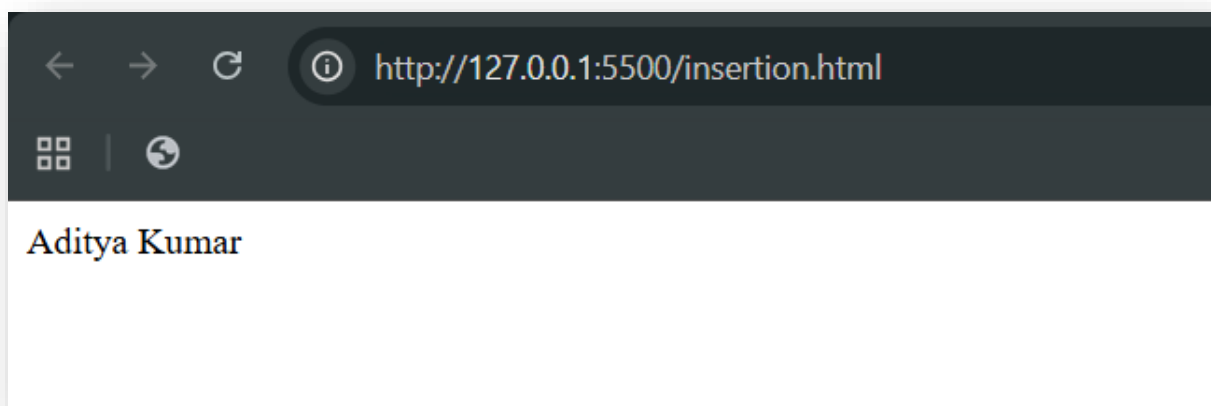
Inserting and Removing Elements using JavaScript:

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

Code:

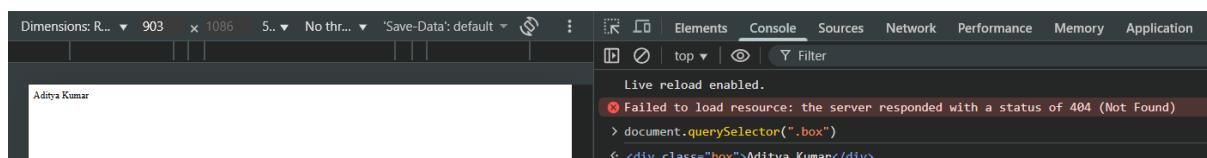
```
insertion.html > html
2  <html lang="en">
3  <head>
6  |   <title>Document</title>
7  </head>
8  <body>
9  |   <div class="container">
10 |     <div class="box">Aditya Kumar</div>
11 |   </div>
12 </body>
13 </html>
```

Output:

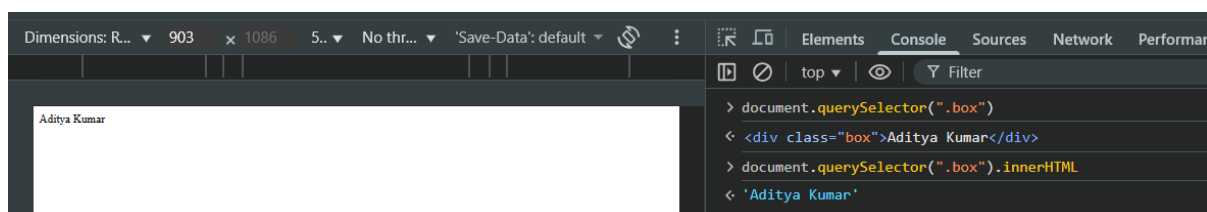


Now, we will be using JavaScript DOM in the browser:

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



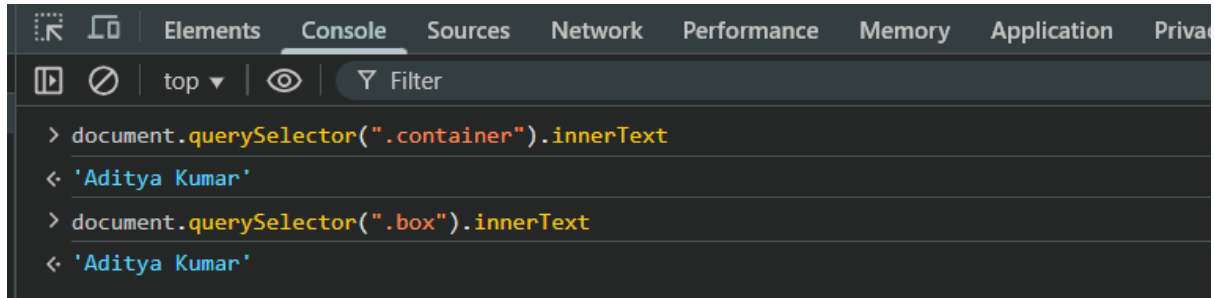
Now, we will be using .innerHTML in order to get the content of that <div>:



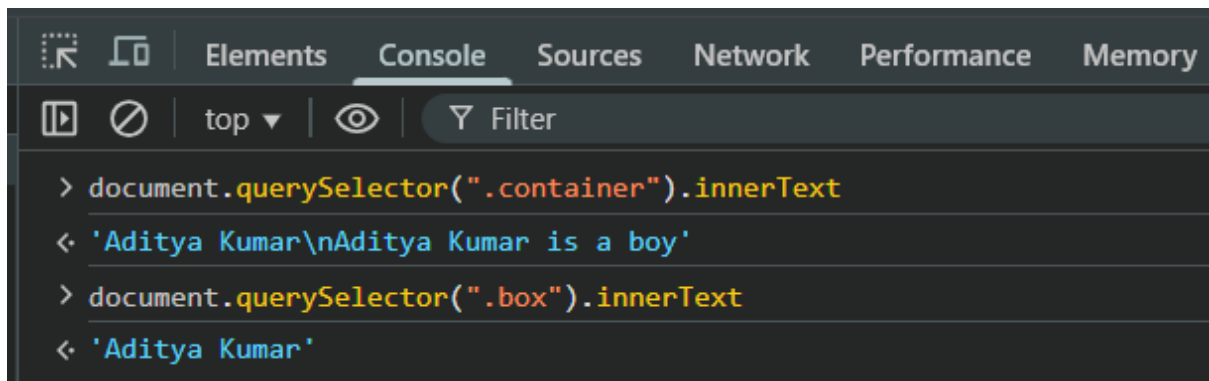
Similarly, we can get the inner content of .container class:

```
> document.querySelector(".container").innerHTML
< '\n      <div class="box">Aditya Kumar</div>\n      '
```

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:



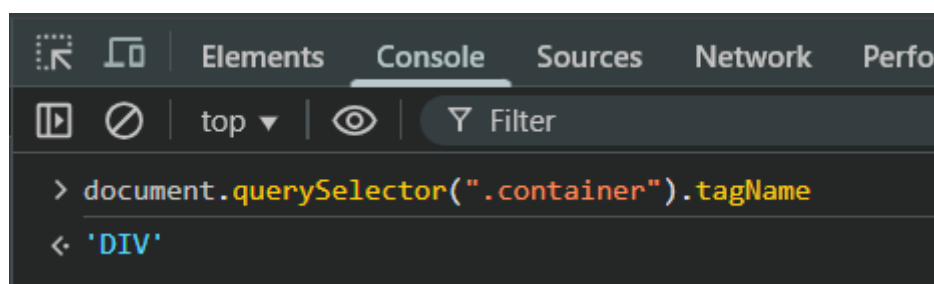
Now, If we uses the .outerHTML, then we will get the container itself and the HTML inside that class:

```
> document.querySelector(".container").outerHTML
< '<div class="container">\n      <div class="box">Aditya Kumar</div>\n      <div class="box">Aditya Kumar is a boy</div>\n      </div>'
```

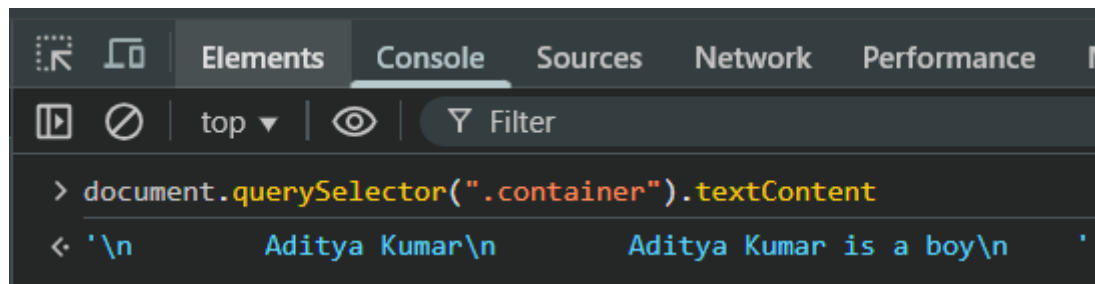
Similarly,

```
> document.querySelector(".box").outerHTML
< '<div class="box">Aditya Kumar</div>'
```

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

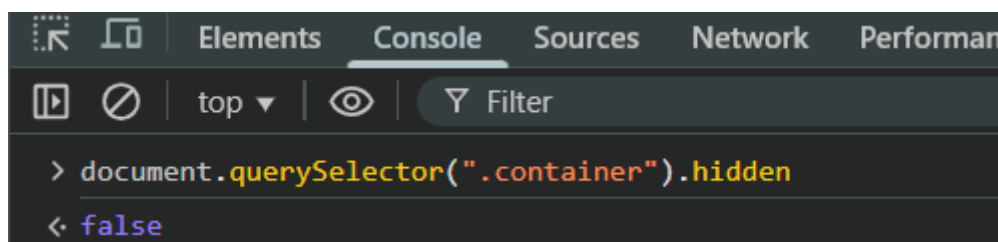


Now, using the `.textContent` we can get the text content which be there inside the container class:



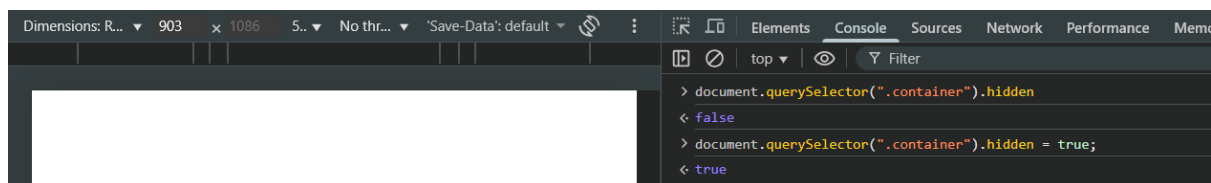
```
> document.querySelector(".container").textContent
< '\n      Aditya Kumar\n      Aditya Kumar is a boy\n    '
```

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



```
> document.querySelector(".container").hidden
< false
```

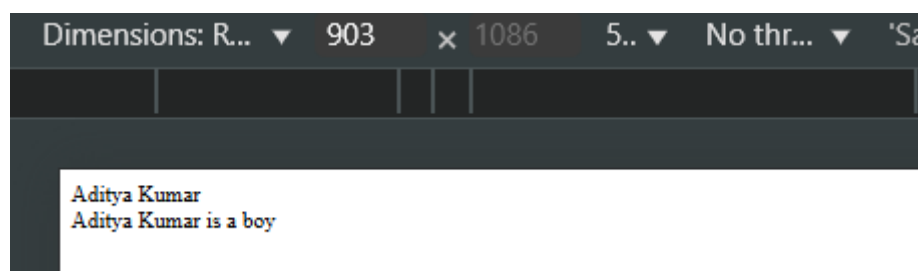
We can actually change it, and can hide it as well: clearly, the content inside it get removed.



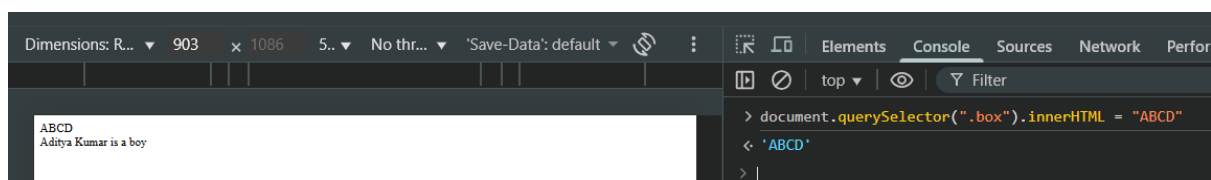
```
> document.querySelector(".container").hidden
< false
> document.querySelector(".container").hidden = true;
< true
```

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

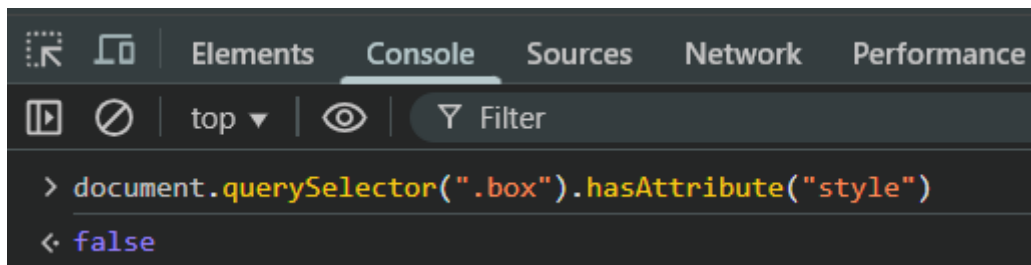
Before:



After:



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



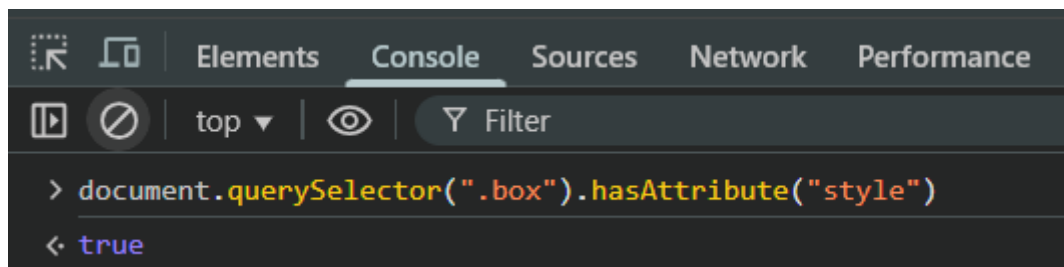
```
document.querySelector(".box").hasAttribute("style")
false
```

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



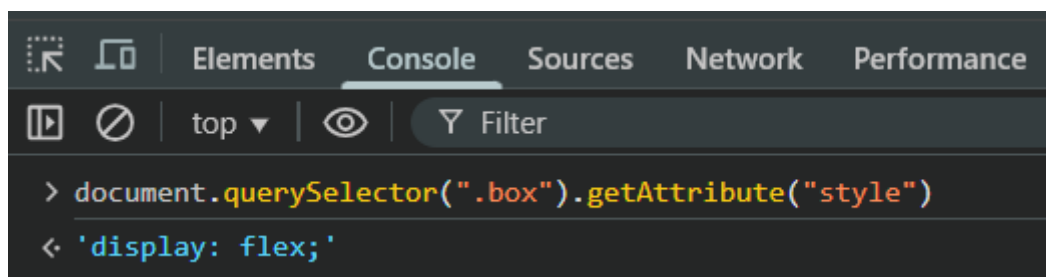
```
insertion.html > html
2 <html lang="en">
3 <head>
7 </head>
8 <body>
9   <div class="container">
10     <div class="box" style="display: flex;">Aditya Kumar</div>
11     <div class="box">Aditya Kumar is a boy</div>
12   </div>
13 </body>
14 </html>
```

At console:



```
document.querySelector(".box").hasAttribute("style")
true
```

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



```
document.querySelector(".box").getAttribute("style")
'display: flex;'
```


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

```
Elements Console Sources Network Performance Memory Appli
top | Filter
> document.querySelector(".box").setAttribute("style","display:inline")
< undefined
> document.querySelector(".box").getAttribute("style")
< 'display:inline'
```

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

```
> document.querySelector(".box").attributes
< ▶ NamedNodeMap {0: class, 1: style, class: class, style: style, length: 2}
> |
```

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

```
> document.querySelector(".box").removeAttribute("style")
< undefined
> document.querySelector(".box").hasAttribute("style")
< false
```

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

```
insertion.html > html
2 <html lang="en">
3 <head>
7 </head>
8 <body>
9   <div class="container">
10     <div class="box" style="display: flex;" data-givenby="Aditya">Aditya Kumar</div>
11     <div class="box">Aditya Kumar is a boy</div>
12   </div>
13 </body>
14 </html>
```

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

```
Elements Console Sources Network Performan
top | Filter
> document.querySelector(".box").dataset
< ▶ DOMStringMap {givenby: 'Aditya'}
```

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

```
insertion.html > html > body
2  <html lang="en">

8  <body>
9    <div class="container">
10     <div class="box" style="display: flex;" data-givenby="Aditya">Aditya Kumar</div>
11     <div class="box">Aditya Kumar is a boy</div>
12   </div>
13   <script>
14     let divvar = document.createElement("div");
15   </script>
16 </body>
17 </html>
```

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

```
insertion.html > html > body > script
2  <html lang="en">

7  </head>
8  <body>
9    <div class="container">
10     <div class="box" style="display: flex;" data-givenby="Aditya">Aditya Kumar</div>
11     <div class="box">Aditya Kumar is a boy</div>
12   </div>
13   <script>
14     let divvar = document.createElement("div");
15     divvar.setAttribute("class","createdusingjs");
16   </script>
17 </body>
18 </html>
```

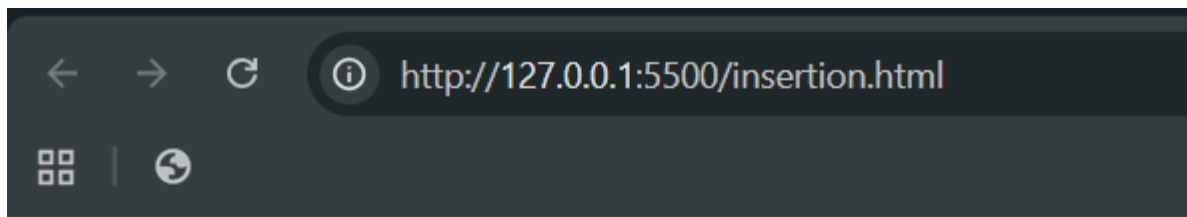
Now, suppose we want to append this new <div> create using JS to the .container class: we will use querySelector as well as .append to do so:

Code:

```
insertion.html > html > body > script
2  <html lang="en">

7  </head>
8  <body>
9    <div class="container">
10     <div class="box" style="display: flex;" data-givenby="Aditya">Aditya Kumar</div>
11     <div class="box">Aditya Kumar is a boy</div>
12   </div>
13   <script>
14     let divvar = document.createElement("div");
15     divvar.innerHTML = "I have been inserted by <b>Aditya</b>";
16     divvar.setAttribute("class","createdusingjs");
17     document.querySelector(".container").append(divvar);
18   </script>
19 </body>
20 </html>
```

Output:



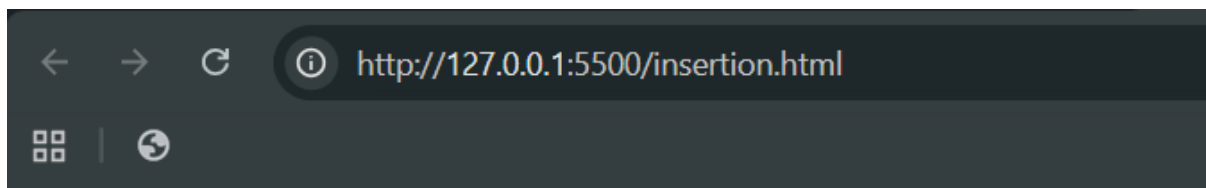
Aditya Kumar
Aditya Kumar is a boy
I have been inserted by **Aditya**

Similarly, we have .before and .after:

Code:

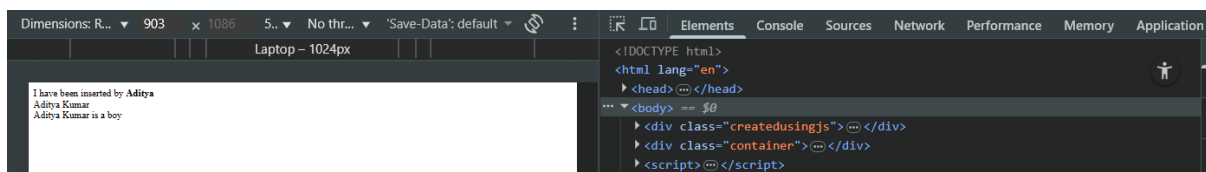
```
9      <div class="container">
10      |   <div class="box" style="display: flex;" data-givenby="Aditya">Aditya Kumar</div>
11      |   <div class="box">Aditya Kumar is a boy</div>
12      | </div>
13      <script>
14      |   let divvar = document.createElement("div");
15      |   divvar.innerHTML = "I have been inserted by <b>Aditya</b>";
16      |   divvar.setAttribute("class", "createdusingjs");
17      |   document.querySelector(".container").before(divvar);
18      | </script>
```

Output:



I have been inserted by **Aditya**
Aditya Kumar
Aditya Kumar is a boy

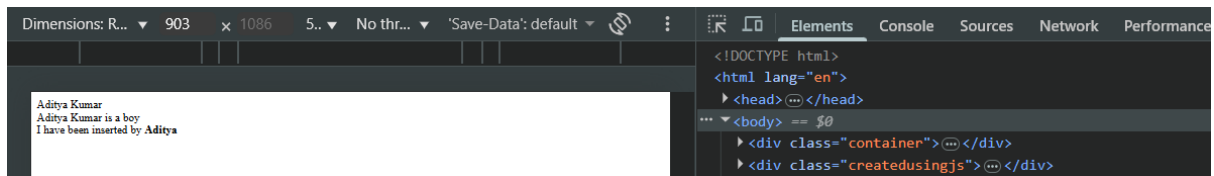
For clarity:



Similarly, we can use the `.after`:

```
insertion.html > html > body > script
2  <html lang="en">
7  </head>
8  <body>
9      <div class="container">
10         <div class="box" style="display: flex;" data-givenby="Aditya">Aditya Kumar</div>
11         <div class="box">Aditya Kumar is a boy</div>
12     </div>
13     <script>
14         let divvar = document.createElement("div");
15         divvar.innerHTML = "I have been inserted by <b>Aditya</b>"
16         divvar.setAttribute("class", "createdusingjs");
17         document.querySelector(".container").after(divvar);
18     </script>
19 </body>
20 </html>
```

Output:

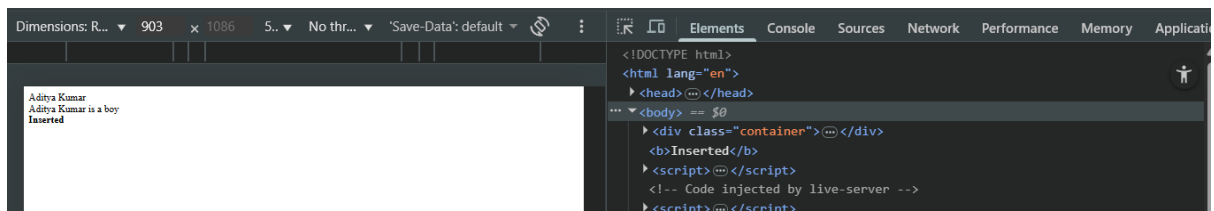


Now, suppose we want to insert an HTML code: we will use `insertAdjacentHTML`

Code: using `afterend`

```
insertion.html > html
2  <html lang="en">
8  <body>
9      <div class="container">
10         <div class="box" style="display: flex;" data-givenby="Aditya">Aditya Kumar</div>
11         <div class="box">Aditya Kumar is a boy</div>
12     </div>
13     <script>
14         let cont = document.querySelector(".container");
15         cont.insertAdjacentHTML("afterend", "<b>Inserted</b>");
16     </script>
17 </body>
18 </html>
```

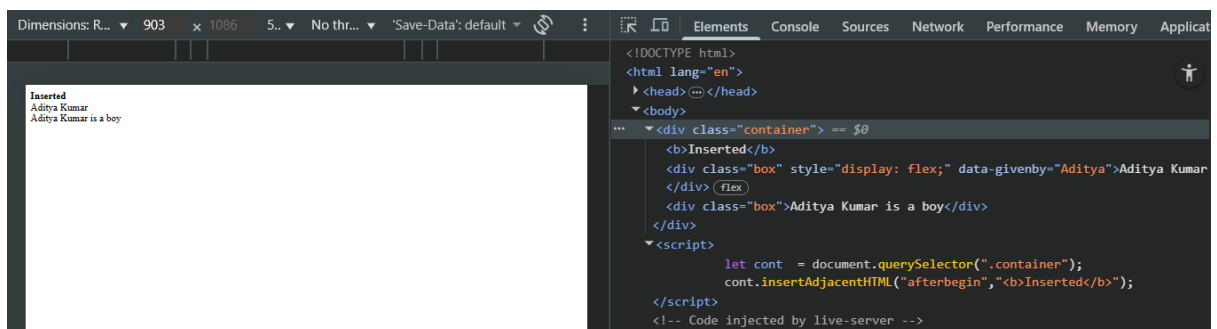
Output:



Code: using afterbegin

```
insertion.html > html
2  <html lang="en">
8  <body>
9    <div class="container">
10     <div class="box" style="display: flex;" data-givenby="Aditya">Aditya Kumar</div>
11     <div class="box">Aditya Kumar is a boy</div>
12   </div>
13   <script>
14     let cont = document.querySelector(".container");
15     cont.insertAdjacentHTML("afterbegin", "<b>Inserted</b>");
16   </script>
17 </body>
18 </html>
```

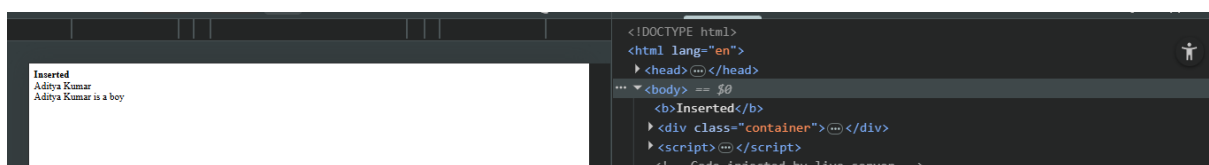
Output:



Code: using beforebegin

```
insertion.html > html > body > script
2  <html lang="en">
8  <body>
9    <div class="container">
10     <div class="box" style="display: flex;" data-givenby="Aditya">Aditya Kumar</div>
11     <div class="box">Aditya Kumar is a boy</div>
12   </div>
13   <script>
14     let cont = document.querySelector(".container");
15     cont.insertAdjacentHTML("beforebegin", "<b>Inserted</b>");
16   </script>
17 </body>
18 </html>
```

Output:

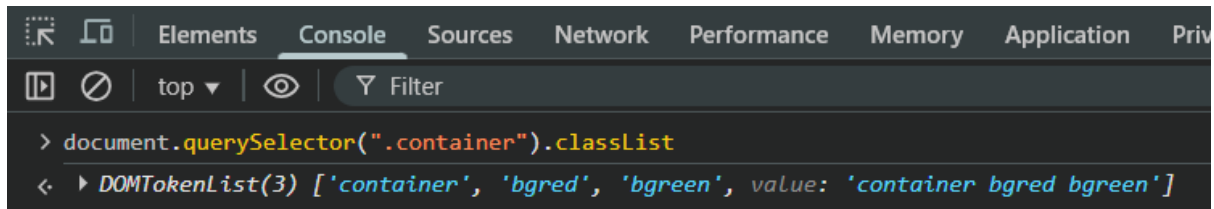


Now, to get the class list, we will use .classList:

Code:

```
<div class="container bgred bgreen">
  <div class="box" style="display: flex;" data-givenby="Aditya">Aditya Kumar</div>
  <div class="box">Aditya Kumar is a boy</div>
</div>
```

Console:



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

Code: same as above

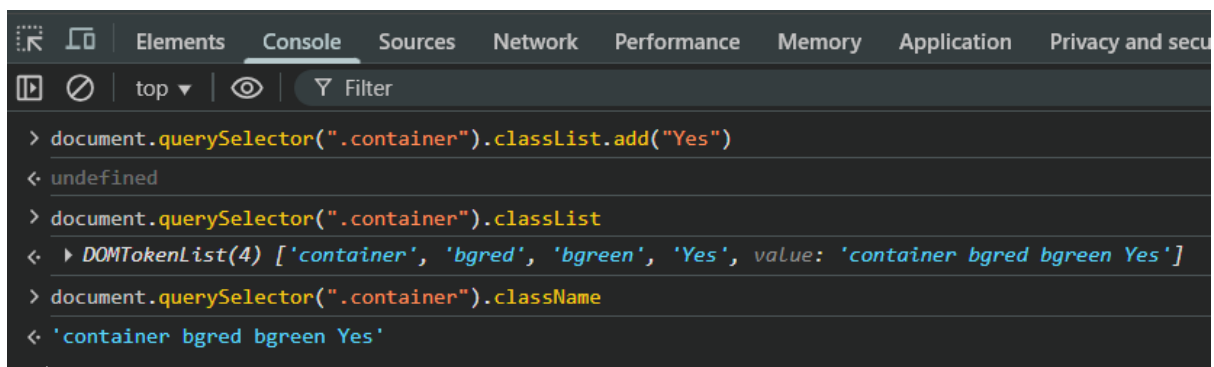
Console:



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

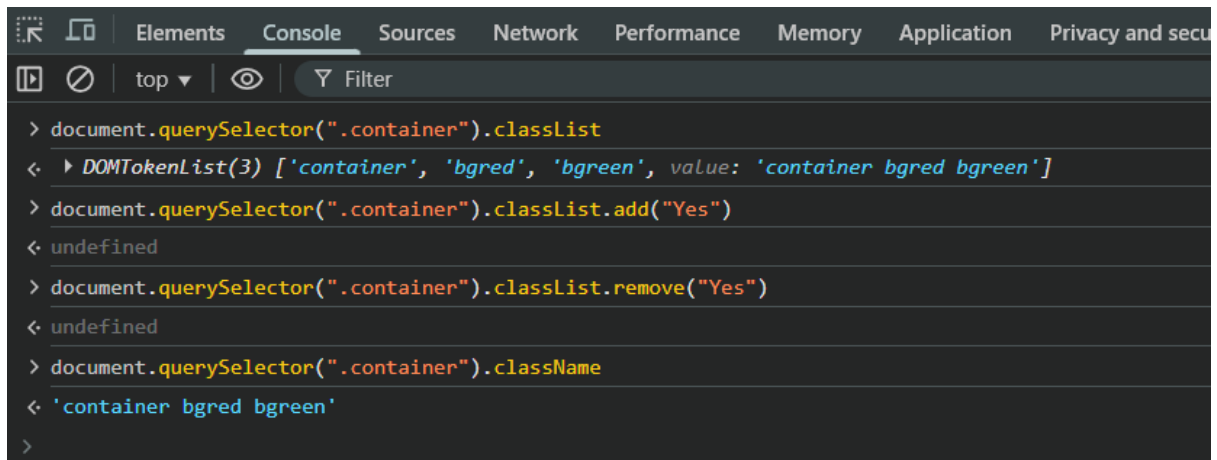
Console:



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:

A screenshot of a web browser's developer console. The console is open, showing a series of commands and their outputs. The commands are: 1. `document.querySelector(".container").classList` which returns `DOMTokenList(3) ['container', 'bgred', 'bgreen', value: 'container bgred bgreen']`. 2. `document.querySelector(".container").classList.add("Yes")` which returns `undefined`. 3. `document.querySelector(".container").classList.remove("Yes")` which returns `undefined`. 4. `document.querySelector(".container").className` which returns `'container bgred bgreen'`. The console interface includes tabs for Elements, Console, Sources, Network, Performance, Memory, Application, and Privacy and security. The Console tab is active, showing a filter input and a list of log entries with expand/collapse icons.

--The End--