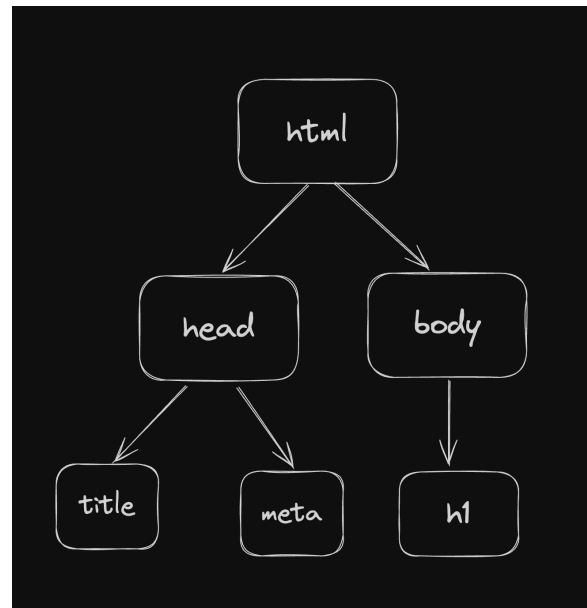




# What is DOM?

The DOM, or Document Object Model, is a programming interface for web documents. It represents the structure of a web page as a tree of objects.

```
<html>
  <head>
    <title>Simple app</title>
    <meta name="description" c
  </head>
  <body>
    <h1>
      hi there
    </h1>
  </body>
</html>
```





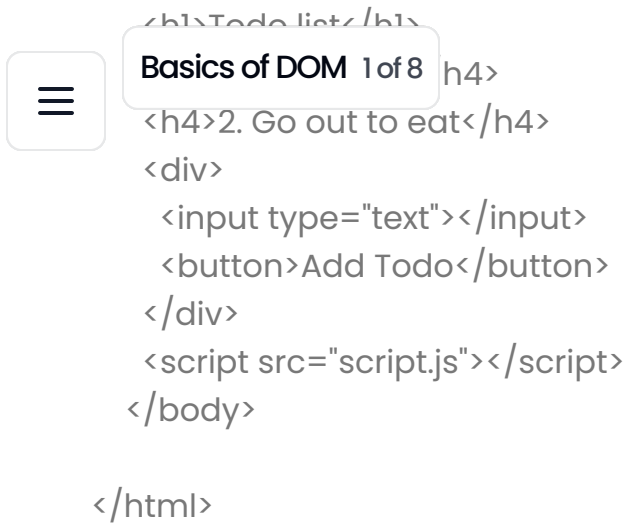
# Static HTML

As the name suggests, **static HTML** represents HTML that does not change.

For example -

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">
    <title>replit</title>
    <link href="style.css" rel="stylesheet" type="text/css" />
  </head>
```





If you click on the **Add Todo** button, nothing happens

# Dynamic HTML

How can you update the elements of the page **dynamically** ?

## Assignment

button, a new TODO should be added.



## document object

In the browser, the **document** object is a fundamental part of the Document Object Model (DOM). It represents the web page currently loaded in the browser and provides a way to interact with and **manipulate** its content.


# Fetching elements

There are 5 popular methods available for fetching DOM elements –

- `querySelector`

`querySelectorAll`

- **Basics of DOM 1 of 8**



- getElementByClassName

- getElementsByClassName

## 1. Fetching the title

```
const title = document.querySelector('h1');  
console.log(title.innerHTML)
```



## 2. Fetching the first TODO (Assignment)

```
const firstTodo = document.querySelector('h4');  
console.log(firstTodo.innerHTML)
```



## 3. Fetching the **second** TODO (Assignment)



```
const secondTodo = document.querySelectorAll('h4')[1];  
console.log(secondTodo.innerHTML)
```



# Updating elements

- `.innerHTML` - Used for updating the **HTML** inside an element
- `.textContent` - Used for updating the **text content** inside an element

## Assignment - Update the first todo's contents

```
const firstTodo = document.querySelector("h4");  
firstTodo.innerHTML = "Dont' take class"
```





# Deleting elements

- `removeChild` - Removes a specific **node** of a **parent**
- `onclick` - function that triggers whenever you **click** on a button

**Assignment – Add a **delete** button right next to the **todo** that deletes that todo**

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width">
  <title>replit</title>
  <link href="style.css" rel="stylesheet" type="text/css" />
</head>

<body>
  <h1>Todo list</h1>
  <div>
```





## Basics of DOM 1 of 8

```

<button onclick="deleteTodo(1)">delete</button>
<div id="todo-2">
  <h4>2. Go out to eat</h4>
  <button onclick="deleteTodo(2)">delete</button>
</div>
</div>
<div>
  <input type="text"></input>
  <button>Add Todo</button>
</div>
</body>

<script>
function deleteTodo(index) {
  const element = document.getElementById("todo-" + index);
  element.parentNode.removeChild(element);
}
</script>

</html>

```

Another experiment we did in class -

```

<html>
  <body id="body">
    <h2>Todo 1</h2>
    <h2>Todo 2</h2>
    <h2>Todo 3</h2>
    <button onclick="deleteRandomTodo()">Delete todo!</button>
  </body>
  <script>
    function deleteRandomTodo() {
      const element = document.querySelector("h2");
      const parentElement = element.parentNode;
      parentElement.removeChild(element);
    }
  </script>
</html>

```







# Adding elements

What we're learning -

- createElement
- appendChild

**Assignment - Write a function to add a TODO `text` to the list of todos**

Steps -

1. Get the current text inside the input element
2. Create a new `div` element
3. Add the `text` from step 1 to the `div` element
4. Append the `div` to the todos list

```
<!DOCTYPE html>  
<html>
```



```
<meta name="viewport" content="width=device-width">
```

## Basics of DOM 1 of 8

```
<link href="style.css" rel="stylesheet" type="text/css" />
</head>
```

```
<body>
  <h1>Todo list</h1>
  <div id="todos">
    <div id="todo-1">
      <h4>1. Take class</h4>
      <button onclick="deleteTodo(1)">delete</button>
    </div>
    <div id="todo-2">
      <h4>2. Go out to eat</h4>
      <button onclick="deleteTodo(2)">delete</button>
    </div>
  </div>
  <div>
    <input id="inp" type="text"></input>
    <button onclick="addTodo()">Add Todo</button>
  </div>
</body>
```

```
<script>
function addTodo() {
  const inputEl = document.getElementById("inp");
  const textNode = document.createElement("div");
  textNode.innerHTML = inputEl.value;
  const parentEl = document.getElementById("todos");
  parentEl.appendChild(textNode);
}
</script>
```

```
</html>
```



# More complex elements

Until now, we created a simple `div` element

```
const textNode = document.createElement("div");  
textNode.innerHTML = inputEl.value;
```



The problem is it doesn't have a corresponding `delete` button.

Can you try to fix it?

## Solution #1

```
<!DOCTYPE html>
```



&lt;head&gt;

&lt;title&gt;Basics of DOM 1 of 8&lt;/title&gt;

&lt;meta name="viewport" content="width=device-width"&gt;

&lt;title&gt;replit&lt;/title&gt;

&lt;link href="style.css" rel="stylesheet" type="text/css" /&gt;

&lt;/head&gt;

&lt;body&gt;

&lt;h1&gt;Todo list&lt;/h1&gt;

&lt;div id="todos"&gt;

&lt;div id="todo-1"&gt;

&lt;h4&gt;1. Take class&lt;/h4&gt;

&lt;button onclick="deleteTodo(1)"&gt;delete&lt;/button&gt;

&lt;/div&gt;

&lt;div id="todo-2"&gt;

&lt;h4&gt;2. Go out to eat&lt;/h4&gt;

&lt;button onclick="deleteTodo(2)"&gt;delete&lt;/button&gt;

&lt;/div&gt;

&lt;/div&gt;

&lt;div&gt;

&lt;input id="inp" type="text"&gt;&lt;/input&gt;

&lt;button onclick="addTodo()"&gt;Add Todo&lt;/button&gt;

&lt;/div&gt;

&lt;/body&gt;

&lt;script&gt;

let currentIndex = 3;

function addTodo() {

const inputEl = document.getElementById("inp");

const textNode = document.createElement("div");

textNode.innerHTML = "&lt;div id='todo-" + currentIndex + "'&gt;&lt;h4&gt;" + inputEl.value

const parentEl = document.getElementById("todos");

parentEl.appendChild(textNode);

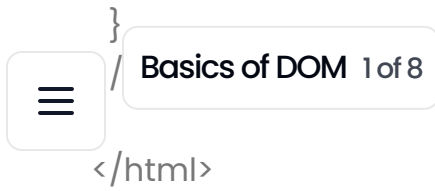
currentIndex = currentIndex + 1;

}

function deleteTodo(index) {

const element = document.getElementById("todo-" + index);

element);



&lt;/html&gt;

## Solution #2

&lt;html&gt;



&lt;head&gt;

&lt;meta charset="utf-8"&gt;

&lt;meta name="viewport" content="width=device-width"&gt;

&lt;title&gt;Todo List&lt;/title&gt;

&lt;link href="style.css" rel="stylesheet" type="text/css" /&gt;

&lt;/head&gt;

&lt;body&gt;

&lt;h1&gt;Todo list&lt;/h1&gt;

&lt;div id="todos"&gt;

&lt;div id="todo-1"&gt;

&lt;h4&gt;1. Take class&lt;/h4&gt;

&lt;button onclick="deleteTodo(1)"&gt;Delete&lt;/button&gt;

&lt;/div&gt;

&lt;div id="todo-2"&gt;

&lt;h4&gt;2. Go out to eat&lt;/h4&gt;

&lt;button onclick="deleteTodo(2)"&gt;Delete&lt;/button&gt;

&lt;/div&gt;

&lt;/div&gt;

&lt;div&gt;

&lt;input id="inp" type="text"&gt;

&lt;button onclick="addTodo()"&gt;Add Todo&lt;/button&gt;

&lt;/div&gt;

&lt;script&gt;

let currentIndex = 3;

function addTodo() {

document.getElementById("inp");

};

**Basics of DOM 1 of 8 {**

```
    alert('Please enter a todo item.');
```

```
    return;
```

```
}
```

```
const parentEl = document.getElementById("todos");
```

```
// Create new todo div
```

```
const newTodo = document.createElement('div');
```

```
newTodo.setAttribute("id", 'todo-' + currentIndex);
```

```
// Create new heading element
```

```
const newHeading = document.createElement('h4');
```

```
newHeading.textContent = currentIndex + '. ' + todoText;
```

```
// Create new button element
```

```
const newButton = document.createElement('button');
```

```
newButton.textContent = 'Delete';
```

```
newButton.setAttribute("onclick", "deleteTodo(" + currentIndex + ")");
```

```
// Append elements to the new todo div
```

```
newTodo.appendChild(newHeading);
```

```
newTodo.appendChild(newButton);
```

```
// Append new todo to the parent element
```

```
parentEl.appendChild(newTodo);
```

```
// Increment the index for the next todo item
```

```
currentIndex++;
```

```
// Clear the input field
```

```
inputEl.value = "";
```

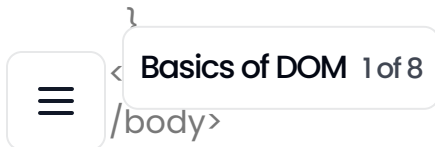
```
}
```

```
function deleteTodo(index) {
```

```
    const element = document.getElementById("todo-" + index);
```

```
    if (element) {
```

```
        element.parentNode.removeChild(element);
```



</html>

## Code to debug

<html>



<body>

<input type="text"></input>

<button onclick="addTodo()">Add todo!</button>

</body>

<script>

let ctr = 1;

function deleteTodo(index) {

const element = document.getElementById(index);

element.parentNode.removeChild(element);

}

function addTodo() {

const inputEl = document.querySelector("input");

const value = inputEl.value;

const newDivEl = document.createElement("div");

newDivEl.setAttribute("id", ctr);

ctr = ctr + 1;

newDivEl.innerHTML = "<div>" + value + '</div><button onclick="deleteTodo(' +

document.querySelector("body").appendChild(newDivEl)

}

</script>

</html>

