6:03 45%

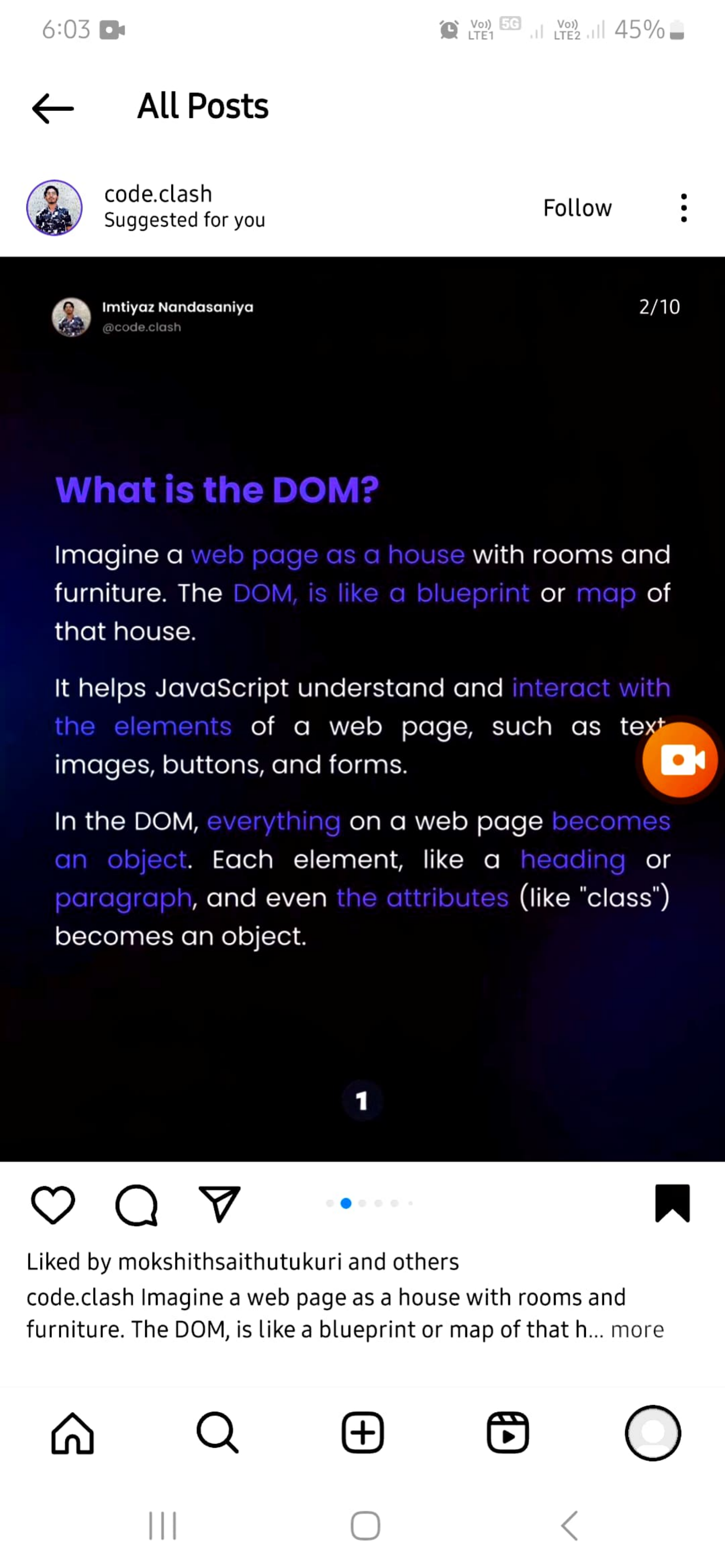
All Posts

code.clash
India

Follow

Liked by mokshithsaithutukuri and others

code.clash Imagine a web page as a house with rooms and furniture. The DOM, is like a blueprint or map of that h... more



6:03 45%

All Posts

code.clash
Suggested for you
Follow

Imtiyaz Nandasaniya
@code.clash
2/10

What is the DOM?

Imagine a web page as a house with rooms and furniture. The DOM, is like a blueprint or map of that house.

It helps JavaScript understand and interact with the elements of a web page, such as text, images, buttons, and forms.

In the DOM, everything on a web page becomes an object. Each element, like a heading or paragraph, and even the attributes (like "class") becomes an object.

1

Liked by mokshithsaithutukuri and others

code.clash Imagine a web page as a house with rooms and furniture. The DOM, is like a blueprint or map of that h... more

6:04 45%

All Posts

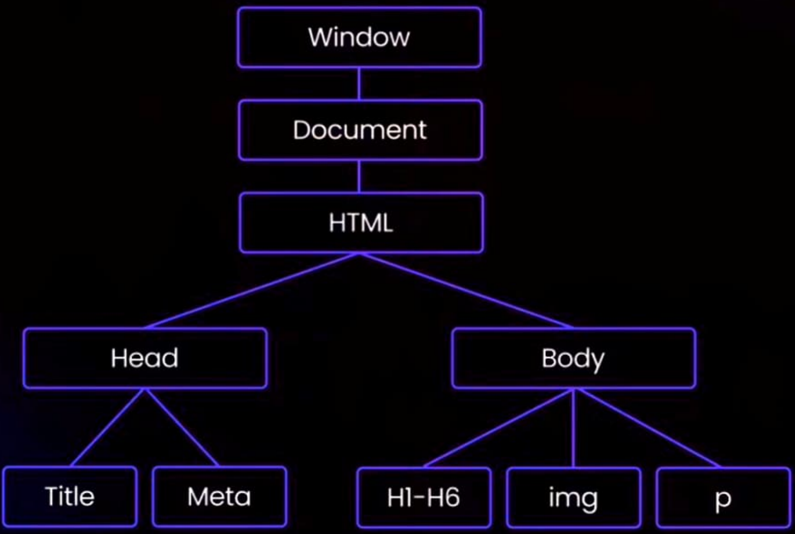
code.clash
India
Follow

Imtiyaz Nandasaniya
@code.clash

3/10

How does the DOM work?

The browser creates a tree-like structure in its memory, where each HTML element becomes a node.



2

Liked by mokshithsaithutukuri and others

code.clash Imagine a web page as a house with rooms and furniture. The DOM, is like a blueprint or map of that h... more

code.clash
India
Follow

Imtiyaz Nandasaniya
@code.clash

4/10

Accessing Elements

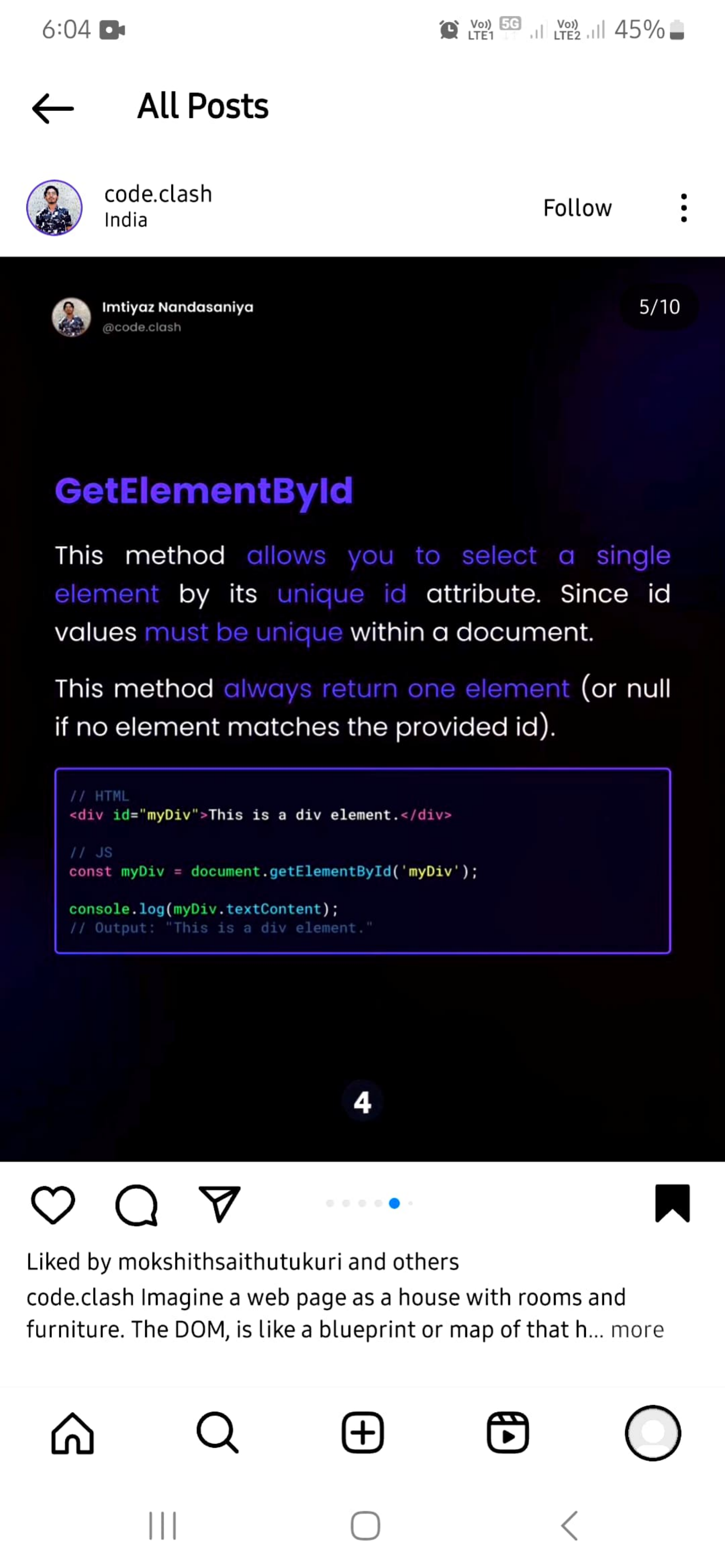
JavaScript can talk to the DOM and ask for specific elements.

For example, it can say, "Hey DOM, can you give me the heading with the ID '`mainHeading`'?"

So, here is the most common method for accessing DOM elements.

1. `getElementById`
2. `getElementsByClassName`
3. `getElementsByTagName`
4. `querySelector`
5. `querySelectorAll`

Liked by mokshithsaithutukuri and others
code.clash Imagine a web page as a house with rooms and furniture. The DOM, is like a blueprint or map of that h... more



All Posts

code.clash
India
Follow

Imtiyaz Nandasaniya
@code.clash

5/10

GetElementById

This method allows you to select a single element by its unique id attribute. Since id values must be unique within a document.

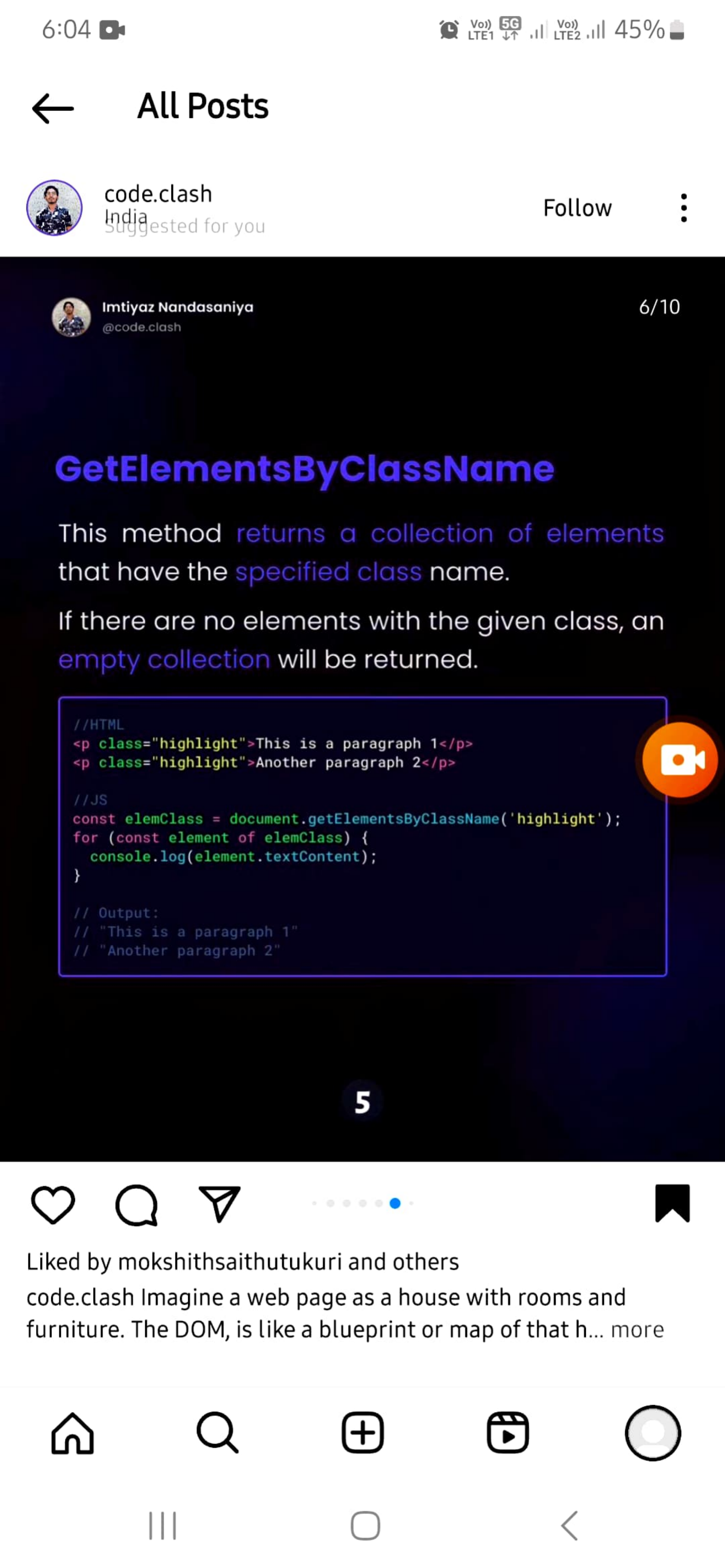
This method always return one element (or null if no element matches the provided id).

```
// HTML
<div id="myDiv">This is a div element.</div>

// JS
const myDiv = document.getElementById('myDiv');

console.log(myDiv.textContent);
// Output: "This is a div element."
```

4

Liked by mokshithsaithutukuri and others

code.clash Imagine a web page as a house with rooms and furniture. The DOM, is like a blueprint or map of that h... more


6:04 45%

All Posts

code.clash
India
Suggested for you
Follow

Imtiyaz Nandasaniya
@code.clash
6/10

GetElementsByClassName

This method returns a collection of elements that have the specified class name.

If there are no elements with the given class, an empty collection will be returned.

```
//HTML
<p class="highlight">This is a paragraph 1</p>
<p class="highlight">Another paragraph 2</p>

//JS
const elemClass = document.getElementsByClassName('highlight');
for (const element of elemClass) {
  console.log(element.textContent);
}

// Output:
// "This is a paragraph 1"
// "Another paragraph 2"
```

5

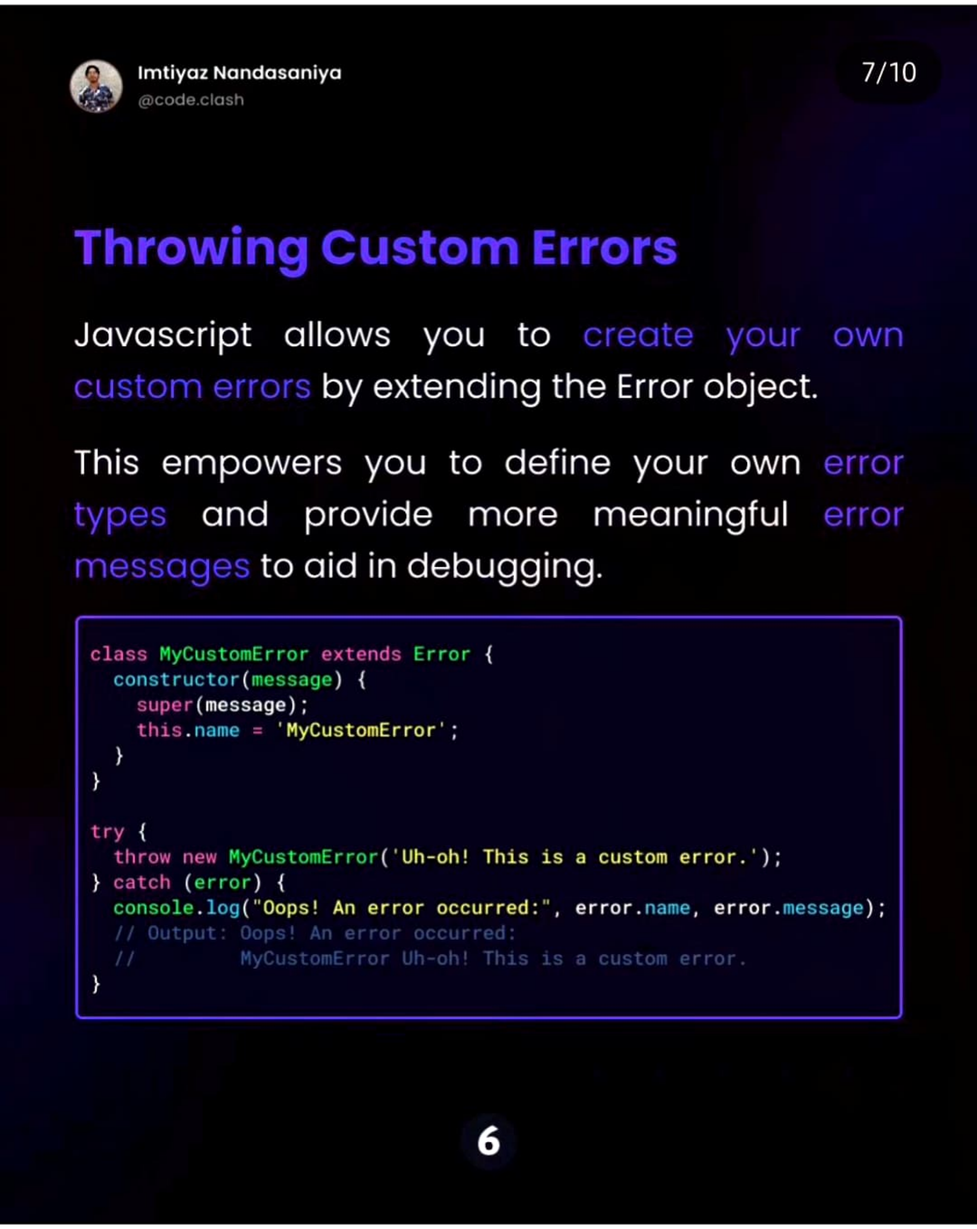
Liked by mokshithsaithutukuri and others

code.clash Imagine a web page as a house with rooms and furniture. The DOM, is like a blueprint or map of that h... more

6:04 45%

All Posts

code.clash
India
Follow



Imtiyaz Nandasaniya
@code.clash

7/10

Throwing Custom Errors

Javascript allows you to create your own custom errors by extending the Error object.

This empowers you to define your own error types and provide more meaningful error messages to aid in debugging.

```
class MyCustomError extends Error {
  constructor(message) {
    super(message);
    this.name = 'MyCustomError';
  }
}

try {
  throw new MyCustomError('Uh-oh! This is a custom error.');
} catch (error) {
  console.log("Oops! An error occurred:", error.name, error.message);
  // Output: Oops! An error occurred:
  //         MyCustomError Uh-oh! This is a custom error.
}
```

6

Liked by mokshithsaithutukuri and others

code.clash Imagine a web page as a house with rooms and furniture. The DOM, is like a blueprint or map of that h... more

Imtiyaz Nandasaniya
@code.clash

QuerySelector

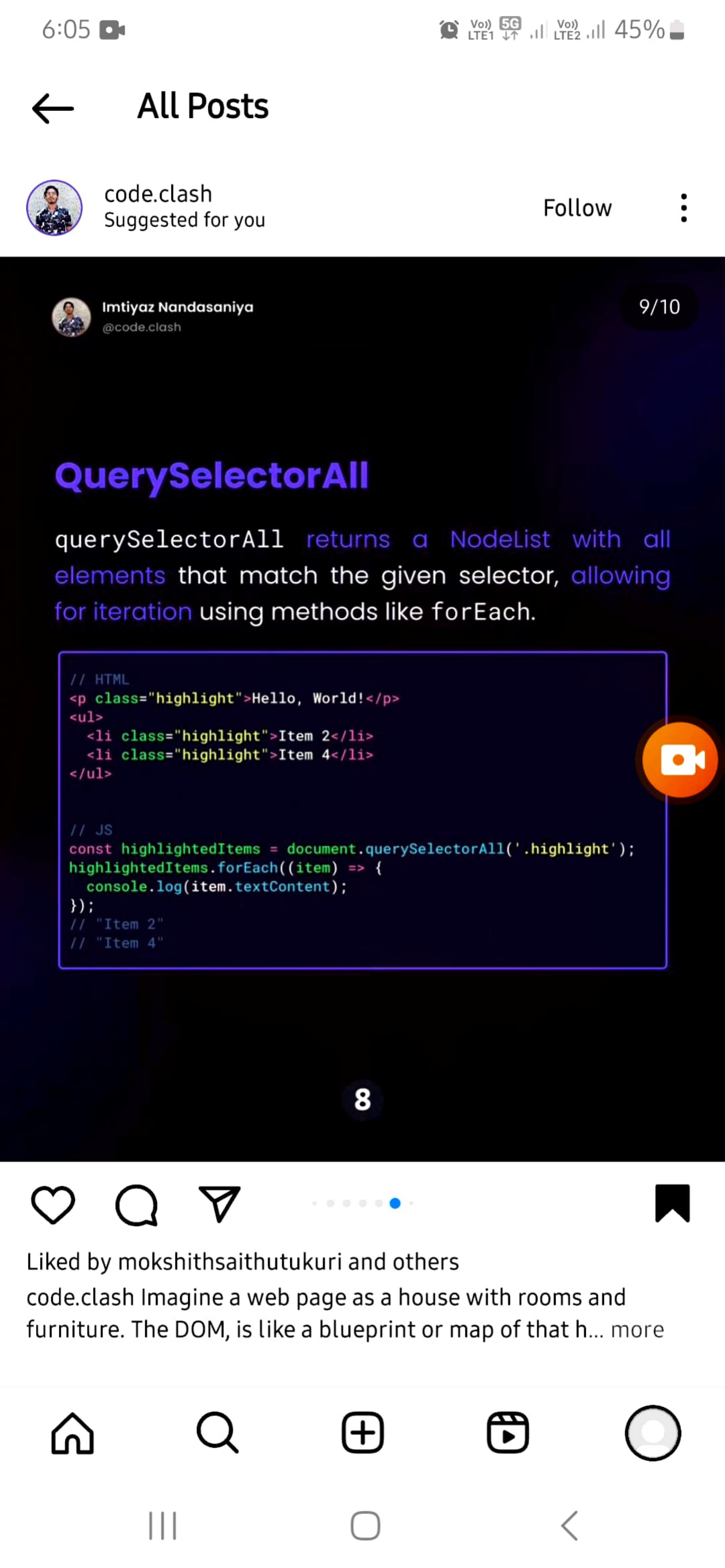
querySelector is a method that selects the first element matching a CSS-like selector and returns it (or null if no match is found).

```
// HTML
<p class="highlight">Hello, World!</p>
<ul>
  <li class="highlight">Item 2</li>
  <li class="highlight">Item 4</li>
</ul>

// JS
const highlightedElement = document.querySelector('.highlight');

console.log(highlightedElement.textContent);
// Output: "Hello, World!"
```

7

Liked by mokshithsaithutukuri and others

code.clash Imagine a web page as a house with rooms and furniture. The DOM, is like a blueprint or map of that h... more



6:05 45%

All Posts

code.clash
Suggested for you
Follow

Imtiyaz Nandasaniya
@code.clash
9/10

QuerySelectorAll

querySelectorAll returns a NodeList with all elements that match the given selector, allowing for iteration using methods like forEach.

```
// HTML
<p class="highlight">Hello, World!</p>
<ul>
  <li class="highlight">Item 2</li>
  <li class="highlight">Item 4</li>
</ul>


// JS
const highlightedItems = document.querySelectorAll('.highlight');
highlightedItems.forEach((item) => {
  console.log(item.textContent);
});
// "Item 2"
// "Item 4"
```

8

Liked by mokshithsaithutukuri and others
code.clash Imagine a web page as a house with rooms and furniture. The DOM, is like a blueprint or map of that h... more

6:07 45%

All Posts

code.clash imagine a web page as a house with rooms and furniture. The DOM, is like a blueprint or map of that h... more

webuniverse02
Bangalore
Follow

1/9

JavaScript
DOM
Manipulating Elements

JS

622 likes

webuniverse02 Hey Developers - Save It For Later JavaScript DOM Manipulate ...... more

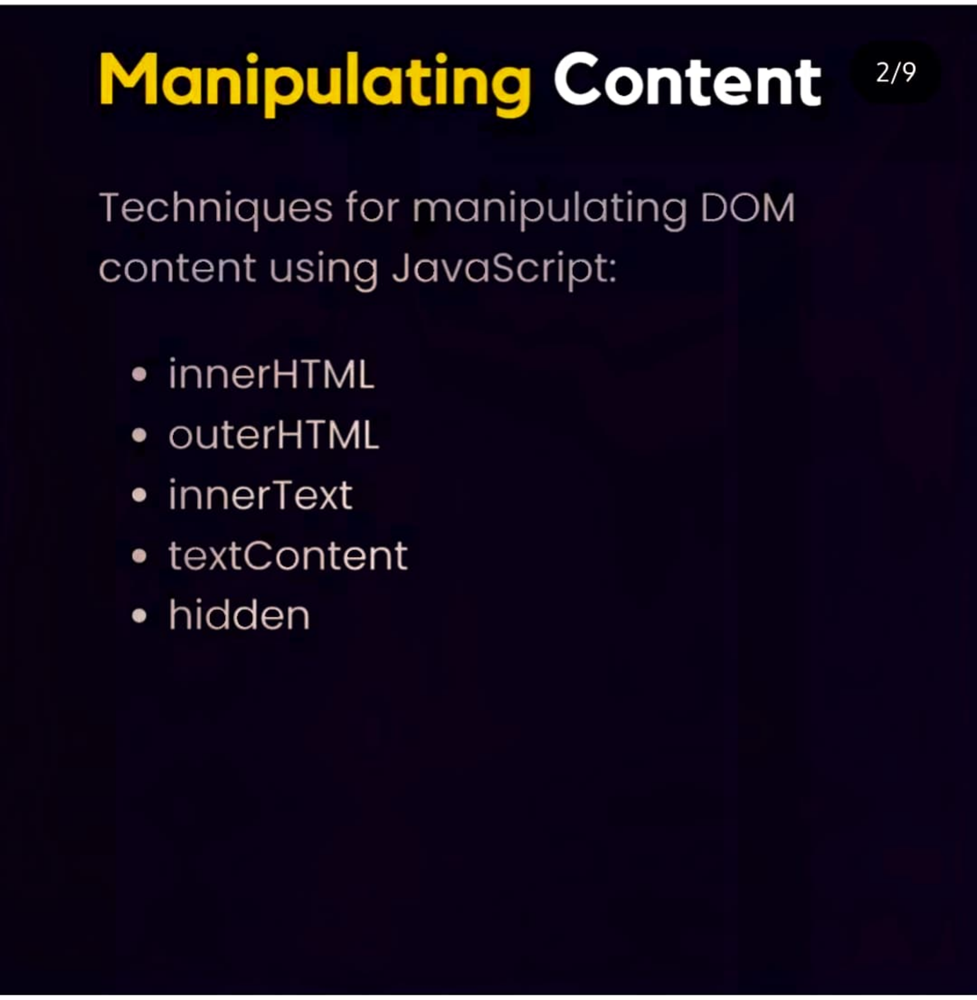
code.clash
Follow

6:07 45%

All Posts

code.clash Imagine a web page as a house with rooms and furniture. The DOM, is like a blueprint or map of that h... more

webuniverse02
Bangalore
Follow



Manipulating Content

2/9

Techniques for manipulating DOM content using JavaScript:

- innerHTML
- outerHTML
- innerText
- textContent
- hidden

622 likes

webuniverse02 Hey Developers - Save It For Later JavaScript DOM Manipulate ...... more

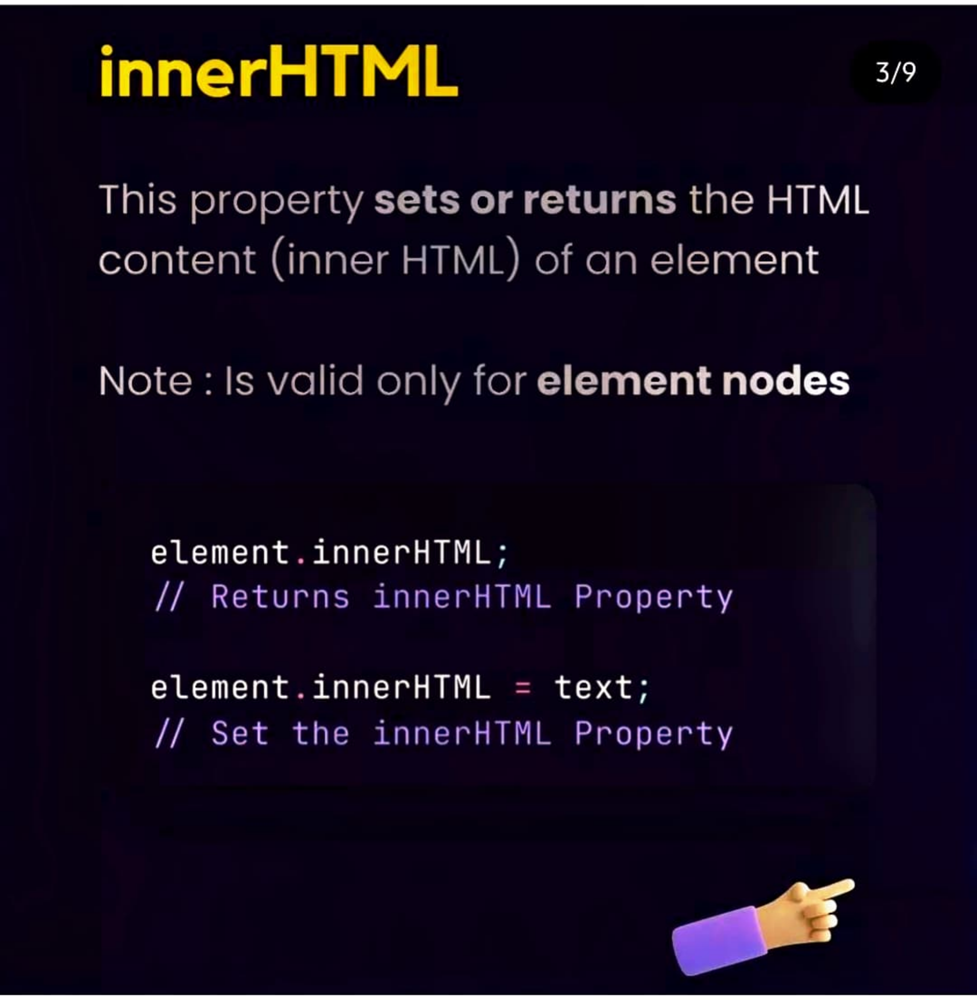
code.clash
Follow

6:07 45%

All Posts

code.clash imagine a web page as a house with rooms and furniture. The DOM, is like a blueprint or map of that h... more

webuniverse02
Bangalore
Follow



innerHTML

3/9

This property **sets or returns** the HTML content (inner HTML) of an element

Note : Is valid only for **element nodes**

```
element.innerHTML;
// Returns innerHTML Property

element.innerHTML = text;
// Set the innerHTML Property
```

622 likes

webuniverse02 Hey Developers - Save It For Later JavaScript DOM Manipulate ...... more

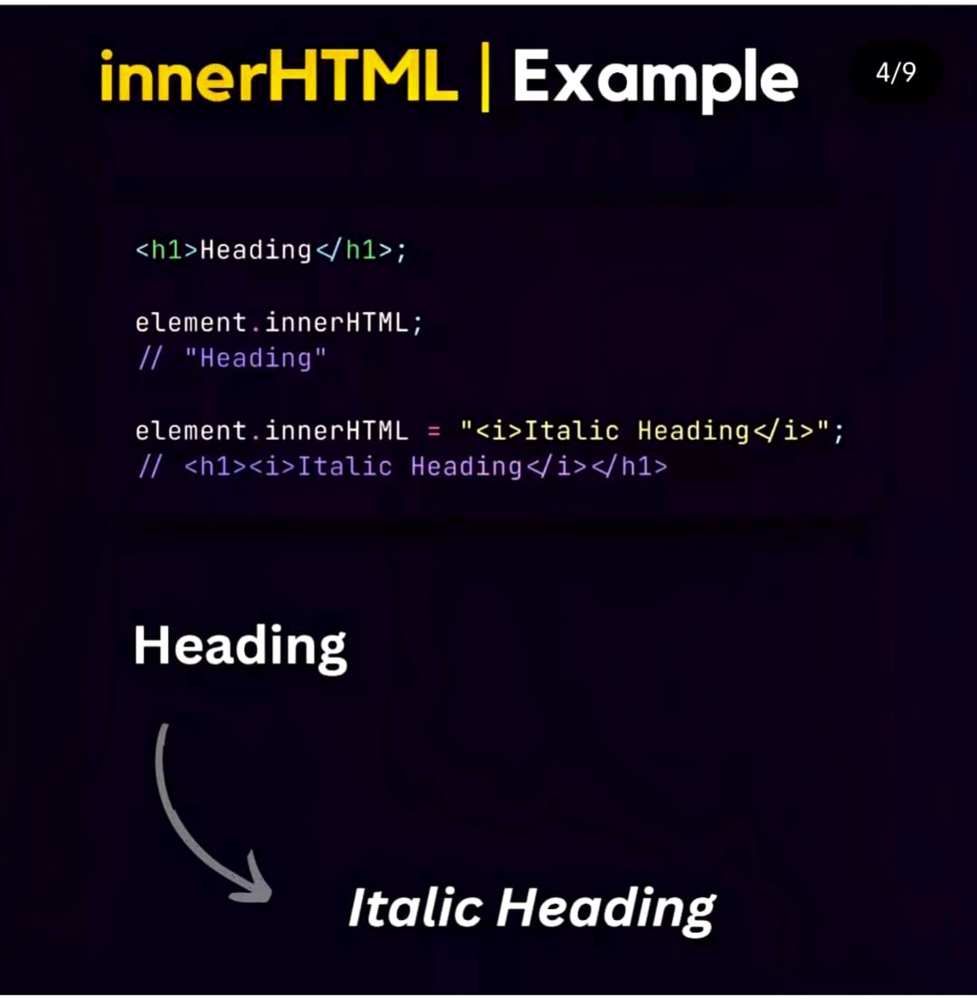
code.clash
Follow

6:07 45%

All Posts

code.clash Imagine a web page as a house with rooms and furniture. The DOM, is like a blueprint or map of that h... more

webuniverse02
Bangalore
Follow



innerHTML | Example

4/9

```
<h1>Heading</h1>;

element.innerHTML;
//  "Heading"

element.innerHTML = "<i>Italic Heading</i>";
//  <h1><i>Italic Heading</i></h1>
```

Heading

Italic Heading

622 likes

webuniverse02 Hey Developers - Save It For Later JavaScript DOM Manipulate ...... more

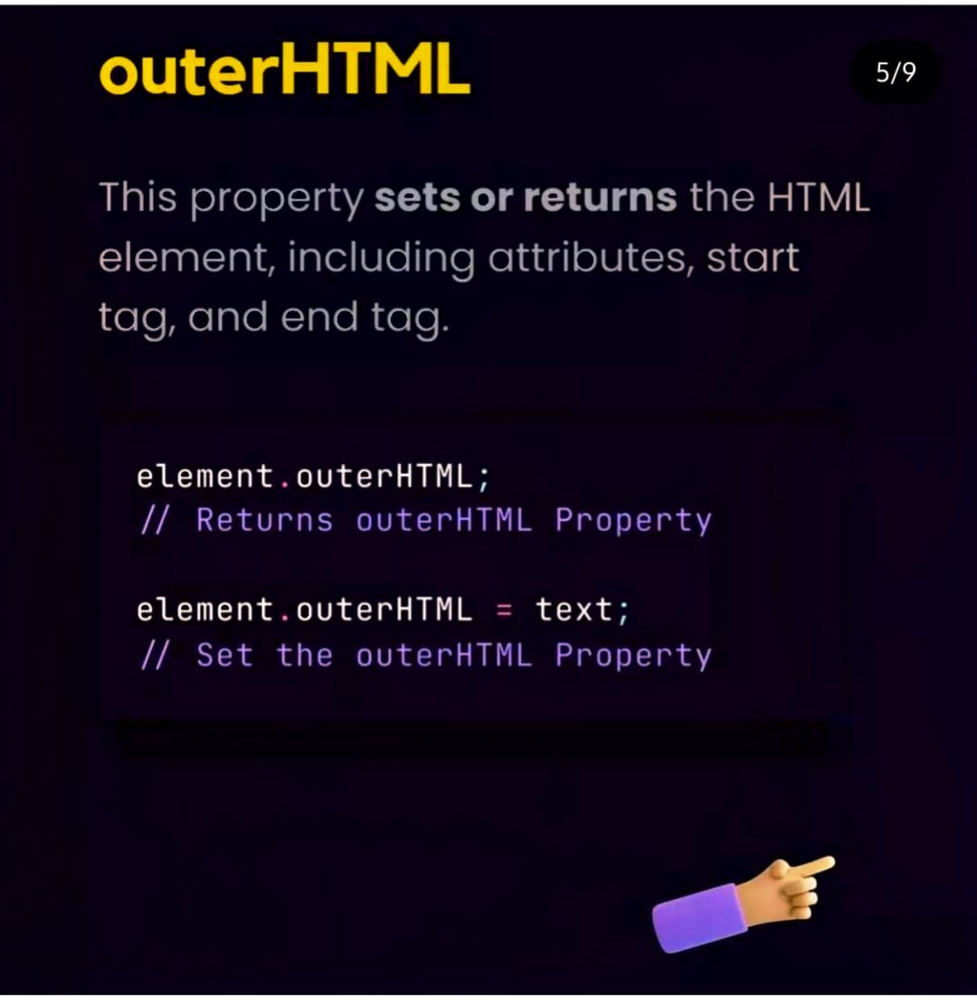
code.clash
Follow

6:07 44%

All Posts

code.clash Imagine a web page as a house with rooms and furniture. The DOM, is like a blueprint or map of that h... more

webuniverse02
Bangalore
Follow



5/9

outerHTML

This property **sets or returns** the HTML element, including attributes, start tag, and end tag.

```
element.outerHTML;
// Returns outerHTML Property

element.outerHTML = text;
// Set the outerHTML Property
```

622 likes

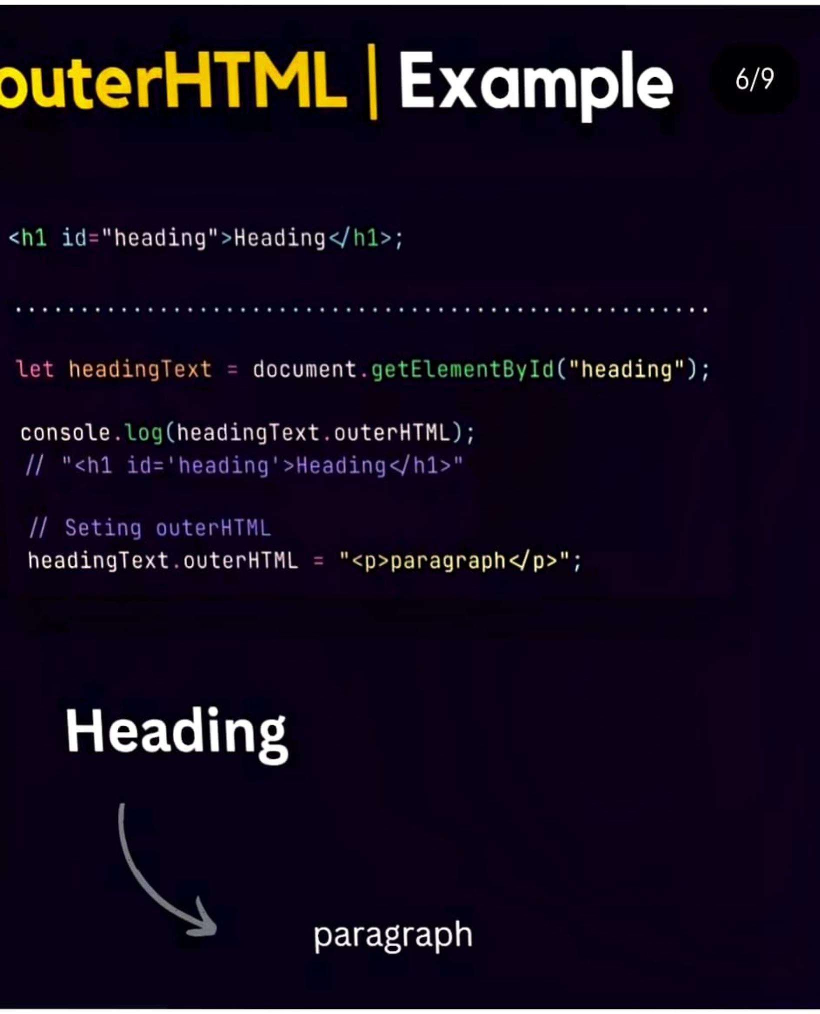
webuniverse02 Hey Developers - Save It For Later JavaScript DOM Manipulate ...... more

code.clash
Follow

All Posts

re. The DOM, is like a blueprint or map of that h... more

webuniverse02
Bangalore
Follow



outerHTML | Example 6/9

```
<h1 id="heading">Heading</h1>;
.................................................
let headingText = document.getElementById("heading");

console.log(headingText.outerHTML);
// "<h1 id='heading'>Heading</h1>"

 // Seting outerHTML
headingText.outerHTML = "<p>paragraph</p>";
```

622 likes
webuniverse02 Hey Developers - Save It For Later JavaScript DOM Manipulate ...... more

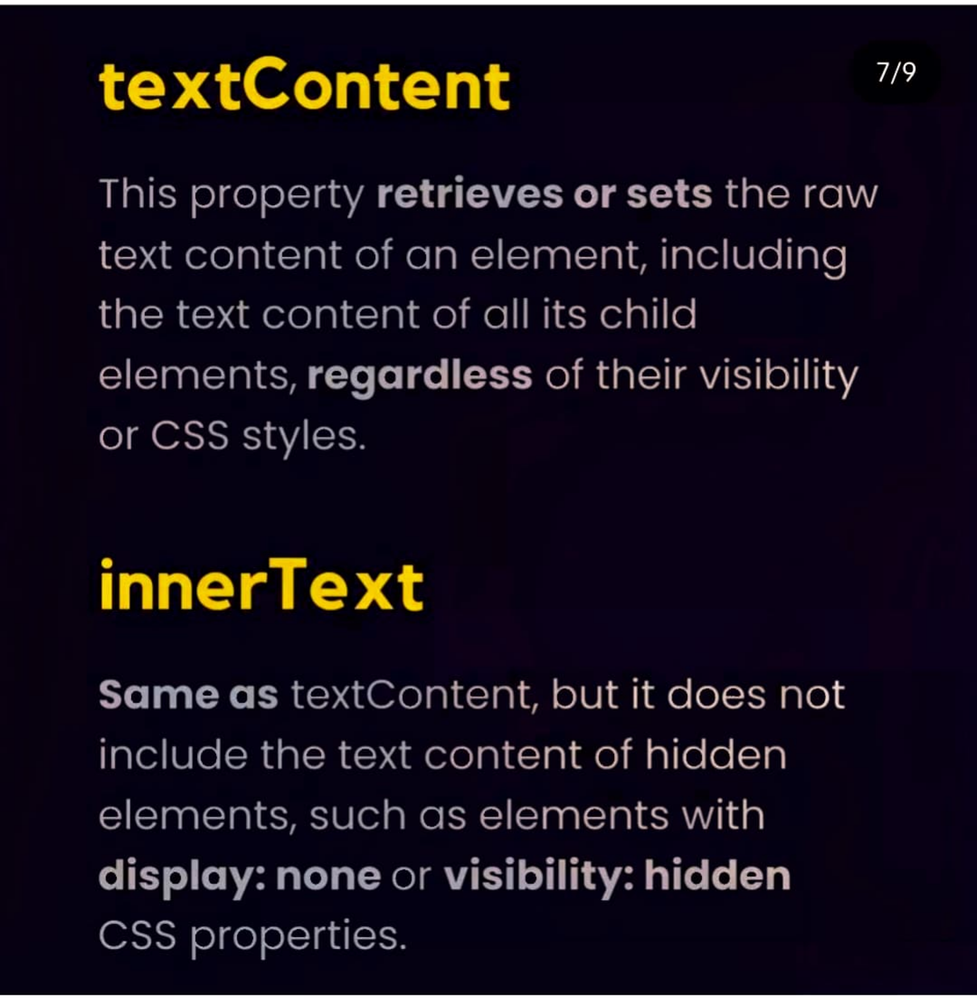
code.clash
Follow

6:08 44%

All Posts

code.clash imagine a web page as a house with rooms and furniture. The DOM, is like a blueprint or map of that h... more

webuniverse02
Bangalore
Follow


7/9

textContent

This property **retrieves or sets** the raw text content of an element, including the text content of all its child elements, **regardless** of their visibility or CSS styles.

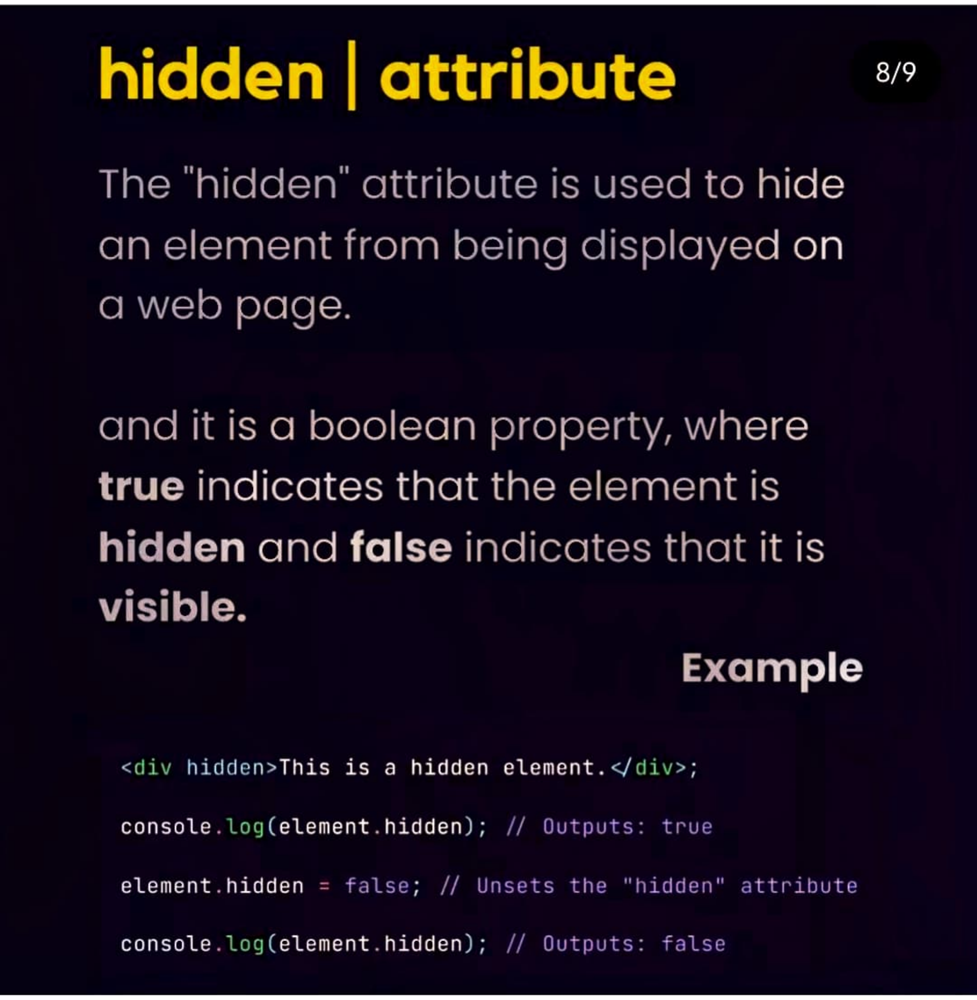
innerText

Same as textContent, but it does not include the text content of hidden elements, such as elements with **display: none** or **visibility: hidden** CSS properties.

622 likes

webuniverse02 Hey Developers - Save It For Later JavaScript DOM Manipulate ...... more

code.clash
Follow



hidden | attribute

8/9

The "hidden" attribute is used to hide an element from being displayed on a web page.

and it is a boolean property, where **true** indicates that the element is **hidden** and **false** indicates that it is **visible.**

Example

```
<div hidden>This is a hidden element.</div>;

console.log(element.hidden); // Outputs: true

element.hidden = false; // Unsets the "hidden" attribute

console.log(element.hidden); // Outputs: false
```

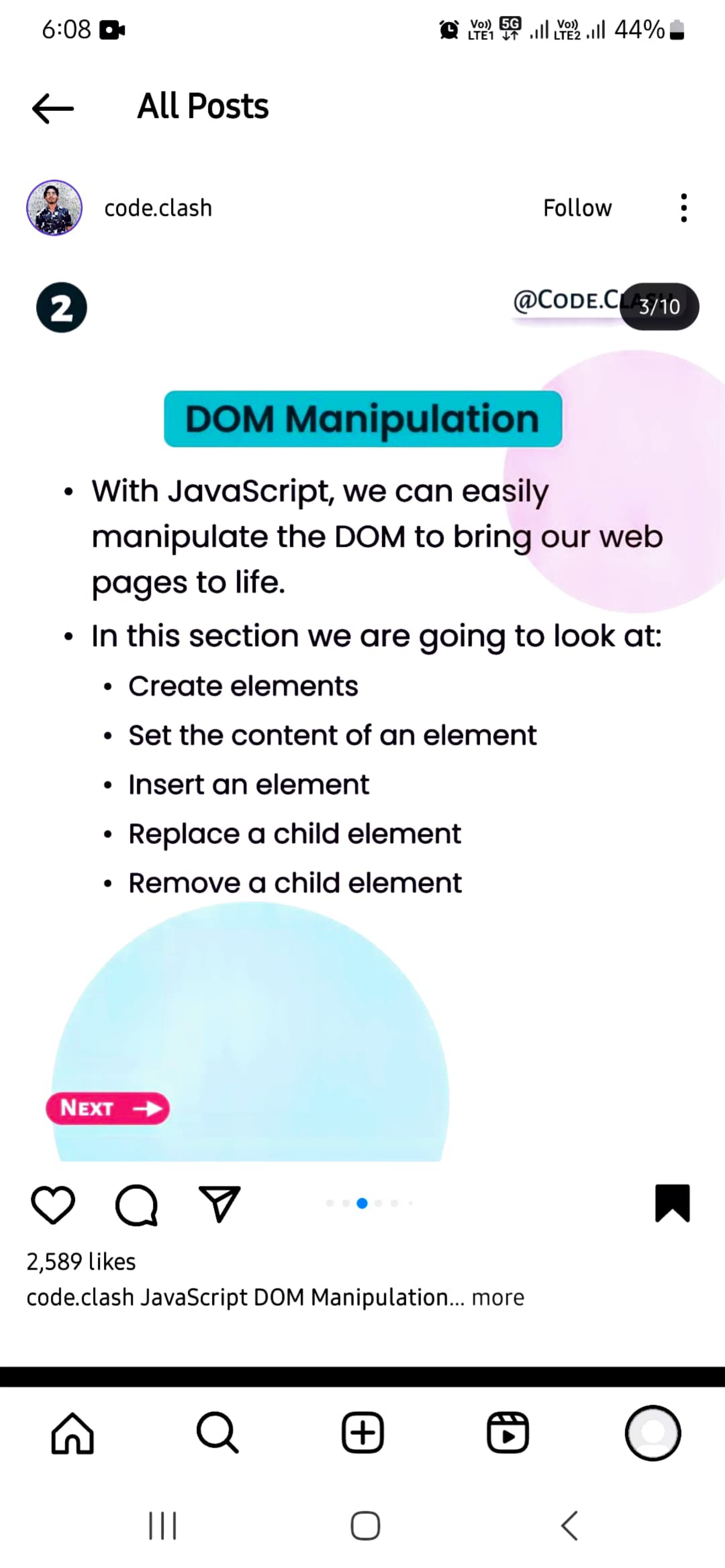
furniture. The DOM, is like a blueprint or map of that h... more

webuniverse02
Bangalore
Follow

622 likes

webuniverse02 Hey Developers - Save It For Later JavaScript DOM Manipulate ...... more

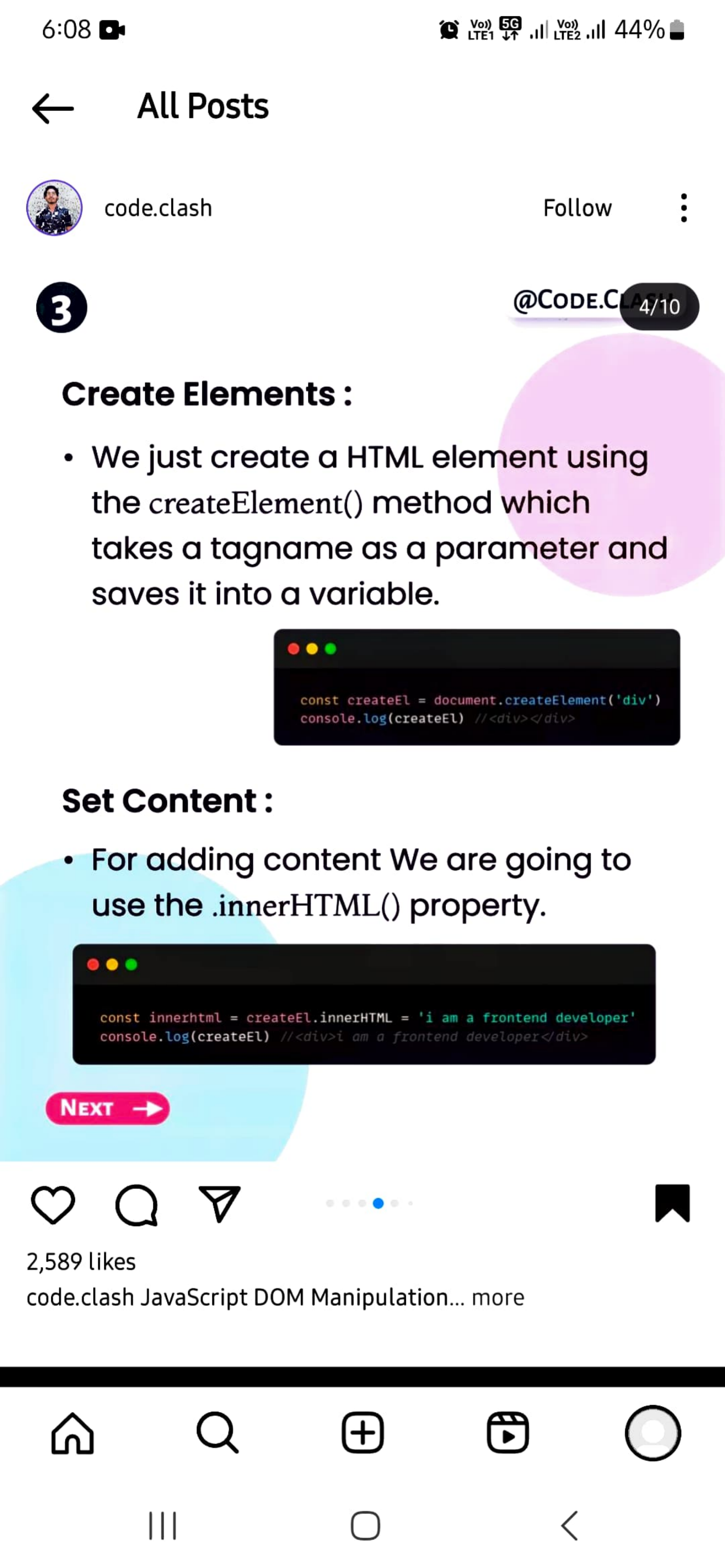
code.clash
Follow



DOM Manipulation

- With JavaScript, we can easily manipulate the DOM to bring our web pages to life.
- In this section we are going to look at:
  - Create elements
  - Set the content of an element
  - Insert an element
  - Replace a child element
  - Remove a child element

code.clash JavaScript DOM Manipulation... more



3

Create Elements :

- We just create a HTML element using the createElement() method which takes a tagname as a parameter and saves it into a variable.

```
const createEl = document.createElement('div')
console.log(createEl) //<div></div>
```

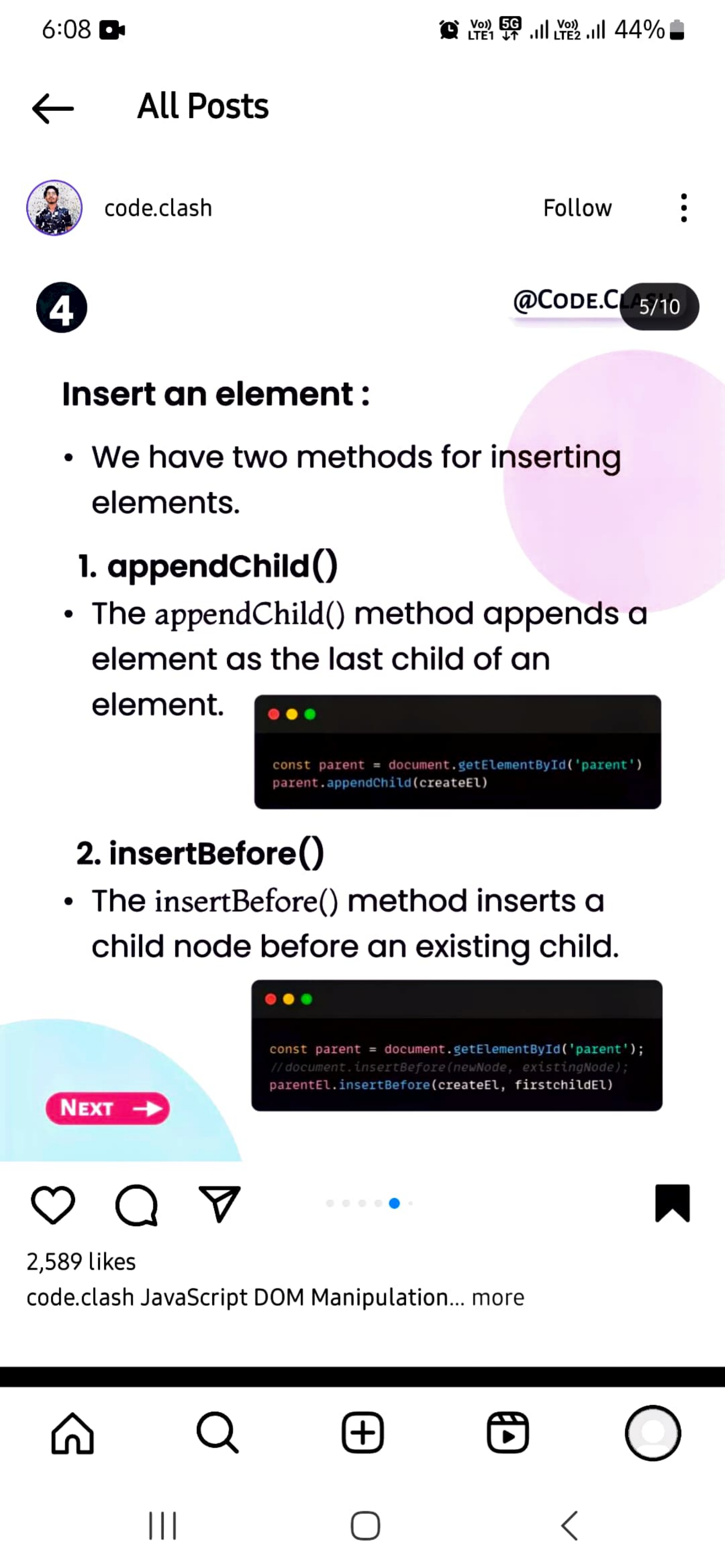
Set Content :

- For adding content We are going to use the .innerHTML() property.

```
const innerhtml = createEl.innerHTML = 'i am a frontend developer'
console.log(createEl) //<div>i am a frontend developer</div>
```

2,589 likes

code.clash JavaScript DOM Manipulation... more



Insert an element :

- We have two methods for inserting elements.

1. appendChild()

- The appendChild() method appends a element as the last child of an element.

```
const parent = document.getElementById('parent')
parent.appendChild(createEl)
```

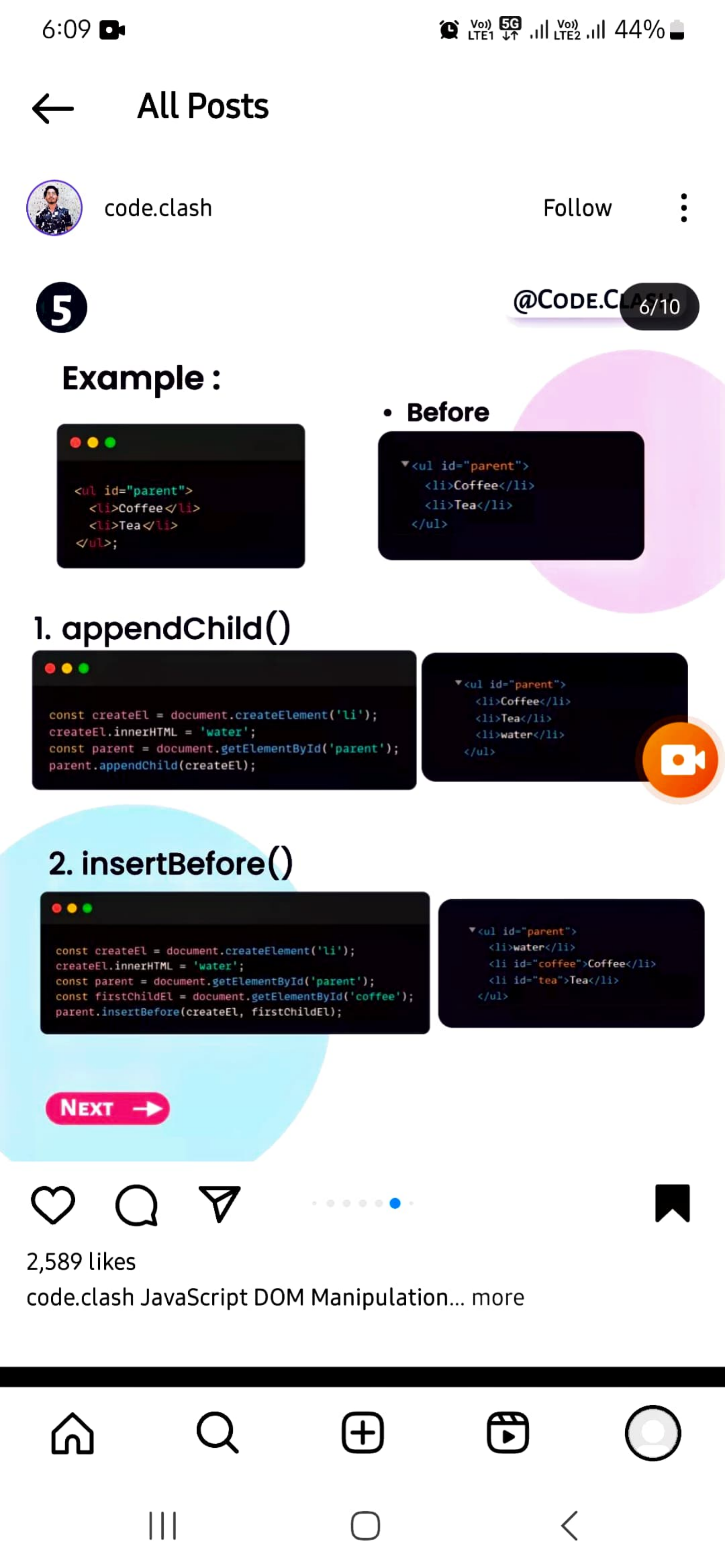
2. insertBefore()

- The insertBefore() method inserts a child node before an existing child.

```
const parent = document.getElementById('parent');
//document.insertBefore(newNode, existingNode);
parentEl.insertBefore(createEl, firstchildEl)
```

2,589 likes

code.clash JavaScript DOM Manipulation... more



6:09 44%

All Posts

code.clash Follow

5

@CODE.CLASH 6/10

Example :

```
<ul id="parent">
  <li>Coffee</li>
  <li>Tea</li>
</ul>;
```

- **Before**

```
▾<ul id="parent">
   <li>Coffee</li>
   <li>Tea</li>
 </ul>
```

1. appendChild()

```
const createEl = document.createElement('li');
createEl.innerHTML = 'water';
const parent = document.getElementById('parent');
parent.appendChild(createEl);
```

```
▾<ul id="parent">
   <li>Coffee</li>
   <li>Tea</li>
   <li>water</li>
 </ul>
```

2. insertBefore()

```
const createEl = document.createElement('li');
createEl.innerHTML = 'water';
const parent = document.getElementById('parent');
const firstChildEl = document.getElementById('coffee');
parent.insertBefore(createEl, firstChildEl);
```

```
▾<ul id="parent">
   <li>water</li>
   <li id="coffee">Coffee</li>
   <li id="tea">Tea</li>
 </ul>
```

NEXT →

2,589 likes

code.clash JavaScript DOM Manipulation... more

6:09 44%

All Posts

code.clash Follow

6

7/10

Replace a Child Element

- Now let's take a look at how we can replace items.

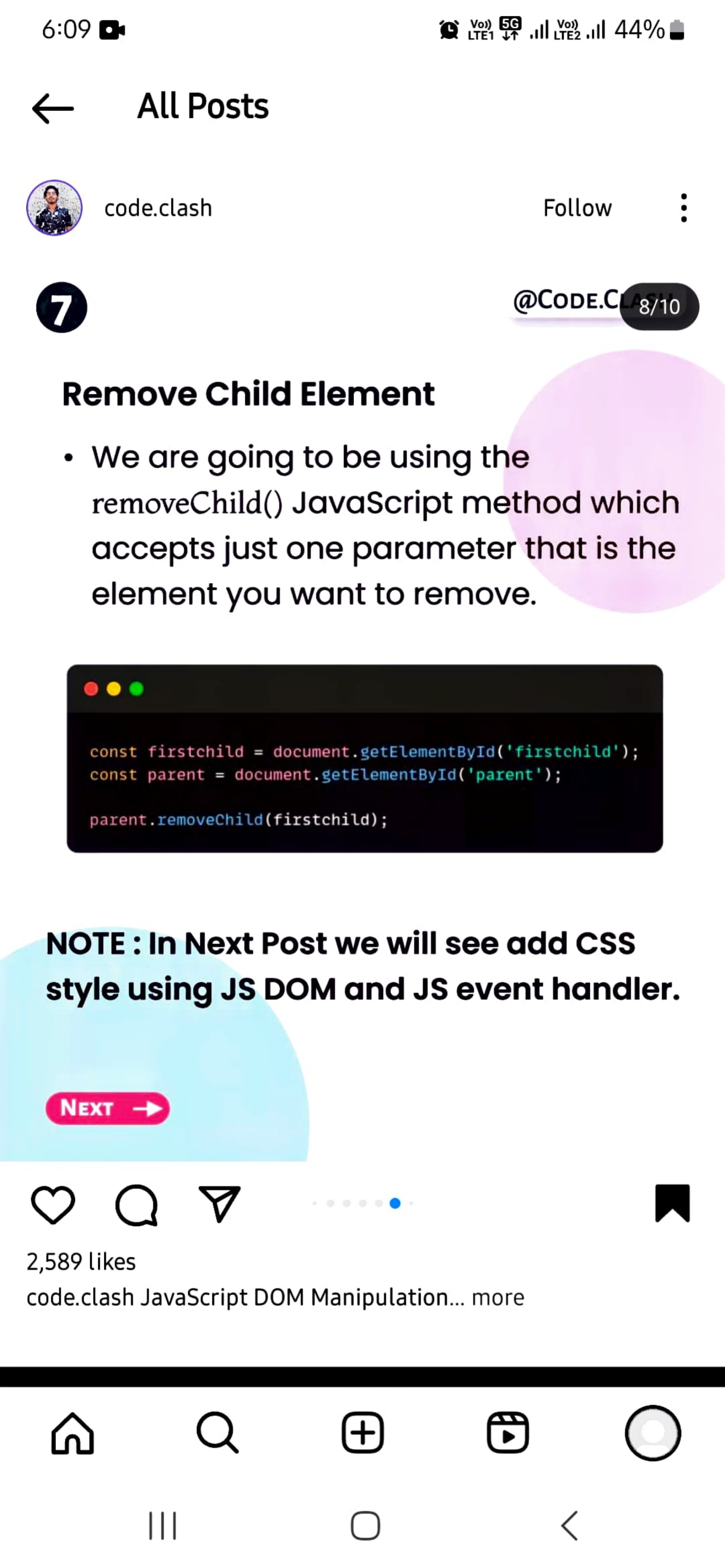
```
const firstchild = document.getElementById('firstchild')
const parent = document.getElementById('parent')

const createEl = document.createElement('div')
const innerhtml = createEl.innerHTML = 'i am a frontend developer'

parent.replaceChild(createEl, firstchild)
```

- Here we replace an element using the replaceChild() method.
- The first argument is the new element and the second argument is the element which we want to replace.

Next →

2,589 likes

code.clash JavaScript DOM Manipulation... more



7

Remove Child Element

- We are going to be using the removeChild() JavaScript method which accepts just one parameter that is the element you want to remove.

```
const firstchild = document.getElementById('firstchild');
const parent = document.getElementById('parent');

parent.removeChild(firstchild);
```

NOTE : In Next Post we will see add CSS style using JS DOM and JS event handler.

2,589 likes

code.clash JavaScript DOM Manipulation... more