

1. innerHTML:

This property is used to get or set the HTML content inside an element.

Example:

```
<div class="container">
  <div class="box">This is a Box.</div>
</div>

<script>
  console.log(document.querySelector(".container").innerHTML);
</script>
```

Output:

```
<div class="box">This is a Box.</div>
```

2. outerHTML:

This property is used to get or set the entire HTML of an element, including the element itself.

Example:

```
<div class="container">
  <div class="box">This is a Box.</div>
</div>
```

```
<script>
  console.log(document.querySelector(".container").outerHTML);
</script>
```

Output:

```
<div class="container">
  <div class="box">This is a Box.</div>
</div>
```

3. innerText:

This property is used to get or set the text content inside an element, without any HTML tags.

Example:

```
<div class="container">
  <div class="box">This is a Box.</div>
</div>

<script>
  console.log(document.querySelector(".container").innerText);
</script>
```

Output:

This is a Box.

4. tagName:

This property returns the tag name of the element in uppercase.

Example:

```
<div class="container">
  <div class="box">This is a Box.</div>
</div>
<script>
  console.log(document.querySelector(".container").tagName);
</script>
```

Output:

DIV

5. nodeName:

This property returns the name of the node in uppercase.

Example:

```
<div class="container">
  <div class="box">This is a Box.</div>
</div>

<script>
  console.log(document.querySelector(".container").nodeName);
</script>
```

Output:

DIV

Difference between tagName and nodeName

1. tagName: Specifically refers to the tag name of an HTML element and only applies to element nodes.
2. nodeName: Applies to all node types (elements, text nodes, comment nodes, etc.) and returns the name of the node.

6. textContent:

It is used to get or set the text content of an element and all its descendants.

Unlike innerText, textContent includes all text, even if it's hidden with CSS styles like display: none

Example:

```
<div class="container">
  <div class="box">This is a Box.</div>
  <div class="fun">Fun</div>
</div>
```

```
<script>  
  console.log(document.querySelector(".container").textContent);  
</script>
```

Output:

```
This is a Box.  
Fun
```

7. hidden:

It is a Boolean attribute used to hide an element from the user. When an element has the hidden attribute, it will not be displayed on the page, and it will behave as if it doesn't exist in the document layout.

Example 1:

```
<div class="container">  
  <div class="box">This is a Box.</div>  
  <div class="fun">Fun</div>  
</div>  
  
<script>  
  console.log(document.querySelector(".container").hidden);  
</script>
```

Output:

```
false
```

Example 2:

```
<div class="container">
  <div class="box">This is a Box.</div>
  <div class="fun">Fun</div>
</div>

<script>
  document.querySelector(".container").hidden = 'true';
</script>
```

It will hide the entire div container and its elements from the main page.

8. hasAttribute:

Checks if the specified element has a certain attribute. Returns true if the attribute exists, otherwise false.

Example:

```
<div class="container">
  <div class="box">This is a Box.</div>
  <div class="fun">Fun</div>
</div>
<script>
  console.log(document.querySelector(".container").hasAttribute("style"));
</script>
```

P.S. Code didn't have any style attribute.

Output:

false

9. getAttribute:

Retrieves the value of the specified attribute from an element. If the attribute doesn't exist, it returns null.

Example:

```
<div class="container">
  <div class="box">This is a Box.</div>
  <div class="fun">Fun</div>
</div>

<script>
  console.log(document.querySelector(".container").getAttribute("style"));
</script>
```

Output:

null

10. setAttribute:

Sets the value of a specified attribute on an element. If the attribute does not exist, it will be created.

Example:

```
<div class="container">
```

```
<div class="box">This is a Box.</div>
<div class="fun">Fun</div>
</div>

<script>
  console.log(document.querySelector(".container").setAttribute("style", "color : red"));
</script>
```

It will make the text color of the elements in the div container red.

11. removeAttribute:

Removes the specified attribute from an element. If the attribute does not exist, this method has no effect.

Example:

```
<div class="container">
  <div class="box">This is a Box.</div>
  <div class="fun">Fun</div>
</div>

<script>
  console.log(document.querySelector(".container").removeAttribute("style"));
</script>
```

Let's say the above code has a style attached to the container div. Then running removeAttribute in the above code will remove all the styling attached to it.

12. attributes

A read-only property that returns a live collection (NamedNodeMap) of all attributes on an element. This collection can be used to

iterate over all attributes of an element.

Example:

```
<div class="container">
  <div class="box">This is a Box.</div>
  <div class="fun">Fun</div>
</div>

<script>
  console.log(document.querySelector(".box").attributes);
</script>
```

Output:

```
NamedNodeMap [ class="container" ]
```

```
0: class="container"
```

```
class: class="container"
```

```
length: 1
```