

Element Specific Properties

In this lesson, we will learn about accessing DOM elements represented by JavaScript objects via the properties that are specific to those JavaScript objects.

Let's begin!

WE'LL COVER THE FOLLOWING



- Listing 6-6: Accessing element via specific properties

DOM elements are represented by JavaScript objects. When you access an element through the document tree, you can access the properties of that specific JavaScript object. Depending on what HTML element is behind the specific object, you can work with different property sets.

For example, when you work with an `` element, you can refer properties that are ``-specific, as shown in Listing 6-6.


Listing 6-6: Accessing element via specific properties

```
<!DOCTYPE html>
<html>
<head>
  <title>Element properties</title>
  <script>
    function logProperties() {
      console.log('ol properties:');
      var ol = document
        .getElementsByName('ol')[0];
      console.log('start: ' + ol.start);
      console.log('type: ' + ol.type);
      console.log('reversed: ' + ol.reversed);
      console.log('reversed attr: ' +
        ol.getAttribute('reversed'));
    }
  </script>
</head>
<body onload="logProperties()">
  <ol start="3" reversed>
    <li id="item1">Item #1</li>
```


```
<li id="item2">Item #2</li>
<li id="item3">Item #3</li>
</ol>

</body>
</html>
```

This code obtains a reference to the `` element in the HTML markup and then logs the start, type, and reversed properties of that object. The code logs the value of the reversed attribute node with the `getAttribute()` function.


 **NOTE:** If you do not remember, the `` element may have start, type, and reversed attributes to define the starting number, the type of number format, and the flag specifying that list order should be descending.

When you display this page with Google Chrome, it will generate this log output:

 console

```
ol properties:
start: 3
type: ""
reversed: false
reversed attr: ""
```

Internet Explorer will produce a different output:

 console

```
ol properties:
start: 3
type: ""
reversed: undefined
reversed attr: ""
```

As you already learned, Internet Explorer does not support the reversed attribute of ``, so when you query it, it retrieves `"undefined"`.

In spite of the missing reversed support, the document tree contains the reversed attribute, because it is specified in the HTML markup.

Generally, all attributes of a node can be accessed through properties. Those

Generally, all attributes of a node can be accessed through properties. Those that are not specified in the HTML markup will be set to their default value unless you change them programmatically. Those that are not implemented by the browser (even if specified in the markup) will return “undefined”.

 **NOTE:** In [Chapter 8](#) you will learn more about the “undefined” value.

In the *next lesson*, we’ll learn to manipulate this document tree.