1) getElementById():-

```
let element = document.getElementById(id);
<html>
   <head>
      <title>JavaScript getElementById() Method</title>
   </head>
   <body>
      <p id="message">A paragraph</p>
  <script>
     const p = document.getElementById('message');
    console.log(p);
  </script>
   </body>
</html>
```

2) getElementsByName()

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="utf-8">
   <title>JavaScript getElementsByName Demo</title>
</head>
<body>
   <p>Please rate the service:</p>
   <p>
      <input type="radio" name="rate" value="Very
poor"> Very poor
      <input type="radio" name="rate" value="Poor">
Poor
      <input type="radio" name="rate" value="OK"> OK
      <input type="radio" name="rate" value="Good">
Good
```

```
        <input type="radio" name="rate" value="Very
Good"> Very Good
    </p>
    <p>
        <button id="btnRate">Submit</button>
    </p>
    <script>
        let btn = document.getElementById('btnRate');

        btn.addEventListener('click', () => {
            let rates = document.getElementsByName('rate');
            rates.forEach((rate) => {
                if (rate.checked) {
                    alert(`You rated: ${rate.value}`);
                }
            })
        });
    </script>
</body>
</html>
```

3) getElementsByTagName()

```
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript getElementsByTagName()
Demo</title>
</head>
<body>
    <h1>JavaScript getElementsByTagName()
Demo</h1>
    <h2>First heading</h2>
```

```
    <p>This is the first paragraph.</p>
    <h2>Second heading</h2>
    <p>This is the second paragraph.</p>
    <h2>Third heading</h2>
    <p>This is the third paragraph.</p>

    <button id="btnCount">Count H2</button>

    <script>
        let btn = document.getElementById('btnCount');
        btn.addEventListener('click', () => {
            let headings =
document.getElementsByTagName('h2');
            alert(`The number of H2 tags:
${headings.length}`);
        });
    </script>
</body>

</html>
```

4) getElementsByClassName()

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

<div class="example">First div element with
class="example".</div>

<div class="example">Second div element with
class="example".</div>
```

```
<p>Click the button to change the text of the first div element with class="example" (index 0).</p>

<button onclick="myFunction()">Try it</button>

<p><strong>Note:</strong> The getElementsByClassName() method is not supported in Internet Explorer 8 and earlier versions.</p>

<script>
function myFunction() {
  var x = document.getElementsByClassName("example");
  x[0].innerHTML = "Hello World!";
}
</script>

</body>
</html>
```

5) querySelector()

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

<h2 class="example">A heading with class="example"</h2>
<p class="example">A paragraph with class="example".</p>

<p>Click the button to add a background color to the first element in the document with class="example".</p>
```

```
<button onclick="myFunction()">Try it</button>

<script>
function myFunction() {

document.querySelector(".example").style.backgroundC
olor = "red";
}
</script>
</body>
</html>
```

6) Get the Parent Element parentNode

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>JavaScript parentNode</title>
</head>
<body>
    <div id="main">
        <p class="note">This is a note!</p>
    </div>

    <script>
        let note = document.querySelector('.note');
        console.log(note.parentNode);
    </script>
</body>
</html>
```

7) Getting Child Elements of a Node in JavaScript

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  margin: 5px;
}
</style>
</head>
<body>

<p>Click the button to add a background color to the second child node (index 1) of div.</p>

<button onclick="myFunction()">Try it</button>

<div id="myDIV">
  <p>First p element</p>
  <p>Second p element</p>
</div>

<p><strong>Note:</strong> Whitespace inside elements is considered as text, and text
is considered as nodes. In this example, index 0, 2 and 4 in DIV are text nodes.</p>

<script>
function myFunction() {
```

```
  var c =
document.getElementById("myDIV").childNodes;
  c[1].style.backgroundColor = "yellow";
}
</script>

</body>
</html>
```

8) Get Parent Element

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  box-sizing: border-box;
  padding: 16px;
  width: 100%;
  background-color: red;
  color: #fff;
}
.closebtn {
  float: right;
  font-size: 30px;
  font-weight: bold;
  cursor: pointer;
}
.closebtn:hover {
  color: #000;
}
</style>
</head>
```

```
<body>

<div>
  <span onclick="this.parentElement.style.display =
'none';" class="closebtn">&times;</span>
  <p>To close this container, click on the X symbol to the
right.</p>
</div>

</body>
</html>
```

9) Getting Siblings

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>JavaScript Siblings</title>
</head>
<body>
    <ul id="menu">
        <li>Home</li>
        <li>Products</li>
        <li class="current">Customer Support</li>
        <li>Careers</li>
        <li>Investors</li>
        <li>News</li>
        <li>About Us</li>
    </ul>

    <script>
```

```
        let getSiblings = function (e) {
            // for collecting siblings
            let siblings = [];
            // if no parent, return no sibling
            if(!e.parentNode) {
                return siblings;
            }
            // first child of the parent node
            let sibling  = e.parentNode.firstChild;
            // collecting siblings
            while (sibling) {
                if (sibling.nodeType === 1 && sibling !== e)
{
                    siblings.push(sibling);
                }
                sibling = sibling.nextSibling;
            }
            return siblings;
        };

        let siblings =
getSiblings(document.querySelector('.current'));
        siblingText = siblings.map(e => e.innerHTML);
        console.log(siblingText);
    </script>
</body>
</html>
```

10) Create Element

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

```
<p>Click the button to make a BUTTON element with
text.</p>

<button onclick="myFunction()">Try it</button>

<script>
function myFunction() {
  var btn = document.createElement("BUTTON");
  btn.innerHTML = "CLICK ME";
  document.body.appendChild(btn);
}
</script>

</body>
</html>
```

11) Text Content:-

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

<p>Click the button get the text content of the button
element.</p>

<button onclick="myFunction()" id="myBtn">Try
it</button>

<p><strong>Note:</strong> The textContent property is
not supported in Internet Explorer 8 and earlier.</p>

<p id="demo"></p>
```

```
<script>
function myFunction() {
  var x =
document.getElementById("myBtn").textContent;
  document.getElementById("demo").innerHTML = x;
}
</script>

</body>
</html>
```

12) Innertext and innerHTML

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

<p>Click the button to make a BUTTON element with
text.</p>

<button onclick="myFunction()">Try it</button>

<script>
function myFunction() {
  var btn = document.createElement("BUTTON");
  btn.innerHTML = "CLICK ME";
  document.body.appendChild(btn);
}
</script>

</body>
</html>
```

13) DocumentFragment:-

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

<p>Click the button to make changes to a list item, using the createDocumentFragment method, then appending the list item as the last child of the list.</p>
<ul><li>Coffee</li><li>Tea</li></ul>
<button onclick="myFunction()">Try it</button>
<script>
function myFunction() {
  var d = document.createDocumentFragment();
d.appendChild(document.getElementsByTagName("LI")[0]);
  d.childNodes[0].childNodes[0].nodeValue = "Milk";
document.getElementsByTagName("UL")[0].appendChild(d);
}
</script>
</body>
</html>
```

14) Insert before

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

<ul id="myList">
  <li>Coffee</li>
  <li>Tea</li>
</ul>
```

```
<p>Click the button to insert an item to the list.</p>

<button onclick="myFunction()">Try it</button>

<p><strong>Example explained:</strong><br>First
create a LI node,<br> then create a Text node,<br> then
append the Text node to the LI node.<br>Finally insert
the LI node before the first child node in the list.</p>

<script>
function myFunction() {
  var newItem = document.createElement("LI");
  var textnode = document.createTextNode("Water");
  newItem.appendChild(textnode);

  var list = document.getElementById("myList");
  list.insertBefore(newItem, list.childNodes[0]);
}
</script>

</body>
</html>
```

15) Insert After

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jq
uery.min.js"></script>
<script>
```

```
$(document).ready(function(){
  $("button").click(function(){
    $("<span>Hello world!</span>").insertAfter("p");
  });
});
</script>
</head>
<body>

<button>Insert span element after each p
element</button>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

16) Append()

```
<!DOCTYPE html>
<html>

<body>

  <div id="idDiv1">
    <div id="idDiv2">Old Content</div>
  </div>

  <button onclick="fun_name()">Append</button>

  <script>
    function fun_name()
```

```
{
  var elem = document.getElementById("idDiv2");
  elem.append("- New Content");

  var elem2 = document.getElementById("idDiv1");
  alert(elem2.outerHTML);
}
</script>
</body>
</html>
```

17) Replacechild

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

<ul
id="myList"><li>Coffee</li><li>Tea</li><li>Milk</li>
</ul>

<p>Click the button to replace the first item in the the
list.</p>

<button onclick="myFunction()">Try it</button>

<p>First create a new Text node called Water.<br>Then
replace the first child node of the first list item, with the
newly created text node.</p>
<p><strong>Note:</strong> This example replaces only
the Text node "Coffee" with a Text node "Water", not the
entire LI element, which would also be an alternative
when replacing nodes.</p>
```

```
<script>
function myFunction() {
  var textnode = document.createTextNode("Water");
  var item =
document.getElementById("myList").childNodes[0];
  item.replaceChild(textnode, item.childNodes[0]);
}
</script>

</body>
</html>
```

18) Clone node

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

<ul id="myList1"><li>Coffee</li><li>Tea</li></ul>
<ul id="myList2"><li>Water</li><li>Milk</li></ul>

<p>Click the button to copy an item from one list to
another.</p>

<button onclick="myFunction()">Try it</button>

<p>Try changing the <em>deep</em> parameter to
false, and only an empty LI element will be cloned.</p>

<script>
function myFunction() {
```

```
  var itm =
document.getElementById("myList2").lastChild;
  var cln = itm.cloneNode(true);

document.getElementById("myList1").appendChild(cln)
;
}
</script>

</body>
</html>
```

19) Removechild()

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

<!-- Note that the <li> elements inside <ul> are not
indented (whitespaces).
If they were, the first child node of <ul> would be a text
node
-->
<ul
id="myList"><li>Coffee</li><li>Tea</li><li>Milk</li>
</ul>

<p>Click the button to remove the first item from the
list.</p>

<button onclick="myFunction()">Try it</button>

<script>
```

```
function myFunction() {
  var list = document.getElementById("myList");
  list.removeChild(list.childNodes[0]);
}
</script>

</body>
</html>
```

20)