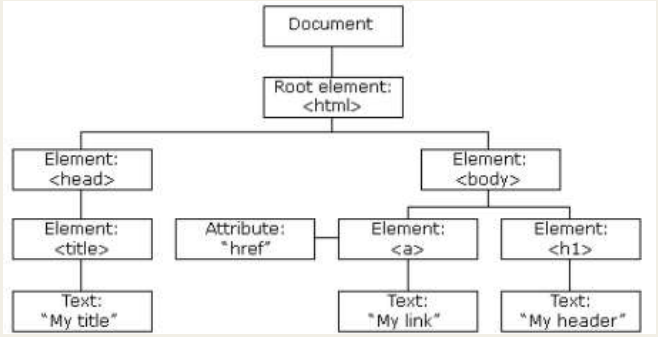
Τι είναι το Μοντέλο Αντικειμένων Εγγράφου (DOM);

Το DOM δημιουργείται από το πρόγραμμα περιήγησης όταν μια ιστοσελίδα φορτώνεται σε έγγραφα HTML ή XML. Χρησιμοποιείται για τον καθορισμό της λογικής δομής αυτών των εγγράφων και για την πρόσβαση και την τροποποίηση των στοιχείων τους.

Το HTML DOM αναφέρεται στο Μοντέλο Αντικειμένων Εγγράφου των εγγράφων HTML, ενώ το XML DOM αναφέρεται στο Μοντέλο Αντικειμένων Εγγράφου των εγγράφων XML. Στο υποκεφάλαιο αυτό, θα εστιάσουμε στο HTML DOM που μπορεί να χρησιμοποιηθεί για την πρόσβαση και την επεξεργασία εγγράφων HTML μέσω της JavaScript.

Το DOM κατασκευάζεται ως μια δομή δέντρου όπου οι καταχωρήσεις των αντικειμένων σε αυτό οργανώνονται ιεραρχικά. Τα αντικείμενα αυτά περιλαμβάνουν όλα τα μέρη ενός εγγράφου HTML όπως στοιχεία, ιδιότητες, κείμενο κ.λπ. Αυτά τα επιμέρους αντικείμενα του δέντρου είναι επίσης γνωστά ως κόμβοι (nodes) και αποτελούνται από γονικούς και θυγατρικούς κόμβους (parent and child nodes). Σε γραφική μορφή, η ιεραρχική δομή του δέντρου DOM μοιάζει με το διάγραμμα που παρατίθεται παρακάτω.



Εικόνα 1 – Δέντρο αντικειμένων DOM HTML
(Πηγή: https://www.w3schools.com/js/js_htmldom.asp)

Εντός του HTML DOM, η JavaScript μπορεί να κάνει τα εξής:

- Αλλαγή όλων των στοιχείων HTML στη σελίδα
- Αλλαγή όλων των ιδιοτήτων HTML στη σελίδα
- Αλλαγή όλων των στυλ CSS στη σελίδα

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικύρωση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

- Κατάργηση υπαρχόντων στοιχείων και ιδιοτήτων HTML
- Προσθήκη νέων στοιχείων και ιδιοτήτων HTML
- Αντίδραση σε όλα τα υπάρχοντα συμβάντα HTML στη σελίδα
- Δημιουργία νέων συμβάντων HTML στη σελίδα

Επιλογή στοιχείων DOM στο JavaScript

Η JavaScript χρησιμοποιείται για να λάβει ή να τροποποιήσει το περιεχόμενο ή τις τιμές των στοιχείων HTML της ιστοσελίδας και να εφαρμόσει κάποια ειδικά εφέ, όπως κινούμενα σχέδια ή απόκρυψη (hide). Για να μπορέσετε να εκτελέσετε οποιαδήποτε ενέργεια, πρέπει να βρείτε ή να επιλέξετε το HTML στοιχείο-στόχος (target element).

Θα εξετάσουμε μερικούς από τους πιο συνηθισμένους τρόπους επιλογής στοιχείων σε μια σελίδα και της επεξεργασίας τους στην JavaScript.

Επιλογή των στοιχείων που βρίσκονται στην κορυφή του δέντρου DOM

Τα στοιχεία που βρίσκονται στην κορυφή μπορούν να γίνουν άμεσα προσβάσιμα ως ιδιότητες εγγράφου.

Για παράδειγμα, για να αποκτήσετε πρόσβαση σε <html> ένα στοιχείο, χρησιμοποιήστε την ιδιότητα document.documentElement. Για <head> ένα στοιχείο, μπορείτε να χρησιμοποιήσετε την ιδιότητα document.head <body> ή την ιδιότητα document.body.

Ας δούμε ένα παράδειγμα για να κατανοήσουμε καλύτερα αυτή τη λειτουργία.

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικύρωση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JS Select Topmost Elements</title>
</head>
<body>
    <script>
    // Display lang attribute value of html element
    alert(document.documentElement.getAttribute("lang")); // Outputs: en

    // Set background color of body element
    document.body.style.background = "yellow";

    // Display tag name of the head element's first child
    alert(document.head.firstElementChild.nodeName); // Outputs: meta
    </script>
</body>
</html>
```

Παράδειγμα στοιχείων που βρίσκονται στην κορυφή του δέντρου DOM
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

* Είναι σημαντικό να σημειωθεί ότι το έγγραφο.body δεν πρέπει να χρησιμοποιείται πριν από το <body> στοιχείο, δεδομένου ότι θα επιστρέψει την τιμή null. Το πρόγραμμα πρέπει να περάσει πρώτα μέσα από το <body> στοιχείο για να αποκτήσει πρόσβαση στην ιδιότητα document.body.
Ας δούμε το παράδειγμα πιο κάτω για να καταλάβουμε γιατί η τίμη <body> θα είναι μηδενική (null):

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JS Document.body Demo</title>
    <script>
    alert("From HEAD: " + document.body); // Outputs: null (since <body> is not
parsed yet)
    </script>
</head>
<body>
    <script>
    alert("From BODY: " + document.body); // Outputs: HTMLBodyElement
    </script>
</body>
</html>
```

Παράδειγμα στοιχείων που βρίσκονται στην κορυφή του δέντρου DOM
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Αυτό το παράδειγμα δείχνει αυτό που είδαμε στην αρχή για τις ιεραρχικές σχέσεις που υπάρχουν μεταξύ των κόμβων. Θα πρέπει να έχετε κατά νου ότι για να αποκτήσετε

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικρότηση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

πρόσβαση στην ιδιότητα document.body, θα πρέπει να ξεκινήσετε από το <body> στοιχείο για να αποφύγετε μηδενικές τιμές.

Επιλογή στοιχείων με αναγνωριστικό (ID)

Αν θέλετε να βρείτε ή να επιλέξετε ένα στοιχείο HTML, ο ευκολότερος τρόπος είναι να το επιλέξετε με βάση το μοναδικό αναγνωριστικό του. Μπορείτε να το κάνετε αυτό με τη μέθοδο getElementById ().

Στο ακόλουθο παράδειγμα, επιλέγεται και επισημαίνεται το στοιχείο που έχει αναγνωριστικό ιδιότητας id ="Mark":

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JS Select Element by ID</title>
</head>
<body>
    <p id="mark">This is a paragraph of text.</p>
    <p>This is another paragraph of text.</p>

    <script>
    // Selecting element with id mark
    var match = document.getElementById("mark");

    // Highlighting element's background
    match.style.background = "yellow";
    </script>
</body>
</html>
```

Επιλογή στοιχείων με αναγνωριστικό (ID)
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Η μέθοδος getElementById () χρησιμοποιείται για την επιστροφή του στοιχείου ως αντικειμένου αν βρεθεί ένα αντίστοιχο στοιχείο. Διαφορετικά, θα επιστρέψει μηδενική τιμή (null).

* Λάβετε υπόψη ότι οποιοδήποτε στοιχείο HTML μπορεί να έχει ένα αναγνωριστικό ιδιότητας, το οποίο πρέπει να είναι μια **μοναδική τιμή μέσα σε μια σελίδα**. Αυτό ουσιαστικά σημαίνει ότι κανένα στοιχείο δεν έχει το ίδιο αναγνωριστικό.

Επιλογή στοιχείων ανά όνομα κλάσης

Αν θέλετε να επιλέξετε όλα τα στοιχεία με συγκεκριμένα ονόματα κλάσης, χρησιμοποιήστε τη μέθοδο getElementsByClassName (). Η επιλογή αυτή θα επιστρέψει ένα αντικείμενο που μοιάζει με πίνακα όλων των θυγατρικών στοιχείων που έχουν όλα τα καθορισμένα ονόματα κλάσης.

Commented [1]: Διευκρίνιση από το πρωτότυπο

Ας δούμε ένα παράδειγμα:

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικρότηση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI SOC-IN)

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JS Select Elements by Class Name</title>
</head>
<body>
    <p class="test">This is a paragraph of text.</p>
    <div class="block test">This is another paragraph of text.</div>
    <p>This is one more paragraph of text.</p>

    <script>
    // Selecting elements with class test
    var matches = document.getElementsByClassName("test");

    // Displaying the selected elements count
    document.write("Number of selected elements: " + matches.length);

    // Applying bold style to first element in selection
    matches[0].style.fontWeight = "bold";

    // Applying italic style to last element in selection
    matches[matches.length - 1].style.fontStyle = "italic";

    // Highlighting each element's background through loop
    for(var elem in matches) {
        matches[elem].style.background = "yellow";
    }
    </script>
</body>
</html>
```

Επιλογή στοιχείων ανά όνομα κλάσης
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Επιλογή στοιχείων ανά όνομα ετικέτας

Αν θέλετε να επιλέξετε στοιχεία με το όνομα της ετικέτας τους (tag), χρησιμοποιήστε τη μέθοδο getElementsByTagName (). Αυτή η μέθοδος θα επιστρέψει επίσης ένα αντικείμενο που μοιάζει με πίνακα όλων των θυγατρικών στοιχείων που έχουν το καθορισμένο όνομα ετικέτας.

Ας δούμε ένα παράδειγμα για να το κατανοήσουμε λίγο καλύτερα:

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικρότηση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JS Select Elements by Tag Name</title>
</head>
<body>
    <p>This is a paragraph of text.</p>
    <div class="test">This is another paragraph of text.</div>
    <p>This is one more paragraph of text.</p>

    <script>
    // Selecting all paragraph elements
    var matches = document.getElementsByTagName("p");

    // Printing the number of selected paragraphs
    document.write("Number of selected elements: " + matches.length);

    // Highlighting each paragraph's background through loop
    for(var elem in matches) {
        matches[elem].style.background = "yellow";
    }
    </script>
</body>
</html>
```

Επιλογή στοιχείων ανά όνομα ετικέτας
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Επιλογή στοιχείων με επιλογείς CSS

Οι Επιλογείς CSS προσφέρουν έναν πολύ δυνατό και αποτελεσματικό τρόπο επιλογής στοιχείων HTML σε ένα έγγραφο. Για να επιλέξετε στοιχεία που ταιριάζουν με τον καθορισμένο επιλογέα CSS, μπορείτε να χρησιμοποιήσετε τη μέθοδο QUERYSelectorAll ().

Αυτή η μέθοδος θα επιστρέψει μια λίστα όλων των στοιχείων που ταιριάζουν με τους καθορισμένους επιλογείς.

Ακολουθήστε το παρακάτω παράδειγμα για να δείτε πώς μπορείτε να εξετάσετε αυτήν τη λίστα σαν ένα πίνακα:

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικρότηση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI SOC-IN)

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JS Select Elements with CSS Selectors</title>
</head>
<body>
    <ul>
        <li>Bread</li>
        <li class="tick">Coffee</li>
        <li>Pineapple Cake</li>
    </ul>

    <script>
    // Selecting all li elements
    var matches = document.querySelectorAll("ul li");

    // Printing the number of selected li elements
    document.write("Number of selected elements: " + matches.length + "<br>");

    // Printing the content of selected li elements
    for(var elem of matches) {
        document.write(elem.innerHTML + "<br>");
    }

    // Applying line through style to first li element with class tick
    matches = document.querySelectorAll("ul li.tick");
    matches[0].style.textDecoration = "line-through";
    </script>
</body>
</html>
```

Επιλογή στοιχείων με επιλογείς CSS
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

* Παρακαλώ σημειώστε ότι η μέθοδος querySelectorAll() υποστηρίζει ψευδο-κλάσεις CSS (pseudo-class) όπως :first-child, : last-child, :hover, κλπ. Ωστόσο, η μέθοδος αυτή θα επιστρέφει πάντα μια κενή λίστα για ψευδο-στοιχεία CSS (pseudo-element) όπως :: before,:after, :: first-line, κλπ.

Στυλιστική ανάπτυξη στοιχείων DOM σε JavaScript

Η JavaScript παρέχει δυναμικούς τρόπους με τους οποίους μπορείτε να αλλάξετε την εμφάνιση των εγγράφων HTML μέσω της εφαρμογής διάφορων στυλ σε στοιχεία HTML. Σχεδόν όλα τα στυλ στοιχείων μπορούν να οριστούν όπως γραμματοσειρές, χρώματα, περιθώρια, περιγράμματα, εικόνες φόντου, στοίχιση κειμένου, πλάτος και ύψος, θέση και ούτω καθεξής.

Στο σημείο αυτό, θα εξετάσουμε τις διάφορες μεθόδους που μπορούν να χρησιμοποιηθούν για να ορισμό ενός στυλ στην JavaScript.

Τρόπος εφαρμογής ενσωματωμένων στυλ σε στοιχεία (Inline Styles)

Η ιδιότητα στυλ χρησιμοποιείται για την εφαρμογή ενσωματωμένων στυλ απευθείας σε ένα συγκεκριμένο στοιχείο HTML. Η ιδιότητα στυλ χρησιμοποιείται στην JavaScript για να λάβει ή να καθορίσει το ενσωματωμένο στυλ ενός στοιχείου.

Στο ακόλουθο παράδειγμα, οι ιδιότητες χρώματος και γραμματοσειράς θα οριστούν για ένα στοιχείο με το αναγνωριστικό id="intro":

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JS Set Inline Styles Demo</title>
</head>
<body>
    <p id="intro">This is a paragraph.</p>
    <p>This is another paragraph.</p>

    <script>
    // Selecting element
    var elem = document.getElementById("intro");

    // Appling styles on element
    elem.style.color = "blue";
    elem.style.fontSize = "18px";
    elem.style.fontWeight = "bold";
    </script>
</body>
</html>
```

Παράδειγμα Ενσωματωμένων Στυλ σε Στοιχεία
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Συμβάσεις Ονοματολογίας Ιδιοτήτων CSS στην JavaScript

Είναι σημαντικό να αναφερθεί ότι πολλές από τις ιδιότητες CSS περιέχουν παύλες (-) στα ονόματά τους, όπως οι ιδιότητες μεγέθους γραμματοσειράς, εικόνας φόντου, διακόσμησης κειμένου κ.λπ. Ωστόσο, στην JavaScript, η παύλα είναι ένας δεσμευμένος τελεστής που συνεπάγεται με το αρνητικό πρόσημο "μείον" . Ως εκ τούτου, δεν είναι δυνατόν να γράψετε μια έκφραση με αυτόν τον τρόπο: elem.style.font-size.

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικύρωση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

Για την επίλυση του θέματος, τα ονόματα ιδιοτήτων CSS στην JavaScript που περιέχουν μία ή περισσότερες παύλες μετατρέπονται σε κεφαλαιοποιημένες λέξεις. Αυτό ουσιαστικά σημαίνει ότι οι παύλες αφαιρούνται και το πρώτο γράμμα μετά την παύλα κεφαλαιοποιείται. Για παράδειγμα, το μέγεθος γραμματοσειράς της ιδιότητας CSS γίνεται μέγεθος γραμματοσειράς στην ιδιότητα DOM.

Λήψη πληροφοριών στυλ από στοιχεία

Η ιδιότητα στυλ χρησιμοποιείται επίσης για να λάβει τα στυλ που εφαρμόζονται στα στοιχεία HTML.

Το παρακάτω παράδειγμα θα λάβει πληροφορίες στυλ από το στοιχείο με αναγνωριστικό id="intro":

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JS Get Element's Style Demo</title>
</head>
<body>
    <p id="intro" style="color:red; font-size:20px;">This is a paragraph.</p>
    <p>This is another paragraph.</p>

    <script>
    // Selecting element
    var elem = document.getElementById("intro");

    // Getting style information from element
    alert(elem.style.color);  // Outputs: red
    alert(elem.style.fontSize);  // Outputs: 20px
    alert(elem.style.fontStyle);  // Outputs nothing
    </script>
</body>
</html>
```

Ιδιότητα στυλ - Παράδειγμα
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Η ιδιότητα στυλ δεν είναι η πιο χρήσιμη όταν πρόκειται για τη λήψη πληροφοριών στυλ από τα στοιχεία, δεδομένου ότι επιστρέφει μόνο τους κανόνες στυλ που ορίζονται στην ιδιότητα στυλ του στοιχείου και όχι εκείνους που προέρχονται από αλλού, όπως οι κανόνες στυλ στα ενσωματωμένα φύλλα στυλ, ή τα εξωτερικά φύλλα στυλ.

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικρότηση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI SOC-IN)

Αν θέλετε να λάβετε τις τιμές όλων των ιδιοτήτων CSS που χρησιμοποιούνται για την ερμηνεία ενός στοιχείου, μπορείτε να χρησιμοποιήσετε τη μέθοδο window.getComputedStyle (), όπως φαίνεται στο παρακάτω παράδειγμα:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>JS Get Computed Style Demo</title>
<style type="text/css">
    #intro {
        font-weight: bold;
        font-style: italic;
    }
</style>
</head>
<body>
    <p id="intro" style="color:red; font-size:20px;">This is a paragraph.</p>
    <p>This is another paragraph.</p>

    <script>
    // Selecting element
    var elem = document.getElementById("intro");

    // Getting computed style information
    var styles = window.getComputedStyle(elem);

    alert(styles.getPropertyValue("color"));  // Outputs: rgb(255, 0, 0)
    alert(styles.getPropertyValue("font-size"));  // Outputs: 20px
    alert(styles.getPropertyValue("font-weight"));  // Outputs: 700
    alert(styles.getPropertyValue("font-style"));  // Outputs: italic
    </script>
</body>
</html>
```

window.getComputedStyle() - Παράδειγμα
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

* Λάβετε υπόψη ότι η τιμή 700 για την ιδιότητα του βάρους μιας γραμματοσειράς CSS είναι η ίδια με τη λέξη-κλειδί Bold. Η λέξη-κλειδί "κόκκινο χρώμα" (colour keyword) είναι η ίδια με την μορφή δεκαεξαδικής τιμής rgb(255,0,0), η οποία είναι η σημειογραφία rgb ενός χρώματος.

Προσθήκη κλάσεων CSS στα στοιχεία

Ένας άλλος τρόπος για να πάρετε ή να καθορίσετε κλάσεις CSS σε στοιχεία HTML είναι χρησιμοποιώντας την ιδιότητα className. Η κλάση είναι μια δεσμευμένη λέξη στην JavaScript. Έτσι, η JavaScript χρησιμοποιεί την ιδιότητα className για να αναφερθεί στην τιμή της ιδιότητας κλάσης HTML.

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικρότηση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

Ας δούμε το ακόλουθο παράδειγμα για να μάθουμε πώς να προσθέτουμε μια νέα κλάση ή να αντικαταστήσουμε όλες τις υπάρχουσες κλάσεις σε ένα στοιχείο με το αναγνωριστικό id="info":

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>JS Add or Replace CSS Classes Demo</title>
<style>
    .highlight {
        background: yellow;
    }
</style>
</head>
<body>
    <div id="info" class="disabled">Something very important!</div>

    <script>
    // Selecting element
    var elem = document.getElementById("info");

    elem.className = "note";  // Add or replace all classes with note class
    elem.className += " highlight";  // Add a new class highlight
    </script>
</body>
</html>
```

Παράδειγμα προσθήκης κλάσεων στα στοιχεία
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Ένας ακόμα καλύτερος τρόπος για να εργαστείτε με τις κλάσεις CSS είναι χρησιμοποιώντας την ιδιότητα classList για να λάβετε, να ρυθμίσετε ή να αφαιρέσετε εύκολα κλάσεις CSS από ένα στοιχείο. Αυτή η ιδιότητα υποστηρίζεται σε όλα τα μεγάλα προγράμματα περιήγησης εκτός από την Internet Explorer, πριν από την έκδοση 10.

Ας δούμε ένα παράδειγμα αυτής της ιδιότητας:

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικρότηση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>JS classList Demo</title>
<style>
    .highlight {
        background: yellow;
    }
</style>
</head>
<body>
    <div id="info" class="disabled">Something very important!</div>

    <script>
    // Selecting element
    var elem = document.getElementById("info");

    elem.classList.add("hide");  // Add a new class
    elem.classList.add("note", "highlight");  // Add multiple classes
    elem.classList.remove("hide"); // Remove a class
    elem.classList.remove("disabled", "note"); // Remove multiple classes
    elem.classList.toggle("visible"); // If class exists remove it, if not add it

    // Determine if class exist
    if(elem.classList.contains("highlight")) {
        alert("The specified class exists on the element.");
    }
    </script>
</body>
</html>
```

Ιδιότητα classList - Παράδειγμα Προσθήκη Κλάσεων σε Στοιχεία
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Δουλεύοντας με τις ιδιότητες

Οι ιδιότητες είναι ειδικές λέξεις που χρησιμοποιούνται μέσα στην ετικέτα εκκίνισης ενός στοιχείου HTML για τον έλεγχο της συμπεριφοράς της ετικέτας ή την παροχή περισσότερων πληροφοριών σχετικά με αυτή.

Στο κεφάλαιο αυτό, θα εξετάσουμε τις διάφορες μεθόδους προσθήκης, αφαίρεσης ή αλλαγής ενός στοιχείου HTML.

Λήψη τιμής ιδιότητας ενός στοιχείου

Για να πάρετε την τρέχουσα τιμή της ιδιότητας ενός στοιχείου, μπορείτε να χρησιμοποιήσετε τη μέθοδο getAttribute (). Εάν η συγκεκριμένη ιδιότητα δεν βρεθεί στο στοιχείο, θα επιστρέψει την τιμή null.

Ας δούμε το ακόλουθο παράδειγμα παρακάτω:

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικρότηση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI SOC-IN)

```html
<a href="https://www.google.com/" target="_blank" id="myLink">Google</a>

<script>
    // Selecting the element by ID attribute
    var link = document.getElementById("myLink");

    // Getting the attributes values
    var href = link.getAttribute("href");
    alert(href); // Outputs: https://www.google.com/

    var target = link.getAttribute("target");
    alert(target); // Outputs: _blank
</script>
```

getAttribute() method – Παράδειγμα Λήψης Τιμής Ιδιότητας Στοιχείου
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-get-set-attributes.php)

Ορισμός ιδιοτήτων στα στοιχεία

Αν θέλετε να ορίσετε μια ιδιότητα σε ένα καθορισμένο στοιχείο, μπορείτε να χρησιμοποιήσετε τη μέθοδο setAttribute (). Εάν η ιδιότητα υπάρχει ήδη στο στοιχείο, η τιμή θα ενημερωθεί. Εάν όχι, θα προστεθεί ένα νέο χαρακτηριστικό με καθορισμένο όνομα και τιμή.

Στο ακόλουθο παράδειγμα, θα προσθέσουμε μια κλάση και ένα απενεργοποιημένη ιδιότητα στο <button> στοιχείο:

```html
<button type="button" id="myBtn">Click Me</button>

<script>
    // Selecting the element
    var btn = document.getElementById("myBtn");

    // Setting new attributes
    btn.setAttribute("class", "click-btn");
    btn.setAttribute("disabled", "");
</script>
```

setAttribute() method – Παράδειγμα Ορισμού Ιδιοτήτων στα Στοιχεία
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-get-set-attributes.php)

Αν θέλετε να ενημερώσετε ή να αλλάξετε την τιμή μιας υπάρχουσας ιδιότητας σε ένα στοιχείο, μπορείτε επίσης να χρησιμοποιήσετε τη μέθοδο setAttribute ().

Ας δούμε ένα παράδειγμα που θα ενημερώσει την τιμή της υπάρχουσας ιδιότητας href ενός στοιχείου σε οξυγώνιες αγκύλες (<a>):

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικύρωση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

```
<a href="#" id="myLink">Tutorial Republic</a>

<script>
    // Selecting the element
    var link = document.getElementById("myLink");

    // Changing the href attribute value
    link.setAttribute("href", "https://www.tutorialrepublic.com");
</script>
```

Η μέθοδος setAttribute() – Παράδειγμα Ενημέρωσης η Αλλαγής Ιδιοτήτων στα Στοιχεία
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-get-set-attributes.php)

Αφαίρεση ιδιοτήτων από τα στοιχεία

Για να αφαιρέσετε μια ιδιότητα από ένα συγκεκριμένο στοιχείο, μπορείτε να χρησιμοποιήσετε τη μέθοδο removeAttribute ().

Θυμηθείτε την ιδιότητα href που αλλάξαμε από το στοιχείο αγκύρωσης. Τώρα θα το αφαιρέσουμε στο ακόλουθο παράδειγμα:

```
<a href="https://www.google.com/" id="myLink">Google</a>

<script>
    // Selecting the element
    var link = document.getElementById("myLink");

    // Removing the href attribute
    link.removeAttribute("href");
</script>
```

Η μέθοδος removeAttribute() – Παράδειγμα Αφαίρεσης Ιδιοτήτων από τα Στοιχεία
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-get-set-attributes.php)

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικύρωση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

Επεξεργασία Στοιχείων DOM στην JavaScript

Μέχρι στιγμής, έχουμε μάθει πώς να επιλέγουμε και να διαμορφώνουμε στοιχεία HTML DOM. Τώρα, θα μάθουμε πώς να προσθέτουμε ή να αφαιρούμε στοιχεία DOM με δυναμικό τρόπο, πώς να λαμβάνουμε το περιεχόμενό τους και πολλά άλλα.

Προσθήκη Νέων Στοιχείων στο DOM

Η μέθοδος document.createElement () χρησιμοποιείται για τη δημιουργία ενός νέου στοιχείου σε ένα έγγραφο HTML. Δημιουργεί ένα νέο στοιχείο. Ωστόσο, δεν το προσθέτει στο DOM.

Απαιτείται ένα ξεχωριστό βήμα για την προσθήκη του στο Dom, όπως φαίνεται στο παρακάτω παράδειγμα:

```
<div id="main">
    <h1 id="title">Hello World!</h1>
    <p id="hint">This is a simple paragraph.</p>
</div>

<script>
// Creating a new div element
var newDiv = document.createElement("div");

// Creating a text node
var newContent = document.createTextNode("Hi, how are you doing?");

// Adding the text node to the newly created div
newDiv.appendChild(newContent);

// Adding the newly created element and its content into the DOM
var currentDiv = document.getElementById("main");
document.body.appendChild(newDiv, currentDiv);
</script>
```

Παράδειγμα Προσθήκης Νέων Στοιχείων
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Στο παράδειγμα που μόλις είδαμε, το appendChild() χρησιμοποιείται για να προσθέσει το νέο στοιχείο στο τέλος οποιωνδήποτε άλλων θυγατρικών κόμβων κάτω από τον καθορισμένο γονικό κόμβο.

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικύρωση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI SOC-IN)

Έχετε επίσης την επιλογή να προσθέσετε το νέο στοιχείο πριν από οποιονδηποτε θυγατρικό κόμβο, όπως φαίνεται στο παρακάτω παράδειγμα:

```
<div id="main">
    <h1 id="title">Hello World!</h1>
    <p id="hint">This is a simple paragraph.</p>
</div>

<script>
// Creating a new div element
var newDiv = document.createElement("div");

// Creating a text node
var newContent = document.createTextNode("Hi, how are you doing?");

// Adding the text node to the newly created div
newDiv.appendChild(newContent);

// Adding the newly created element and its content into the DOM
var currentDiv = document.getElementById("main");
document.body.insertBefore(newDiv, currentDiv);
</script>
```

Παράδειγμα Προσθήκης Νέων Στοιχείων
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Λήψη ή Ρύθμιση περιεχομένων HTML σε DOM

Αν θέλετε να λάβετε ή να ορίσετε τα περιεχόμενα των στοιχείων HTML, μπορείτε να χρησιμοποιήσετε την ιδιότητα innerHTML. Αυτή η ιδιότητα χρησιμοποιείται για τη ρύθμιση ή τη λήψη της σήμανσης HTML μέσα στο στοιχείο, το οποίο έχει το περιεχόμενο μεταξύ των ετικετών ανοίγματος και κλεισίματος.

Ας δούμε ένα παράδειγμα για να το κατανοήσουμε καλύτερα:

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικρότηση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

```
<div id="main">
    <h1 id="title">Hello World!</h1>
    <p id="hint">This is a simple paragraph.</p>
</div>

<script>
// Getting inner HTML contents
var contents = document.getElementById("main").innerHTML;
alert(contents); // Outputs inner html contents

// Setting inner HTML contents
var mainDiv = document.getElementById("main");
mainDiv.innerHTML = "<p>This is <em>newly inserted</em> paragraph.</p>";
</script>
```

Λήψη ή Ρύθμιση περιεχομένων HTML σε DOM
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Όπως μπορούμε να δούμε στο παράδειγμα, τα νέα στοιχεία εισάγονται πολύ εύκολα στο Dom με την εσωτερική ιδιότητα innerHTML. Ωστόσο, αυτή η ιδιότητα αντικαθιστά όλο το υπάρχον περιεχόμενο ενός στοιχείου.

Ως εκ τούτου, εάν δεν θέλετε να αντικαταστήσετε τα υπάρχοντα περιεχόμενα ενός στοιχείου, μπορείτε να χρησιμοποιήσετε τη μέθοδο insertAdjacentHTML (). Η μέθοδος αυτή λαμβάνει δύο παραμέτρους: την HTML που θα εισαχθεί και τη θέση της. Η θέση πρέπει να είναι μία από τις ακόλουθες: "prebegin", "afterbegin", "beforeend" και "afterend". Είναι επίσης σημαντικό να σημειωθεί ότι αυτή η μέθοδος υποστηρίζεται από όλα τα μεγάλα προγράμματα περιήγησης.

Στο παρακάτω παράδειγμα, μπορείτε να δείτε πώς λειτουργεί η τοποθέτηση:

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικρότηση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI SOC-IN)

```html
<!-- beforebegin -->
<div id="main">
    <!-- afterbegin -->
    <h1 id="title">Hello World!</h1>
    <!-- beforeend -->
</div>
<!-- afterend -->

<script>
// Selecting target element
var mainDiv = document.getElementById("main");

// Inserting HTML just before the element itself, as a previous sibling
mainDiv.insertAdjacentHTML('beforebegin', '<p>This is paragraph one.</p>');

// Inserting HTML just inside the element, before its first child
mainDiv.insertAdjacentHTML('afterbegin', '<p>This is paragraph two.</p>');

// Inserting HTML just inside the element, after its last child
mainDiv.insertAdjacentHTML('beforeend', '<p>This is paragraph three.</p>');

// Inserting HTML just after the element itself, as a next sibling
mainDiv.insertAdjacentHTML('afterend', '<p>This is paragraph four.</p>');
</script>
```

insertAdjacentHTML() method - Παράδειγμα Λήψης ή Ρύθμισης Περιεχομένων HTML στο DOM
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

* Λάβετε υπόψη ότι για τις θέσεις beforebegin και afterend για να λειτουργήσει ο κόμβος πρέπει να είναι στο δέντρο Dom και να έχει ένα γονικό στοιχείο.
* Επίσης, κατά την εισαγωγή ενός HTML σε μια σελίδα, προσέξτε να μην χρησιμοποιείτε είσοδο χρήστη που δεν έχει διαφύγει (escaped)/καθαριστεί (sanitised), για να αποτρέψετε τις επιθέσεις τύπου XSS.

Αφαίρεση υπαρχόντων στοιχείων από το DOM

Για να αφαιρέσετε έναν θυγατρικό κόμβο από το Dom, μπορείτε να χρησιμοποιήσετε τη μέθοδο removeChild (). Αυτή η μέθοδος θα επιστρέψει επίσης τον κόμβο που αφαιρέθηκε.

Ας δούμε το ακόλουθο παράδειγμα παρακάτω:

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικύρωση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI SOC-IN)

```
<div id="main">
    <h1 id="title">Hello World!</h1>
    <p id="hint">This is a simple paragraph.</p>
</div>

<script>
var parentElem = document.getElementById("main");
var childElem = document.getElementById("hint");
parentElem.removeChild(childElem);
</script>
```

Αφαίρεση υπαρχόντων στοιχείων από το DOM
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Μπορείτε επίσης να αφαιρέσετε το θυγατρικό στοιχείο χωρίς να γνωρίζετε το γονικό στοιχείο. Μπορείτε να βρείτε το θυγατρικό στοιχείο και να χρησιμοποιήσετε την ιδιότητα parentNode για να βρείτε τον γονέα του. Θα επιστρέψει τον γονέα του δοσμένου κόμβου στο δέντρο Dom.

```
<div id="main">
    <h1 id="title">Hello World!</h1>
    <p id="hint">This is a simple paragraph.</p>
</div>

<script>
var childElem = document.getElementById("hint");
childElem.parentNode.removeChild(childElem);
</script>
```

Αφαίρεση υπαρχόντων στοιχείων από το DOM
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Αντικατάσταση υπαρχόντων στοιχείων στο DOM

Έχετε επίσης την επιλογή να αντικαταστήσετε ένα στοιχείο στο HTML DOM με ένα άλλο χρησιμοποιώντας τη μέθοδο replaceChild (). Η μέθοδος αυτή λαμβάνει δύο παραμέτρους: τον κόμβο που πρόκειται να εισαχθεί και τον κόμβο που πρόκειται να

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικρότηση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

αντικατασταθεί. Η σύνταξη χρησιμοποιείται ως εξής: parentNode.replaceChild (newChild, oldChild);

```
<div id="main">
    <h1 id="title">Hello World!</h1>
    <p id="hint">This is a simple paragraph.</p>
</div>

<script>
var parentElem = document.getElementById("main");
var oldPara = document.getElementById("hint");

// Creating new element
var newPara = document.createElement("p");
var newContent = document.createTextNode("This is a new paragraph.");
newPara.appendChild(newContent);

// Replacing old paragraph with newly created paragraph
parentElem.replaceChild(newPara, oldPara);
</script>
```

Αντικατάσταση υπαρχόντων στοιχείων στο DOM
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Πλοήγηση μεταξύ κόμβων DOM

Μέχρι τώρα, θα πρέπει να σχηματίσουμε μια καλύτερη ιδέα για τον τρόπο με τον οποίο επιλέγονται τα επιμέρους στοιχεία σε μια ιστοσελίδα. Υπάρχουν πολλές περιπτώσεις όπου θα πρέπει να έχετε πρόσβαση σε στοιχεία θυγατρικών και γονικών κόμβων ή προγόνων (ancestor). Στην αρχή αυτού του υποκεφάλαιου, έχουμε αναφερθεί στους κόμβους (nodes) και τώρα θα δούμε πώς μπορούμε να έχουμε πρόσβαση στους διαφορετικούς τύπους κόμβων που υπάρχουν.

Οι DOM κόμβοι έχουν αρκετές ιδιότητες και μεθόδους που σας επιτρέπουν να περιηγηθείτε ή να διασχίσετε τη δομή του δέντρου DOM και να κάνετε τις απαραίτητες αλλαγές αρκετά εύκολα.

Πρόσβαση στους θυγατρικούς κόμβους

Οι ιδιότητες firstChild και lastChild σας επιτρέπουν να έχετε πρόσβαση στον πρώτο και τελευταίο άμεσο θυγατρικό κόμβο ενός κόμβου αντίστοιχα. Εάν ένας κόμβος δεν έχει θυγατρικό στοιχείο, θα επιστρέψει την τιμή null.

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικρότηση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI SOC-IN)

Ας ρίξουμε μια ματιά στο παρακάτω παράδειγμα:

```
<div id="main">
    <h1 id="title">My Heading</h1>
    <p id="hint"><span>This is some text.</span></p>
</div>

<script>
var main = document.getElementById("main");
console.log(main.firstChild.nodeName); // Prints: #text

var hint = document.getElementById("hint");
console.log(hint.firstChild.nodeName); // Prints: SPAN
</script>
```

Παράδειγμα πρόσβασης σε θυγατρικούς κόμβους
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

* Σημειώστε ότι ο κόμβος Name είναι μια ιδιότητα μόνο για ανάγνωση, η οποία επιστρέφει το όνομα του τρέχοντος κόμβου ως συμβολοσειρά. Για παράδειγμα, θα επιστρέψει το όνομα ετικέτας ενός κόμβου στοιχείου, #text για τον κόμβο κειμένου, #comment για τον κόμβο σχολίου, #document για τον κόμβο εγγράφου και ούτω καθεξής.

Στο παράδειγμα που μόλις είδαμε, το nodeName του πρώτου θυγατρικού κόμβου των κύριων στοιχείων DIV επέστρεψε το #text αντί για H1. Αυτό συμβαίνει επειδή ο κενός χώρος, δηλαδή τα διαστήματα (spaces), οι καρτέλες (tabs), οι νέες γραμμές (newlines) και ούτω καθεξής, θεωρούνται έγκυροι χαρακτήρες και γίνονται μέρος του δέντρου DOM με τη μορφή κόμβων #text. Στη συνέχεια, η ετικέτα που περιέχει μια νέα γραμμή πριν από το <h1> θα δημιουργήσει ένα #text κόμβο.

Για να αποτρέψετε αυτό το πρόβλημα με τους κόμβους firstChild και lastChild που επιστρέφουν τους κόβμους #text ή #comment, μπορείτε να χρησιμοποιήσετε τις ιδιότητες firstElementChild και lastElementChild ως εναλλακτική λύση. Αυτές οι ιδιότητες θα επιστρέψουν μόνο το πρώτο και το τελευταίο στοιχείο του κόμβου αντίστοιχα. Ωστόσο, αυτό δεν θα λειτουργήσει στο Internet Explorer πριν από την έκδοση 9.

Το παρακάτω παράδειγμα θα σας βοηθήσει να το κατανοήσετε καλύτερα:

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικρότηση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI SOC-IN)

```
<div id="main">
    <h1 id="title">My Heading</h1>
    <p id="hint"><span>This is some text.</span></p>
</div>

<script>
var main = document.getElementById("main");
alert(main.firstElementChild.nodeName); // Outputs: H1
main.firstElementChild.style.color = "red";

var hint = document.getElementById("hint");
alert(hint.firstElementChild.nodeName); // Outputs: SPAN
hint.firstElementChild.style.color = "blue";
</script>
```

Παράδειγμα Πρόσβασης σε Θυγατρικούς Κόμβους
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Για να αποκτήσετε πρόσβαση σε όλους τους θυγατρικούς κόμβους ενός δεδομένου στοιχείου, μπορείτε επίσης να χρησιμοποιήσετε την ιδιότητα childNodes. Λάβετε υπόψη ότι στον πρώτο θυγατρικό κόμβο προσαρτάται ο δείκτης 0.

Παρατηρήστε το παράδειγμα που παρατίθεται πιο κάτω:

```
<div id="main">
    <h1 id="title">My Heading</h1>
    <p id="hint"><span>This is some text.</span></p>
</div>

<script>
var main = document.getElementById("main");

// First check that the element has child nodes
if(main.hasChildNodes()) {
    var nodes = main.childNodes;

    // Loop through node list and display node name
    for(var i = 0; i < nodes.length; i++) {
        alert(nodes[i].nodeName);
    }
}
</script>
```

Παράδειγμα Πρόσβασης στους Θυγατρικούς Κόμβους
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικύρωση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI SOC-IN)

Εδώ, η ιδιότητα childNodes επιστρέφει *όλους τους θυγατρικούς κόμβους*, συμπεριλαμβανομένων των μη στοιχειακών κόμβων όπως οι κόμβοι κειμένου και σχολίων.

Αν θέλετε να λάβετε *μια συλλογή μόνο από στοιχεία*, θα πρέπει να χρησιμοποιήσετε την ιδιότητα των θυγατρικών κόμβων.

Ας δούμε ένα παράδειγμα για να καταλάβουμε πώς χρησιμοποιείται αυτή ιδιότητα:

```
<div id="main">
    <h1 id="title">My Heading</h1>
    <p id="hint"><span>This is some text.</span></p>
</div>

<script>
var main = document.getElementById("main");

// First check that the element has child nodes
if(main.hasChildNodes()) {
    var nodes = main.children;

    // Loop through node list and display node name
    for(var i = 0; i < nodes.length; i++) {
        alert(nodes[i].nodeName);
    }
}
</script>
```

Παράδειγμα Πρόσβασης στους Θυγατρικούς Κόμβους
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Πρόσβαση στους Γονικούς Κόμβους

Για να αποκτήσετε πρόσβαση στον γονικό κόμβο ενός συγκεκριμένου κόμβου στο δέντρο Dom, μπορείτε να χρησιμοποιήσετε την ιδιότητα parentNode.

* Σημειώστε ότι η ιδιότητα parentNode θα επιστρέφει πάντα μηδενικές τιμές (null values) για κόμβους εγγράφου επειδή δεν έχουν γονικές σχέσεις.

Στο παρακάτω παράδειγμα, μπορείτε να δείτε πώς χρησιμοποιείται η ιδιότητα parentNode:

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικύρωση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

```
<div id="main">
    <h1 id="title">My Heading</h1>
    <p id="hint"><span>This is some text.</span></p>
</div>

<script>
var hint = document.getElementById("hint");
alert(hint.parentNode.nodeName); // Outputs: DIV
alert(document.documentElement.parentNode.nodeName); // Outputs: #document
alert(document.parentNode); // Outputs: null
</script>
```

Παράδειγμα Πρόσβασης σε Γονικούς Κόμβους
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Είναι καλό να γνωρίζετε ότι οι κόμβοι που βρίσκονται στην κορυφή ενός δέντρου DOM μπορούν να καταστούν προσβάσιμοι απευθείας ως ιδιότητες εγγράφου (document properties). Έχουμε δει κάποια παραδείγματα σχετικά με τους κόμβους που βρίσκονται στην κορυφή ενός δέντρου DOM σε προηγούμενες ενότητες όπως το <html> στοιχείο, το οποίο μπορεί να καταστεί προσβάσιμο με την ιδιότητα document.documentElement. Επίσης, το <head> στοιχείο μπορεί να καταστεί προσβάσιμο με την ιδιότητα document.head και το <body> στοιχείο με την ιδιότητα document.body.

Υπάρχει επίσης η επιλογή να λάβετε μόνο κόμβους στοιχείων με την ιδιότητα parentElement, όπως φαίνεται στο παρακάτω παράδειγμα:

```
<div id="main">
    <h1 id="title">My Heading</h1>
    <p id="hint"><span>This is some text.</span></p>
</div>

<script>
var hint = document.getElementById("hint");
alert(hint.parentNode.nodeName); // Outputs: DIV
hint.parentNode.style.backgroundColor = "yellow";
</script>
```

Παράδειγμα Πρόσβασης στους Γονικούς Κόμβους
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικύρωση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

Πρόσβαση στους Αμφιθαλείς Κόμβους (ή κόμβοι αδέλφια/ Sibling Nodes)

Για να αποκτήσετε πρόσβαση στον προηγούμενο και τον επόμενο κόμβο σε ένα δέντρο Dom, μπορείτε να χρησιμοποιήσετε τις προηγούμενες ιδιότητες previousSibling και nextSibling αντίστοιχα.

Ας δούμε ένα παράδειγμα:

```
<div id="main">
    <h1 id="title">My Heading</h1>
    <p id="hint"><span>This is some text.</span></p><hr>
</div>

<script>
var title = document.getElementById("title");
alert(title.previousSibling.nodeName); // Outputs: #text

var hint = document.getElementById("hint");
alert(hint.nextSibling.nodeName); // Outputs: HR
</script>
```

Παράδειγμα Πρόσβασης στους Αμφιθαλείς Κόμβους
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Για να παρακάμψετε οποιουσδήποτε κόμβους κενών κειμένων, μπορείτε να χρησιμοποιήσετε τις δηλώσεις previousElementSibling και nextElementSibling ως εναλλακτικές λύσεις για να λάβετε τα προηγούμενα και επόμενα στοιχεία αμφιθαλών κόμβων. Εάν δεν βρεθεί ένας τέτοιος κόμβος αδελφός, αυτές οι ιδιότητες θα επιστρέψουν μηδενικές τιμές.

Ας δούμε το ακόλουθο παράδειγμα παρακάτω:

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικύρωση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

```html
<div id="main">
    <h1 id="title">My Heading</h1>
    <p id="hint"><span>This is some text.</span></p>
</div>

<script>
var hint = document.getElementById("hint");
alert(hint.previousElementSibling.nodeName); // Outputs: H1
alert(hint.previousElementSibling.textContent); // Outputs: My Heading

var title = document.getElementById("title");
alert(title.nextElementSibling.nodeName); // Outputs: P
alert(title.nextElementSibling.textContent); // Outputs: This is some text.
</script>
```

Παράδειγμα Πρόσβασης στους Αμφιθαλείς Κόμβους
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Η ιδιότητα textContent που χρησιμοποιείται εδώ δηλώνει το περιεχόμενο του κειμένου ενός κόμβου και όλων των απογόνων του.

Τύποι Κόμβων DOM

Το δέντρο DOM αποτελείται από διαφορετικούς τύπους κόμβων που περιλαμβάνει στοιχεία, κείμενο, σχόλια και πολλά άλλα.

Κάθε κόμβος έχει μια ιδιότητα nodeType που μπορεί να σας βοηθήσει να καταλάβετε πώς μπορείτε να έχετε πρόσβαση και να επεξεργάζεστε τον εν λόγω κόμβο. Ο παρακάτω πίνακας παρέχει μια λίστα με τους πιο σημαντικούς και συχνά χρησιμοποιημένους τύπους κόμβων που πρέπει να γνωρίζει κανείς:

Constant	Value	Description
ELEMENT_NODE	1	An element node such as `<p>` or `<img>`
TEXT_NODE	3	The actual text of element
COMMENT_NODE	8	A comment node i.e. `<!-- some comment -->`
DOCUMENT_NODE	9	A document node i.e. the parent of `<html>` element
DOCUMENT_TYPE_NODE	10	A document type node e.g. `<!DOCTYPE html>` for HTML5 documents

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικρότηση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

Πίνακας των πιο κοινών τύπων κόμβων DOM
(Πηγή: https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php)

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά επικρότηση του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI SOC-IN)