



Με συγχρηματοδότηση από το
πρόγραμμα «Erasmus+»
της Ευρωπαϊκής Ένωσης

</>
code4sp
coding for social promotion

Εκπαιδευτικό Υλικό JavaScript Υποενότητα 2 – JavaScript & DOM

WP3: Εκπαιδευτικό Υλικό του Έργου Code4SP

Εκπονήθηκε από το:



CITIZENS
IN POWER



Center for Social
Innovation



ZAUG
gGmbH



social
hackers
academy





Υποκεφάλαιο 2: JavaScript & DOM



Με συγχρηματοδότηση από το
πρόγραμμα «Erasmus+»
της Ευρωπαϊκής Ένωσης

Τι είναι το Μοντέλο Αντικειμένων Εγγράφου (DOM);

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

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 καταχωρίσεις των αντικειμένων σε αυτό οργανώνονται ιεραρχικά. Τα αντικείμενα αυτά περιλαμβάνουν όλα τα μέρη ενός εγγράφου HTML όπως στοιχεία, ιδιότητες, κείμενο κ.λπ.

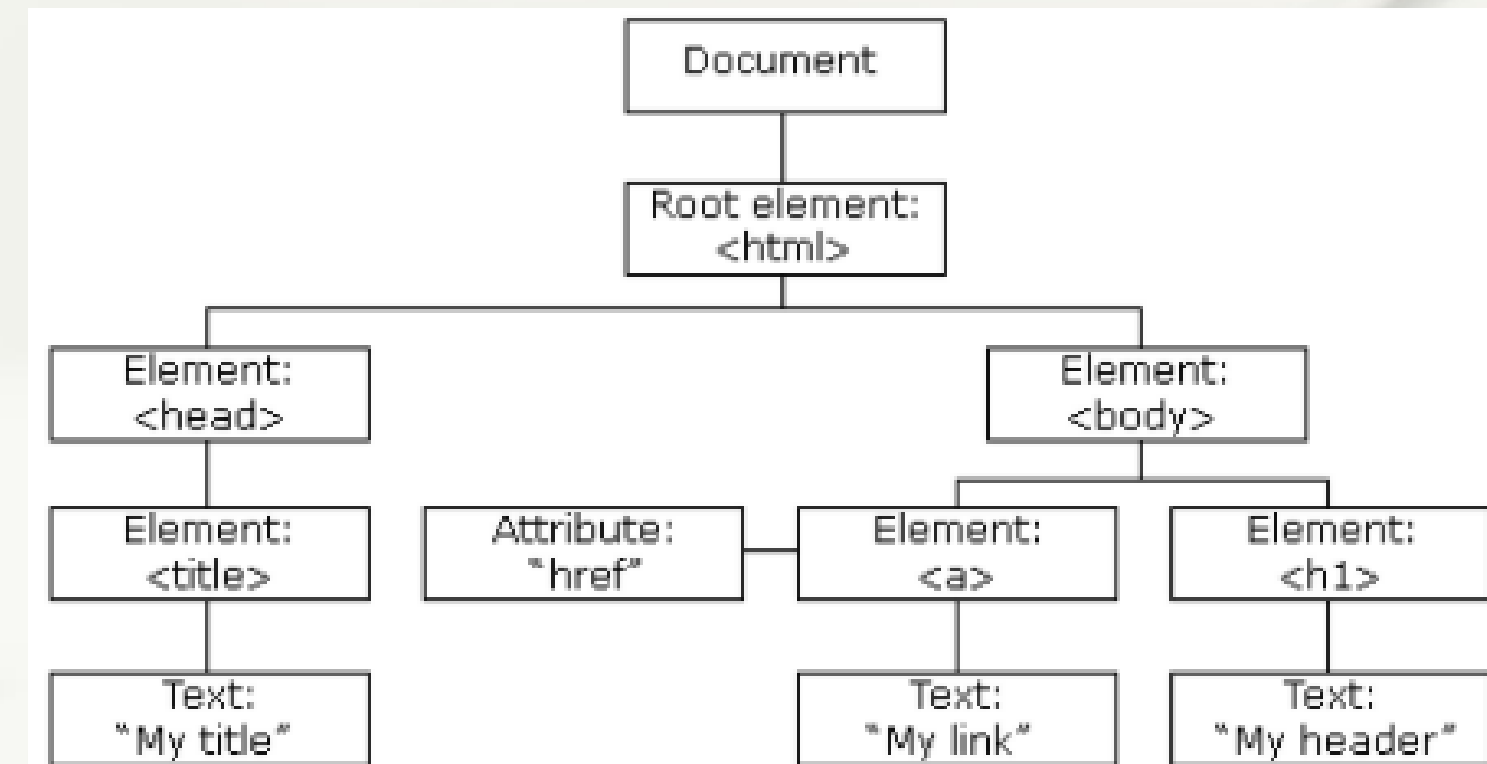


Figure 1 – HTML DOM Tree of Objects
(Source: https://www.w3schools.com/js/js_htmlDOM.asp)

Τι μπορεί να κάνει η JavaScript μέσα στο HTML DOM?

Μπορεί να κάνει τα ακόλουθα:

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

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 HTMLστοιχείο στόχος (target element).

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



Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης



Η επιλογή των στοιχείων που βρίσκονται στην κορυφή του δέντρου

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`.



Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Η επιλογή των στοιχείων που βρίσκονται στην κορυφή του δέντρου

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>JS Select Topmost Elements</title>
6 </head>
7 <body>
8   <script>
9     // Display lang attribute value of html element
10    alert(document.documentElement.getAttribute("lang")); // Outputs: en
11
12    // Set background color of body element
13    document.body.style.background = "yellow";
14
15    // Display tag name of the head element's first child
16    alert(document.head.firstChild.nodeName); // Outputs: meta
17  </script>
18 </body>
19 </html>
```

* 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`.

Παράδειγμα στοιχείων που βρίσκονται στην κορυφή του δέντρου

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php>)



Με συγχρηματοδότηση από το
πρόγραμμα «Erasmus+»
της Ευρωπαϊκής Ένωσης

Η επιλογή των στοιχείων που βρίσκονται στην κορυφή

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>JS Document.body Demo</title>
6   <script>
7     alert("From HEAD: " + document.body); // Outputs: null (since <body> is not
parsed yet)
8   </script>
9 </head>
10 <body>
11   <script>
12     alert("From BODY: " + document.body); // Outputs: HTMLBodyElement
13   </script>
14 </body>
15 </html>
```

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

Παράδειγμα στοιχείων που βρίσκονται στην κορυφή

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php>)



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 ()`.

Η μέθοδος `getElementById ()` χρησιμοποιείται για την επιστροφή του στοιχείου ως αντικείμενου 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.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>JS Select Element by ID</title>
6 </head>
7 <body>
8   <p id="mark">This is a paragraph of text.</p>
9   <p>This is another paragraph of text.</p>
10
11 <script>
12   // Selecting element with id mark
13   var match = document.getElementById("mark");
14
15   // Highlighting element's background
16   match.style.background = "yellow";
17 </script>
18 </body>
19 </html>
```

Η επιλογή στοιχείων με αναγνωριστικό (ID)

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php>)



Επιλογή στοιχείων ανά όνομα κλάσης

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>JS Select Elements by Class Name</title>
6 </head>
7 <body>
8   <p class="test">This is a paragraph of text.</p>
9   <div class="block test">This is another paragraph of text.</div>
10  <p>This is one more paragraph of text.</p>
11
12  <script>
13    // Selecting elements with class test
14    var matches = document.getElementsByClassName("test");
15
16    // Displaying the selected elements count
17    document.write("Number of selected elements: " + matches.length);
18
19    // Applying bold style to first element in selection
20    matches[0].style.fontWeight = "bold";
21
22    // Applying italic style to last element in selection
23    matches[matches.length - 1].style.fontStyle = "italic";
24
25    // Highlighting each element's background through loop
26    for(var elem in matches) {
27      matches[elem].style.background = "yellow";
28    }
29  </script>
30 </body>
31 </html>
```

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.

Επιλογή στοιχείων ανά όνομα κλάσης

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-selectors.php>)



Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Επιλογή στοιχείων ανά όνομα ετικέτας

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>JS Select Elements by Tag Name</title>
6 </head>
7 <body>
8   <p>This is a paragraph of text.</p>
9   <div class="test">This is another paragraph of text.</div>
10  <p>This is one more paragraph of text.</p>
11
12  <script>
13    // selecting all paragraph elements
14    var matches = document.getElementsByTagName("p");
15
16    // Printing the number of selected paragraphs
17    document.write("Number of selected elements: " + matches.length);
18
19    // Highlighting each paragraph's background through loop
20    for(var elem in matches) {
21      matches[elem].style.background = "yellow";
22    }
23  </script>
24 </body>
25 </html>
```

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.

Επιλογή στοιχείων ανά όνομα κλάσης

(Πηγή: <https://www.tutorialpublic.com/javascript-tutorial/javascript-dom-selectors.php>)



Με συγχρηματοδότηση από το
πρόγραμμα «Erasmus+»
της Ευρωπαϊκής Ένωσης

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 επιλογή CSS, μπορείτε να χρησιμοποιήσετε τη μέθοδο `querySelectorAll()`.

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

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>JS Select Elements with CSS Selectors</title>
6 </head>
7 <body>
8   <ul>
9     <li>Bread</li>
10    <li class="tick">Coffee</li>
11    <li>Pineapple Cake</li>
12  </ul>
13
14  <script>
15    // Selecting all li elements
16    var matches = document.querySelectorAll("ul li");
17
18    // Printing the number of selected li elements
19    document.write("Number of selected elements: " + matches.length + "<hr>");
20
21    // Printing the content of selected li elements
22    for(var elem of matches) {
23      document.write(elem.innerHTML + "<br>");
24    }
25
26    // Applying line through style to first li element with class tick
27    matches = document.querySelectorAll("ul li.tick");
28    matches[0].style.textDecoration = "line-through";
29  </script>
30 </body>
31 </html>
```

Παράδειγμα επιλογής στοιχείων με επιλογείς CSS

(Πηγή: <https://www.tutorialpublic.com/javascript-tutorial/javascript-dom-selectors.php>)



Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Στυλιστική Ανάπτυξη στοιχείων DOM JavaScript

Η JavaScript παρέχει δυναμικούς τρόπους με τους οποίους μπορείται 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.



Εφαρμογή ενσωματωμένων στυλ σε στοιχεία (Inline Styles)

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>JS Set Inline Styles Demo</title>
6 </head>
7 <body>
8   <p id="intro">This is a paragraph.</p>
9   <p>This is another paragraph.</p>
10
11 <script>
12 // Selecting element
13 var elem = document.getElementById("intro");
14
15 // Applying styles on element
16 elem.style.color = "blue";
17 elem.style.fontSize = "18px";
18 elem.style.fontWeight = "bold";
19 </script>
20 </body>
21 </html>
```

Παράδειγμα εφαρμογής ενσωματωμένων στυλ σε στοιχεία

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-styling.php>)

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":





Συμβάσεις Ονοματολογίας Ιδιοτήτων CSS στην JavaScript

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

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.



Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Λήψη πληροφοριών στυλ από στοιχεία

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

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>JS Get Element's Style Demo</title>
6 </head>
7 <body>
8   <p id="intro" style="color:red; font-size:20px;">This is a paragraph.</p>
9   <p>This is another paragraph.</p>
10
11 <script>
12 // Selecting element
13 var elem = document.getElementById("intro");
14
15 // Getting style information from element
16 alert(elem.style.color); // Outputs: red
17 alert(elem.style.fontSize); // Outputs: 20px
18 alert(elem.style.fontStyle); // Outputs nothing
19 </script>
20 </body>
21 </html>
```

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.

Λήψη πληροφοριών στυλ από στοιχεία

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>JS Get Computed Style Demo</title>
6 <style type="text/css">
7   #intro {
8     font-weight: bold;
9     font-style: italic;
10  }
11 </style>
12 </head>
13 <body>
14   <p id="intro" style="color:red; font-size:20px;">This is a paragraph.</p>
15   <p>This is another paragraph.</p>
16
17   <script>
18     // Selecting element
19     var elem = document.getElementById("intro");
20
21     // Getting computed style information
22     var styles = window.getComputedStyle(elem);
23
24     alert(styles.getPropertyValue("color")); // Outputs: rgb(255, 0, 0)
25     alert(styles.getPropertyValue("font-size")); // Outputs: 20px
26     alert(styles.getPropertyValue("font-weight")); // Outputs: 700
27     alert(styles.getPropertyValue("font-style")); // Outputs: italic
28   </script>
29 </body>
30 </html>
```

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:

* 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.

`window.getComputedStyle()` – Παράδειγμα Μεθόδου

(Μέθοδος: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-styling.php>)



Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Προσθήκη κλάσεων CSS στα στοιχεία

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <title>JS Add or Replace CSS Classes Demo</title>
6 <style>
7   .highlight {
8     background: yellow;
9   }
10 </style>
11 </head>
12 <body>
13   <div id="info" class="disabled">Something very important!</div>
14
15   <script>
16     // Selecting element
17     var elem = document.getElementById("info");
18
19     elem.className = "note"; // Add or replace all classes with note class
20     elem.className += " highlight"; // Add a new class highlight
21   </script>
22 </body>
23 </html>
```

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.

Παράδειγμα προσθήκης κλάσεων CSS στα στοιχεία

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-styling.php>)



Με συγχρηματοδότηση από το
πρόγραμμα «Erasmus+»
της Ευρωπαϊκής Ένωσης

Προσθήκη κλάσεων CSS στα στοιχεία

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <title>JS classList Demo</title>
6 <style>
7   .highlight {
8     background: yellow;
9   }
10 </style>
11 </head>
12 <body>
13   <div id="info" class="disabled">Something very important!</div>
14
15   <script>
16     // Selecting element
17     var elem = document.getElementById("info");
18
19     elem.classList.add("hide"); // Add a new class
20     elem.classList.add("note", "highlight"); // Add multiple classes
21     elem.classList.remove("hide"); // Remove a class
22     elem.classList.remove("disabled", "note"); // Remove multiple classes
23     elem.classList.toggle("visible"); // If class exists remove it, if not add it
24
25     // Determine if class exist
26     if(elem.classList.contains("highlight")) {
27       alert("The specified class exists on the element.");
28     }
29   </script>
30 </body>
31 </html>
```

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.

classList property – Παράδειγμα προσθήκης κλάσεων CSS στα στοιχεία

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-styling.php>)



Με συγχρηματοδότηση από το
πρόγραμμα «Erasmus+»
της Ευρωπαϊκής Ένωσης

Δουλεύοντας με τις ιδιότητες

Οι ιδιότητες είναι ειδικές λέξεις που χρησιμοποιούνται μέσα στην ετικέτα έναρξης/εκκίνησης 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 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`.

```
1 <a href="https://www.google.com/" target="_blank" id="myLink">Google</a>
2
3 <script>
4     // Selecting the element by ID attribute
5     var link = document.getElementById("myLink");
6
7     // Getting the attributes values
8     var href = link.getAttribute("href");
9     alert(href); // Outputs: https://www.google.com/
10
11     var target = link.getAttribute("target");
12     alert(target); // Outputs: _blank
13 </script>
```

Η μέθοδος `getAttribute()` – Παράδειγμα λήψης τιμής ιδιότητας ενός 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.

```
1 <button type="button" id="myBtn">Click Me</button>
2
3 <script>
4     // selecting the element
5     var btn = document.getElementById("myBtn");
6
7     // Setting new attributes
8     btn.setAttribute("class", "click-btn");
9     btn.setAttribute("disabled", "");
10 </script>
```

Η μέθοδος `setAttribute()` – Παράδειγμα ορισμού ιδιοτήτων στα στοιχεία

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-get-set-attributes.php>)



Με συγχρηματοδότηση από το
πρόγραμμα «Erasmus+»
της Ευρωπαϊκής Ένωσης

Ορισμός ιδιοτήτων στα στοιχεία

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 μέσα σε οξυγώνιες αγκύλες (`<a>`):

```
1 <a href="#" id="myLink">Tutorial Republic</a>
2
3 <script>
4     // Selecting the element
5     var link = document.getElementById("myLink");
6
7     // Changing the href attribute value
8     link.setAttribute("href", "https://www.tutorialrepublic.com");
9 </script>
```

Η μέθοδος `setAttribute()` – Παράδειγμα ορισμού ιδιοτήτων στα στοιχεία

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-get-set-attributes.php>)



Με συγχρηματοδότηση από το
πρόγραμμα «Erasmus+»
της Ευρωπαϊκής Ένωσης

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 μέσα στις οξυγώνιες αγκύλες. Τώρα θα το αφαιρέσουμε στο ακόλουθο παράδειγμα:

```
1 <a href="https://www.google.com/" id="myLink">Google</a>
2
3 <script>
4     // Selecting the element
5     var link = document.getElementById("myLink");
6
7     // Removing the href attribute
8     link.removeAttribute("href");
9 </script>
```

Η μέθοδος `removeAttribute()` – Παράδειγμα αφαίρεσης ιδιοτήτων στα στοιχεία
(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-get-set-attributes.php>)



Με συγχρηματοδότηση από το
πρόγραμμα «Erasmus+»
της Ευρωπαϊκής Ένωσης

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

```
1 <div id="main">
2   <h1 id="title">Hello World!</h1>
3   <p id="hint">This is a simple paragraph.</p>
4 </div>
5
6 <script>
7 // Creating a new div element
8 var newDiv = document.createElement("div");
9
10 // Creating a text node
11 var newContent = document.createTextNode("Hi, how are you doing?");
12
13 // Adding the text node to the newly created div
14 newDiv.appendChild(newContent);
15
16 // Adding the newly created element and its content into the DOM
17 var currentDiv = document.getElementById("main");
18 document.body.appendChild(newDiv, currentDiv);
19 </script>
```

Παράδειγμα προσθήκης νέων στοιχείων στο DOM

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-manipulation.php>)

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. 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.

Προσθήκη νέων στοιχείων στο DOM

```
1 <div id="main">
2   <h1 id="title">Hello World!</h1>
3   <p id="hint">This is a simple paragraph.</p>
4 </div>
5
6 <script>
7   // Creating a new div element
8   var newDiv = document.createElement("div");
9
10  // Creating a text node
11  var newContent = document.createTextNode("Hi, how are you doing?");
12
13  // Adding the text node to the newly created div
14  newDiv.appendChild(newContent);
15
16  // Adding the newly created element and its content into the DOM
17  var currentDiv = document.getElementById("main");
18  document.body.insertBefore(newDiv, currentDiv);
19 </script>
```

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

Παράδειγμα προσθήκης νέων στοιχείων στο DOM

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-manipulation.php>)



Λήψη ή ορισμός 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.

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.

```
1 <div id="main">
2   <h1 id="title">Hello World!</h1>
3   <p id="hint">This is a simple paragraph.</p>
4 </div>
5
6 <script>
7 // Getting inner HTML contents
8 var contents = document.getElementById("main").innerHTML;
9 alert(contents); // Outputs inner html contents
10
11 // Setting inner HTML contents
12 var mainDiv = document.getElementById("main");
13 mainDiv.innerHTML = "<p>This is <em>newly inserted</em> paragraph.</p>";
14 </script>
```

Παράδειγμα λήψη ή ορισμός περιεχομένων HTML σε DOM

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-manipulation.php>)

Λήψη ή Ορισμός Περιεχομένων HTML σε DOM

```
1 <!-- beforebegin -->
2 <div id="main">
3   <!-- afterbegin -->
4   <h1 id="title">Hello World!</h1>
5   <!-- beforeend -->
6 </div>
7 <!-- afterend -->
8
9 <script>
10 // Selecting target element
11 var mainDiv = document.getElementById("main");
12
13 // Inserting HTML just before the element itself, as a previous sibling
14 mainDiv.insertAdjacentHTML('beforebegin', '<p>This is paragraph one.</p>');
15
16 // Inserting HTML just inside the element, before its first child
17 mainDiv.insertAdjacentHTML('afterbegin', '<p>This is paragraph two.</p>');
18
19 // Inserting HTML just inside the element, after its last child
20 mainDiv.insertAdjacentHTML('beforeend', '<p>This is paragraph three.</p>');
21
22 // Inserting HTML just after the element itself, as a next sibling
23 mainDiv.insertAdjacentHTML('afterend', '<p>This is paragraph four.</p>');
24 </script>
```

Παράδειγμα Λήψης ή Ορισμού Περιεχομένων HTML σε DOM

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-manipulation.php>)

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: "beforebegin", "afterbegin", "beforeend" και "afterend". Είναι επίσης σημαντικό να σημειωθεί ότι αυτή η μέθοδος υποστηρίζεται από όλα τα μεγάλα προγράμματα περιήγησης.



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.

```
1 <div id="main">
2   <h1 id="title">Hello World!</h1>
3   <p id="hint">This is a simple paragraph.</p>
4 </div>
5
6 <script>
7 var parentElem = document.getElementById("main");
8 var childElem = document.getElementById("hint");
9 parentElem.removeChild(childElem);
10 </script>
```

Παράδειγμα αφαίρεσης υπαρχόντων στοιχείων από το DOM

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-manipulation.php>)



Αφαίρεση Υπαρχόντων Στοιχείων από το DOM

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.

```
1 <div id="main">
2   <h1 id="title">Hello World!</h1>
3   <p id="hint">This is a simple paragraph.</p>
4 </div>
5
6 <script>
7 var childElem = document.getElementById("hint");
8 childElem.parentNode.removeChild(childElem);
9 </script>
```

Παράδειγμα Αφαίρεσης Υπαρχόντων Στοιχείων από το DOM

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-manipulation.php>)



Αντικατάσταση Υπαρχόντων Στοιχείων στο DOM

```
1 <div id="main">
2   <h1 id="title">Hello World!</h1>
3   <p id="hint">This is a simple paragraph.</p>
4 </div>
5
6 <script>
7 var parentElem = document.getElementById("main");
8 var oldPara = document.getElementById("hint");
9
10 // Creating new element
11 var newPara = document.createElement("p");
12 var newContent = document.createTextNode("This is a new paragraph.");
13 newPara.appendChild(newContent);
14
15 // Replacing old paragraph with newly created paragraph
16 parentElem.replaceChild(newPara, oldPara);
17 </script>
```

Παράδειγμα Αντικατάστασης Υπαρχόντων Στοιχείων στο DOM

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-manipulation.php>)

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 αντικατασταθεί.

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



Πλοήγηση μεταξύ Κόμβων DOM

Μέχρι τώρα, θα πρέπει να σχηματίσατε 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 τον τελευταίο άμεσο θυγατρικό κόμβο ενός κόμβου αντίστοιχα. Εάν ένας κόμβος δεν έχει θυγατρικό στοιχείο, θα επιστρέψει την τιμή `null`.

```
1 <div id="main">
2   <h1 id="title">My Heading</h1>
3   <p id="hint"><span>This is some text.</span></p>
4 </div>
5
6 <script>
7 var main = document.getElementById("main");
8 console.log(main.firstChild.nodeName); // Prints: #text
9
10 var hint = document.getElementById("hint");
11 console.log(hint.firstChild.nodeName); // Prints: SPAN
12 </script>
```

* 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.

Παράδειγμα Πρόσβασης στους Θυγατρικούς Κόμβους

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-navigation.php>)



Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Πρόσβαση στους Θυγατρικούς Κόμβους

```
1 <div id="main">
2   <h1 id="title">My Heading</h1>
3   <p id="hint"><span>This is some text.</span></p>
4 </div>
5
6 <script>
7 var main = document.getElementById("main");
8 console.log(main.firstChild.nodeName); // Prints: #text
9
10 var hint = document.getElementById("hint");
11 console.log(hint.firstChild.nodeName); // Prints: SPAN
12 </script>
```

Παράδειγμα Πρόσβασης στους Θυγατρικούς Κόμβους

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-navigation.php>)

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 `<div>` ετικέτα που περιέχει μια νέα γραμμή πριν από το `<h1>` θα δημιουργήσει ένα `#text` κόμβο.



Πρόσβαση στους Θυγατρικούς Κόμβους

```
1 <div id="main">
2   <h1 id="title">My Heading</h1>
3   <p id="hint"><span>This is some text.</span></p>
4 </div>
5
6 <script>
7 var main = document.getElementById("main");
8 alert(main.firstChild.nodeName); // Outputs: H1
9 main.firstChild.style.color = "red";
10
11 var hint = document.getElementById("hint");
12 alert(hint.firstChild.nodeName); // Outputs: SPAN
13 hint.firstChild.style.color = "blue";
14 </script>
```

Παράδειγμα Πρόσβασης στους θυγατρικούς κόμβους

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-navigation.php>)

Για να αποτρέψετε αυτό το πρόβλημα με τους κόμβους `firstChild` και `lastChild` που επιστρέφουν τους κόμβους `#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.



Πρόσβαση στους Θυγατρικούς Κόμβους

```
1 <div id="main">
2   <h1 id="title">My Heading</h1>
3   <p id="hint"><span>This is some text.</span></p>
4 </div>
5
6 <script>
7 var main = document.getElementById("main");
8
9 // First check that the element has child nodes
10 if(main.hasChildNodes()) {
11   var nodes = main.childNodes;
12
13   // Loop through node list and display node name
14   for(var i = 0; i < nodes.length; i++) {
15     alert(nodes[i].nodeName);
16   }
17 }
18 </script>
```

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.

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

Παράδειγμα Πρόσβασης στους Θυγατρικούς Κόμβους

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-navigation.php>)



Με συγχρηματοδότηση από το
πρόγραμμα «Erasmus+»
της Ευρωπαϊκής Ένωσης

Πρόσβαση στους Θυγατρικούς Κόμβους

```
1 <div id="main">
2   <h1 id="title">My Heading</h1>
3   <p id="hint"><span>This is some text.</span></p>
4 </div>
5
6 <script>
7 var main = document.getElementById("main");
8
9 // First check that the element has child nodes
10 if(main.hasChildNodes()) {
11   var nodes = main.children;
12
13   // Loop through node list and display node name
14   for(var i = 0; i < nodes.length; i++) {
15     alert(nodes[i].nodeName);
16   }
17 }
18 </script>
```

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

Παράδειγμα Πρόσβασης στους Θυγατρικούς Κόμβους

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-navigation.php>)



Με συγχρηματοδότηση από το
πρόγραμμα «Erasmus+»
της Ευρωπαϊκής Ένωσης

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.

```
1 <div id="main">
2   <h1 id="title">My Heading</h1>
3   <p id="hint"><span>This is some text.</span></p>
4 </div>
5
6 <script>
7 var hint = document.getElementById("hint");
8 alert(hint.parentNode.nodeName); // Outputs: DIV
9 alert(document.documentElement.parentNode.nodeName); // Outputs: #document
10 alert(document.parentNode); // Outputs: null
11 </script>
```

Είναι καλό να γνωρίζουμε ότι **οι κόμβοι που βρίσκονται στην κορυφή ενός δέντρου DOM μπορούν να καταστούν προσβάσιμοι απευθείας ως ιδιότητες εγγράφου**, όπως το στοιχείο `<html>`, το οποίο μπορεί να καταστεί προσβάσιμο με την ιδιότητα `document.documentElement`.

Παράδειγμα Πρόσβασης στους Γονικούς Κόμβους

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-navigation.php>)



Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Πρόσβαση στους Γονικούς Κόμβους

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

```
1 <div id="main">
2   <h1 id="title">My Heading</h1>
3   <p id="hint"><span>This is some text.</span></p>
4 </div>
5
6 <script>
7 var hint = document.getElementById("hint");
8 alert(hint.parentNode.nodeName); // Outputs: DIV
9 hint.parentNode.style.backgroundColor = "yellow";
0 </script>
```

Παράδειγμα Πρόσβασης στους Γονικούς Κόμβους

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-navigation.php>)

Πρόσβαση στους Αμφιθαλείς Κόμβους (ή κόμβοι αδέρφια/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.

```
1 <div id="main">
2   <h1 id="title">My Heading</h1>
3   <p id="hint"><span>This is some text.</span></p><hr>
4 </div>
5
6 <script>
7 var title = document.getElementById("title");
8 alert(title.previousSibling.nodeName); // Outputs: #text
9
10 var hint = document.getElementById("hint");
11 alert(hint.nextSibling.nodeName); // Outputs: HR
12 </script>
```

Πρόσβαση στους Αμφιθαλείς Κόμβους

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-navigation.php>)

Πρόσβαση στους Αμφιθαλείς Κόμβους

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.

```
1 <div id="main">
2   <h1 id="title">My Heading</h1>
3   <p id="hint"><span>This is some text.</span></p>
4 </div>
5
6 <script>
7 var hint = document.getElementById("hint");
8 alert(hint.previousElementSibling.nodeName); // Outputs: H1
9 alert(hint.previousElementSibling.textContent); // Outputs: My Heading
10
11 var title = document.getElementById("title");
12 alert(title.nextElementSibling.nodeName); // Outputs: P
13 alert(title.nextElementSibling.textContent); // Outputs: This is some text.
14 </script>
```

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.

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

Πίνακας των πιο κοινών τύπων κόμβων DOM

(Πηγή: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-navigation.php>)



Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ας κάνουμε εξάσκηση!

Μέχρι στιγμής έχετε μάθει πολλά νέα πράγματα, επομένως τώρα πρέπει να τα εφαρμόσετε στην πράξη!

Για να ξεκινήσετε, ακολουθήστε είτε τον 1^ο είτε τον 2^ο σύνδεσμο:

- <https://www.w3resource.com/javascript-exercises/javascript-dom-exercises.php>
- <https://www.tutorialrepublic.com/javascript-examples.php>



ΕΥΧΑΡΙΣΤΟΥΜΕ!

ΕΠΟΜΕΝΟ ΚΕΦΑΛΑΙΟ: JavaScript & BOM



Με συγχρηματοδότηση από το
πρόγραμμα «Erasmus+»
της Ευρωπαϊκής Ένωσης