



JavaScript™

Τεχνολογίες προγραμματισμού στη
πλευρά του πελάτη
client side scripting technologies

Document **O**bject **M**odel

DOCUMENT OBJECT MODEL

Το DOM είναι ένα σετ από κανόνες που καθορίζει τον τρόπο που οι Browser πρέπει να δημιουργήσουν το μοντέλο μιας HTML σελίδας και πώς η Javascript μπορεί να αποκτήσει πρόσβαση και να ενημερώσει το περιεχόμενο μιας ιστοσελίδας.

Ακολουθείται από όλους τους κατασκευαστές Φυλλομετρητών και αποτελείται από δυο κύριες ομάδες:

- Κανόνες δημιουργίας του μοντέλου της σελίδας.
DOM tree.
- Κανόνες τροποποίησης της ιστοσελίδας

DOCUMENT OBJECT MODEL

Όταν ο Browser φορτώνει μια ιστοσελίδα, δημιουργεί ένα μοντέλο της σελίδας στη μνήμη. Το DOM ορίζει τον τρόπο με τον οποίο ο Browser πρέπει να δομήσει αυτό το μοντέλο χρησιμοποιώντας ένα DOM Tree.

Το DOM ονομάζεται μοντέλο αντικειμένου επειδή το μοντέλο (το δέντρο DOM) αποτελείται από αντικείμενα. Κάθε αντικείμενο αντιπροσωπεύει ένα διαφορετικό στοιχείο της σελίδας που έχει φορτωθεί στο παράθυρο του Browser.

Το DOM ορίζει επίσης μεθόδους (methods) και ιδιότητες (properties) για πρόσβαση και ενημέρωση κάθε αντικείμενου σε αυτό το μοντέλο, το οποίο με τη σειρά του ενημερώνεται για το τι βλέπει ο χρήστης στο πρόγραμμα περιήγησης.

DOCUMENT OBJECT MODEL

Όταν ο Browser φορτώνει μια ιστοσελίδα, δημιουργεί ένα μοντέλο της σελίδας στη μνήμη. Το DOM ορίζει τον τρόπο με τον οποίο ο Browser πρέπει να δομήσει αυτό το μοντέλο χρησιμοποιώντας ένα DOM Tree.

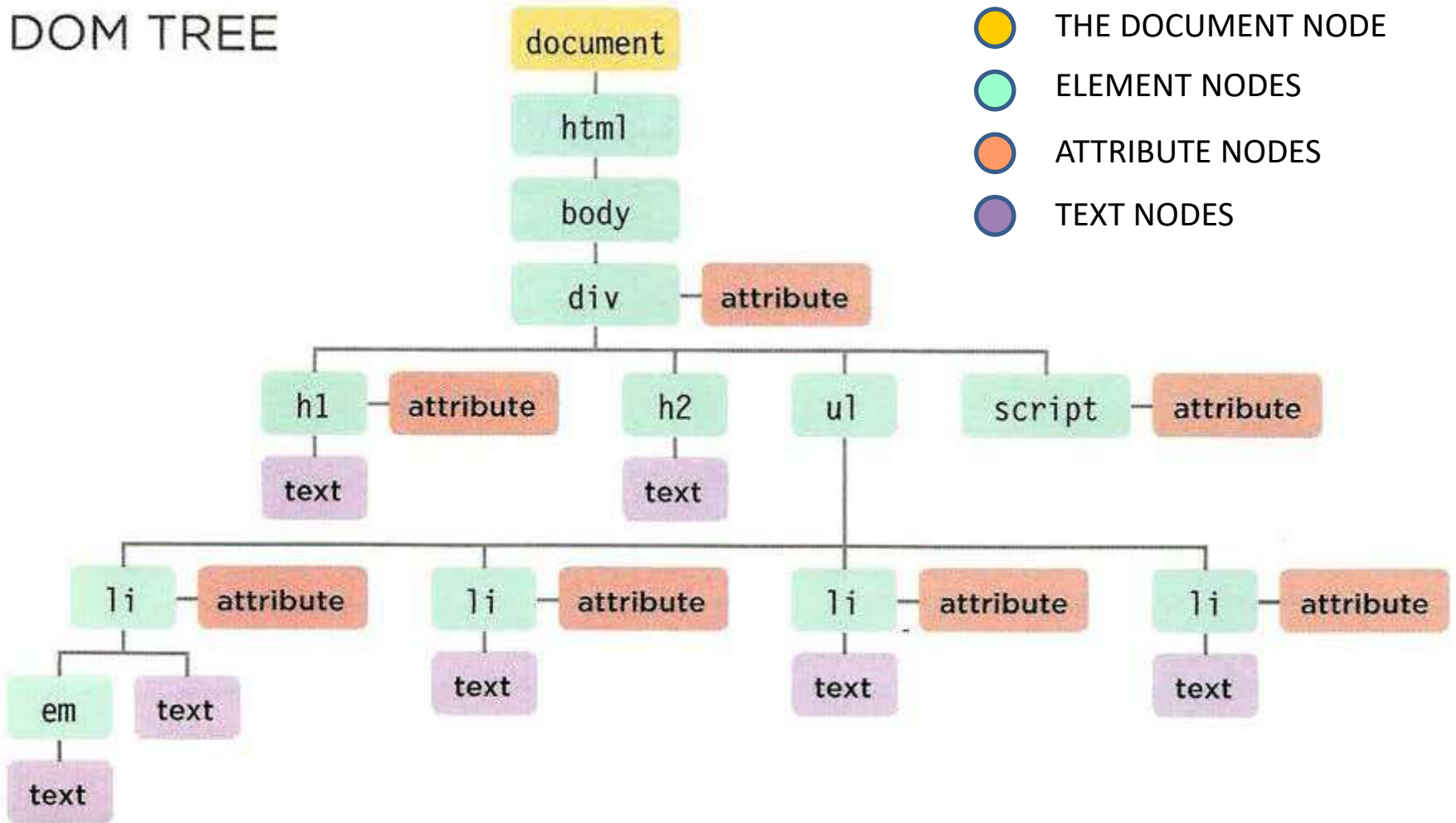
Το DOM ονομάζεται μοντέλο αντικειμένου επειδή το μοντέλο (το δέντρο DOM) αποτελείται από αντικείμενα. Κάθε αντικείμενο αντιπροσωπεύει ένα διαφορετικό στοιχείο της σελίδας που έχει φορτωθεί στο παράθυρο του Browser.

CREATING DOM

```
<html>
<body>
  <div id="page">
    <h1 id="header">List</h1>
    <h2>Buy groceries</h2>
    <ul>
      <li id="one" class="hot"><em>fresh</em> figs</li>
      <li id="two" class="hot">pine nuts</li>
      <li id="three" class="hot">honey</li>
      <li id="four">balsamic vinegar</li>
    </ul>
    <script src="js/list.js"></script>
  </div>
</body>
</html>
```

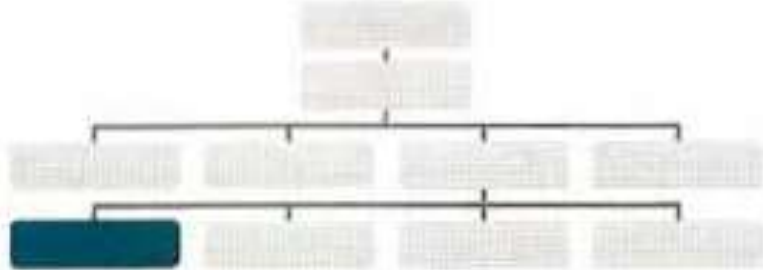
DOM Tree

DOM TREE



Κάθε κόμβος είναι ένα αντικείμενο με μεθόδους και ιδιότητες. Τα scripts έχουν πρόσβαση και μπορούν να ενημερώσουν αυτή τη δομή DOM (όχι το αρχείο HTML).

DOM Tree Εντοπισμός στοιχείου



Επιλογή κόμβου ενός μοναδικού στοιχείου

`getElementById()`

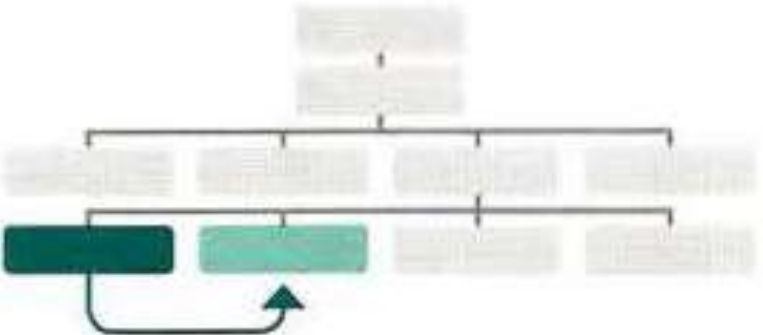
`querySelector()`



Επιλογή των κόμβων πολλαπλών στοιχείων

`getElementsByClassName()`

`getElementsByTagName()`



Μετακίνηση μεταξύ των σχετικών κόμβων του επιλεγμένου στοιχείου

`parentNode`

`previousSibling` / `nextSibling`

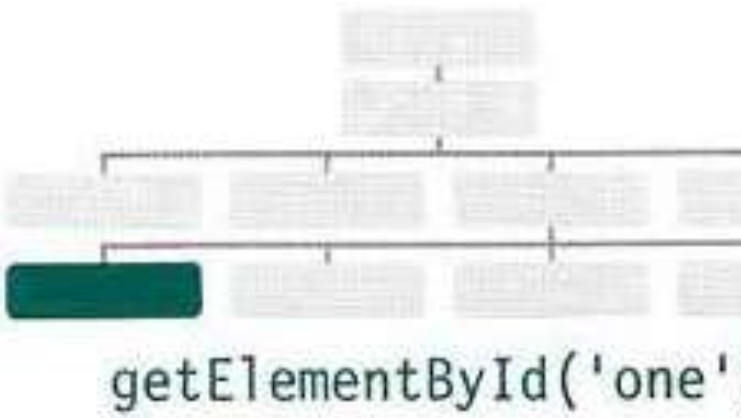
`firstChild` / `lastChild`

Επιλογή κόμβου ενός μοναδικού στοιχείου 1

getElementById('id')

```
getElementById("one");
```

```
var itemone = getElementById("one");
```



```
<html>
  <body>
    <div id="page">
      <h1 id="header">List</h1>
      <h2>Buy groceries</h2>
      <ul>
        <li id="one" class="hot"><em>fresh</em> figs</li>
        <li id="two" class="hot">pine nuts</li>
        <li id="three" class="hot">honey</li>
        <li id="four">balsamic vinegar</li>
      </ul>
      <script src="js/list.js"></script>
    </div>
  </body>
</html>
```

Επιλογή κόμβου ενός μοναδικού στοιχείου 2

querySelector('css selector')

Επιστρέφει μόνο το πρώτο στοιχείο που ταιριάζει

```
querySelector("li.hot");
```

```
var fhot = querySelector("li.hot");
```

```
<html>
  <body>
    <div id="page">
      <h1 id="header">List</h1>
      <h2>Buy groceries</h2>
      <ul>
        <li id="one" class="hot"><em>fresh</em> figs</li>
        <li id="two" class="hot">pine nuts</li>
        <li id="three" class="hot">honey</li>
        <li id="four">balsamic vinegar</li>
      </ul>
      <script src="js/list.js"></script>
    </div>
  </body>
</html>
```



querySelector('li.hot')

Επιλογή ενός ή περισσότερων κόμβων 1

getElementsByTagName('class')

getElementsByTagName("hot")

var Allhot = getElementsByTagName("hot")



```
<html>
  <body>
    <div id="page">
      <h1 id="header">List</h1>
      <h2>Buy groceries</h2>
      <ul>
        <li id="one" class="hot"><em>fresh</em> figs</li>
        <li id="two" class="hot">pine nuts</li>
        <li id="three" class="hot">honey</li>
        <li id="four">balsamic vinegar</li>
      </ul>
      <script src="js/list.js"></script>
    </div>
  </body>
</html>
```

getElementsByTagName('hot')

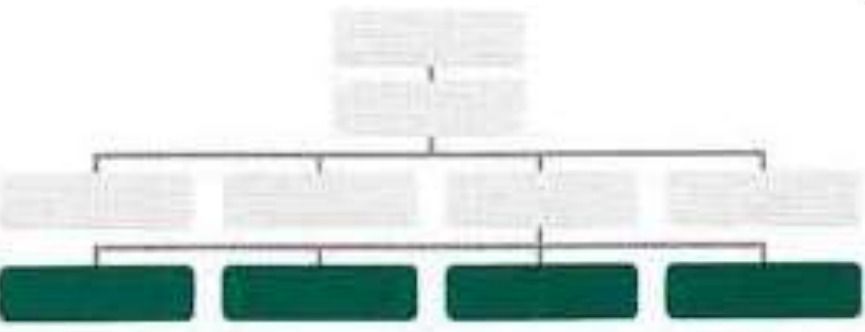
Επιλογή ενός ή περισσότερων κόμβων 2

getElementsByTagName('Tag')

getElementsByTagName("li")

var liAll = getElementsByTagName("li")

```
<html>
  <body>
    <div id="page">
      <h1 id="header">List</h1>
      <h2>Buy groceries</h2>
      <ul>
        <li id="one" class="hot"><em>fresh</em> figs</li>
        <li id="two" class="hot">pine nuts</li>
        <li id="three" class="hot">honey</li>
        <li id="four">balsamic vinegar</li>
      </ul>
      <script src="js/list.js"></script>
    </div>
  </body>
</html>
```



getElementsByTagName('li')

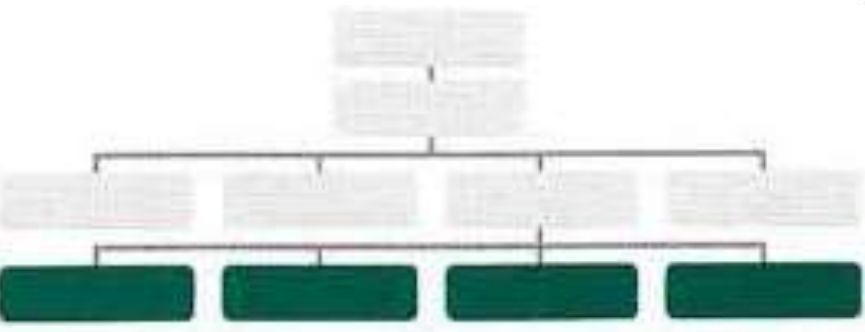
Επιλογή ενός ή περισσότερων κόμβων 3

`querySelectorAll('css selector')`

`querySelectorAll('li.hot')`

`var lihotall = getElementsByTagName("li.hot")`

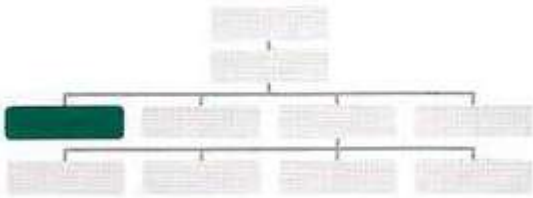
```
<html>
  <body>
    <div id="page">
      <h1 id="header">List</h1>
      <h2>Buy groceries</h2>
      <ul>
        <li id="one" class="hot"><em>fresh</em> figs</li>
        <li id="two" class="hot">pine nuts</li>
        <li id="three" class="hot">honey</li>
        <li id="four">balsamic vinegar</li>
      </ul>
      <script src="js/list.js"></script>
    </div>
  </body>
</html>
```



`getElementsByTagName('li')`

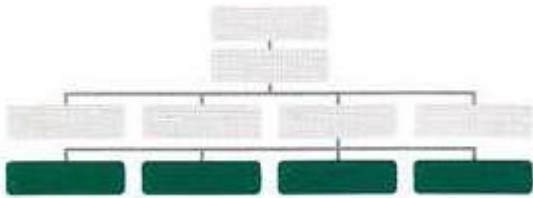
Επιλογή ενός ή περισσότερων κόμβων **NODELISTS**

Εφόσον μια μέθοδος DOM μπορεί να επιστρέψει περισσότερα από ένα στοιχεία, επιστρέφει μια λίστα (Nodelist)



`getElementsByTagName('h1')`

<u>INDEX</u>	<u>NUMBER & ELEMENT</u>
0	<h1>

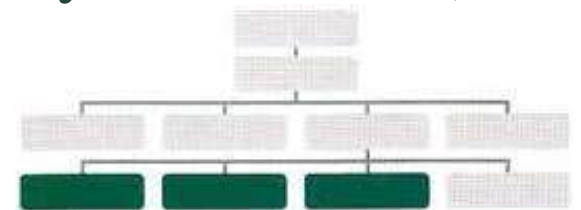


`getElementsByTagName('li')`

<u>INDEX</u>	<u>NUMBER & ELEMENT</u>
0	<li id="one" class="hot">
1	<li id="two" class="hot">
2	<li id="three" class="hot">
3	<li id="four">

<u>INDEX</u>	<u>NUMBER & ELEMENT</u>
0	<li id="one" class="hot">
1	<li id="two" class="hot">
2	<li id="three" class="hot">

`getElementsByClassName('hot')`



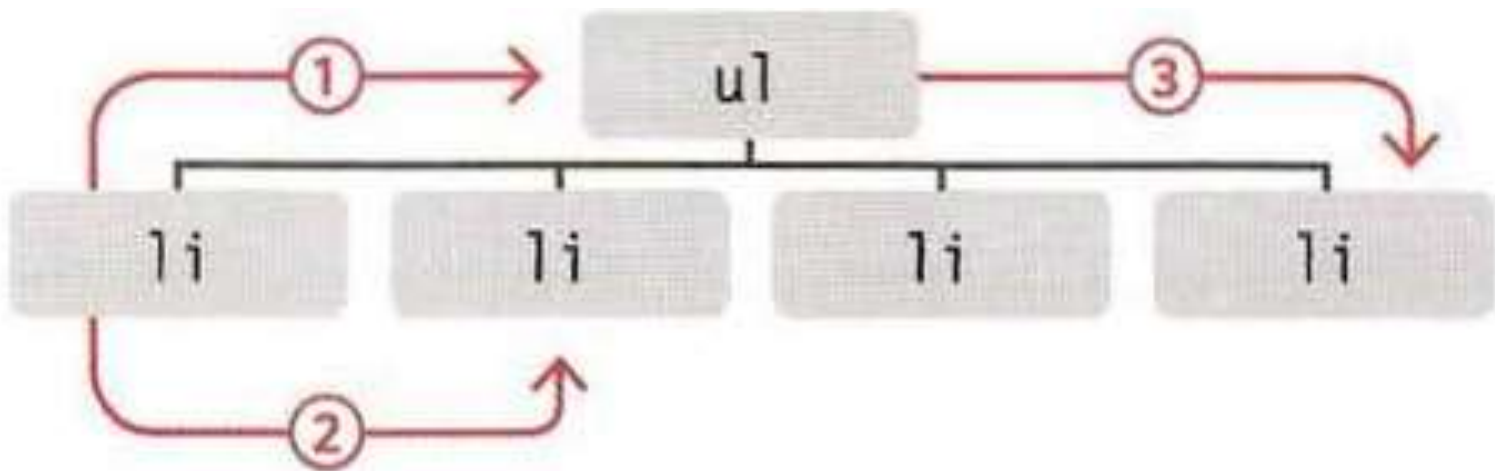
Μετακίνηση μεταξύ σχετικών κόμβων 1

Όταν έχετε έναν κόμβο στοιχείου, μπορείτε να επιλέξετε ένα άλλο στοιχείο σε σχέση με αυτό χρησιμοποιώντας αυτές τις πέντε ιδιότητες.

`parentNode` (1)

`previousSibling` / `nextSibling` (2)

`firstChild` / `lastChild` (3)

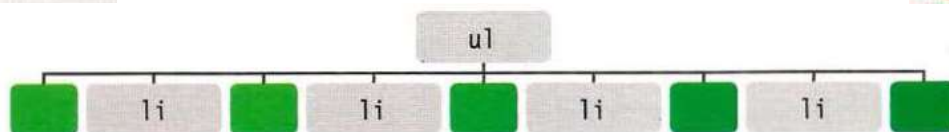
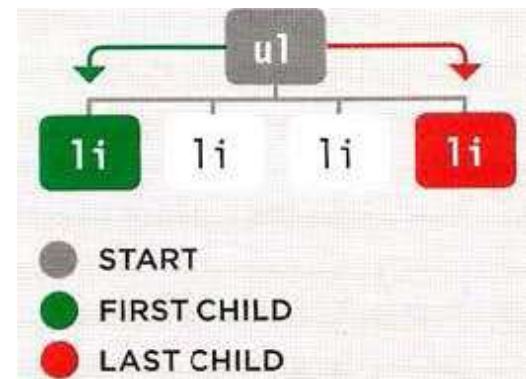
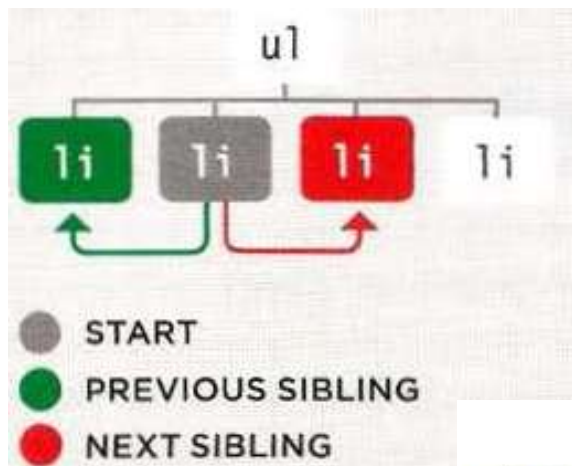


Μετακίνηση μεταξύ σχετικών κόμβων 1

Όταν έχετε έναν κόμβο στοιχείου, μπορείτε να επιλέξετε ένα άλλο στοιχείο σε σχέση με αυτό χρησιμοποιώντας αυτές τις πέντε ιδιότητες.

// Select the starting point and find its siblings

```
var startItem = document.getElementById('two');  
var prevItem = startItem.previousSibling;  
var nextItem = startItem.nextSibling;
```



Προσοχή στα Κενά μεταξύ στοιχείων

Παράδειγμα 1

```
<!DOCTYPE html>
<html lang="el-GR"> <head>
  <title>JavaScript DOM - Basic Page</title>
  <meta charset="utf-8">
  <link rel="stylesheet" href="css/c05b.css">
</head>
<body>
  <div id="page">
    <h1 id="header">Εργασίες για το σπίτι</h1>
    <h2>Λίστα Εργασιών</h2>
    <ul>
      <li id="one" class="hot"><em>Ομαδική</em> Έρευνα</li>
      <li id="two" class="hot">Ασκήσεις Μαθηματικών</li>
      <li id="three" class="hot">Ιστοσελίδα</li>
      <li id="four">Ηλεκτρονικό Κύκλωμα</li>
    </ul>
    <!-- <script src="js/lget-element-by-id.js"></script> -->
  </div>
</body> </html>
```



Εργασίες για το σπίτι

ΛΙΣΤΑ ΕΡΓΑΣΙΩΝ

Ομαδική Έρευνα

Ασκήσεις Μαθηματικών

Ιστοσελίδα

Ηλεκτρονικό Κύκλωμα

```
// Select the element and store it in a variable.
var el = document.getElementById('one');
```

```
// Change the value of the class attribute.
el.className = 'cool';
```

Παράδειγμα 2 (ClassName)

Εργασίες για το σπίτι

ΛΙΣΤΑ ΕΡΓΑΣΙΩΝ

Ομαδική Έρευνα

Ασκήσεις Μαθηματικών

Ιστοσελίδα

Ηλεκτρονικό Κύκλωμα

```
<body>
  <div id="page">
    <h1 id="header">Εργασίες για το σπίτι</h1>
    <h2>Λιστα Εργασιών</h2>
    <ul>
      <li id="one" class="hot"><em>Ομαδική</em> Έρευνα</li>
      <li id="two" class="hot">Ασκήσεις Μαθηματικών</li>
      <li id="three" class="hot">Ιστοσελίδα</li>
      <li id="four">Ηλεκτρονικό Κύκλωμα</li>
    </ul>
    <script src="js/2get-elements-by-class-name.js"></script>
  </div>
</body>
```

```
// Find hot items
```

```
var elements = document.getElementsByClassName('hot');
```

```
// If 3 or more are found
```

```
if (elements.length > 2) {
```

```
// Select the third one from the NodeList
```

```
  var el = elements[2];
```

```
// Change the value of its class attribute
```

```
  el.className = 'cool';
```

```
}
```

Εργασίες για το σπίτι

ΛΙΣΤΑ ΕΡΓΑΣΙΩΝ

Ομαδική Έρευνα

Ασκήσεις Μαθηματικών

Ιστοσελίδα

Ηλεκτρονικό Κύκλωμα

Παράδειγμα 3 (TagName)

```
<body>
  <div id="page">
    <h1 id="header">Εργασίες για το σπίτι</h1>
    <h2>Λίστα Εργασιών</h2>
    <ul>
      <li id="one" class="hot"><em>Ομαδική</em> Έρευνα</li>
      <li id="two" class="hot">Ασκήσεις Μαθηματικών</li>
      <li id="three" class="hot">Ιστοσελίδα</li>
      <li id="four">Ηλεκτρονικό Κύκλωμα</li>
    </ul>
    <script src="js/3get-elements-by-tag-name.js"></script>
  </div>
</body>
```



Εργασίες για το σπίτι
ΛΙΣΤΑ ΕΡΓΑΣΙΩΝ

Ομαδική Έρευνα

Ασκήσεις Μαθηματικών

Ιστοσελίδα

Ηλεκτρονικό Κύκλωμα

```
// Find <li> elements
var elements = document.getElementsByTagName('li');
// If 1 or more are found
if (elements.length > 0) {
  // Select the first one using array syntax
  var el = elements[0];
  // Change the value of the class attribute
  el.className = 'cool';
}
```

Παράδειγμα 4 (css Selector)



Εργασίες για το σπίτι

ΛΙΣΤΑ ΕΡΓΑΣΙΩΝ

Ομαδική Έρευνα

Ασκήσεις Μαθηματικών

Ιστοσελίδα

Ηλεκτρονικό Κύκλωμα

```
<body>
  <div id="page">
    <h1 id="header">Εργασίες για το σπίτι</h1>
    <h2>Λιστα Εργασιών</h2>
    <ul>
      <li id="one" class="hot"><em>Ομαδική</em> Έρευνα</li>
      <li id="two" class="hot">Ασκήσεις Μαθηματικών</li>
      <li id="three" class="hot">Ιστοσελίδα</li>
      <li id="four">Ηλεκτρονικό Κύκλωμα</li>
    </ul>
    <script src="js/4query-selector.js"></script>
  </div>
</body>
```

```
// querySelector only returns the first match.
var el = document.querySelector('li.hot');
el.className = 'cool';
// querySelectorAll returns a NodeList.
// The third li element is then selected and changed
var els = document.querySelectorAll('li.hot');
els[1].className = 'cool';
```



Εργασίες για το σπίτι

ΛΙΣΤΑ ΕΡΓΑΣΙΩΝ

Ομαδική Έρευνα

Ασκήσεις Μαθηματικών

Ιστοσελίδα

Ηλεκτρονικό Κύκλωμα

Παράδειγμα 5 (Nodelist Looping)

Εργασίες για το σπίτι

ΛΙΣΤΑ ΕΡΓΑΣΙΩΝ

Ομαδική Έρευνα

Ασκήσεις Μαθηματικών

Ιστοσελίδα

Ηλεκτρονικό Κύκλωμα

```
<body>
  <div id="page">
    <h1 id="header">Εργασίες για το σπίτι</h1>
    <h2>Λιστα Εργασιών</h2>
    <ul>
      <li id="one" class="hot"><em>Ομαδική</em> Έρευνα</li>
      <li id="two" class="hot">Ασκήσεις Μαθηματικών</li>
      <li id="three" class="hot">Ιστοσελίδα</li>
      <li id="four">Ηλεκτρονικό Κύκλωμα</li>
    </ul>
    <script src="js/5node-list.js"></script>
  </div>
</body>
```

```
// Store NodeList in array
var hotItems = document.querySelectorAll('li.hot');
// If it contains items
if (hotItems.length > 0) {
  // Loop through each item
  for (var i = 0; i < hotItems.length; i++) {
    // Change value of class attribute
    hotItems[i].className = 'cool';
  }
}
```

Εργασίες για το σπίτι

ΛΙΣΤΑ ΕΡΓΑΣΙΩΝ

Ομαδική Έρευνα

Ασκήσεις Μαθηματικών

Ιστοσελίδα

Ηλεκτρονικό Κύκλωμα

Παράδειγμα 6 (PREVIOUS & NEXT SIBLING)

```
<body>
  <div id="page">
    <h1 id="header">Εργασίες για το σπίτι</h1>
    <h2>Λιστα Εργασιών</h2>
    <ul><li id="one" class="hot"><em>Ομαδική</em> Έρευνα</li><li id="two" class="hot">Α<
      <script src="js/6sibling.js"></script>
    </div>
</body>
```

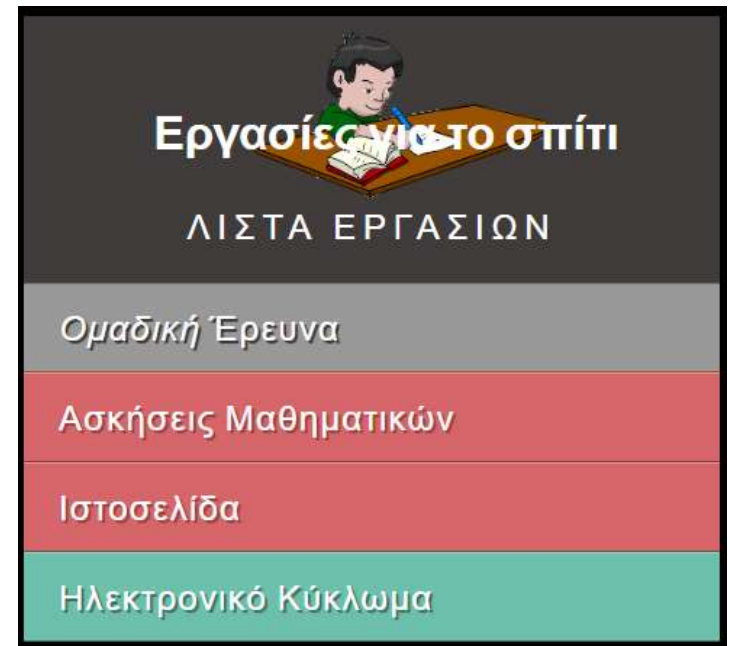


```
// Select the starting point and find its siblings.
var startItem = document.getElementById('two');
var prevItem = startItem.previousSibling;
var nextItem = startItem.nextSibling;

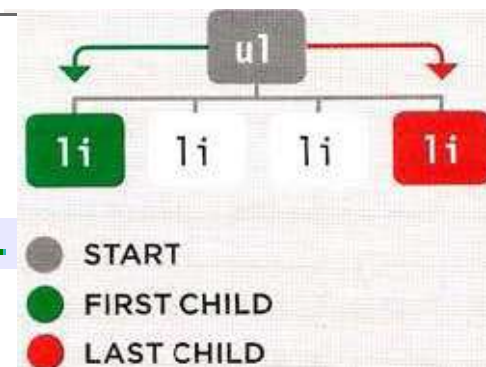
// Change the values of the siblings' class attributes.
prevItem.className = 'complete';
nextItem.className = 'cool';
```

Παράδειγμα 7 (FIRST & LAST CHILD)

```
<body>
  <div id="page">
    <h1 id="header">Εργασίες για το σπίτι</h1>
    <h2>Λιστα Εργασιών</h2>
    <ul><li id="one" class="hot"><em>Ομαδική</em> Έρευνα</li><li id="two"
    <script src="js/7child.js"></script>
  </div>
</body>
```

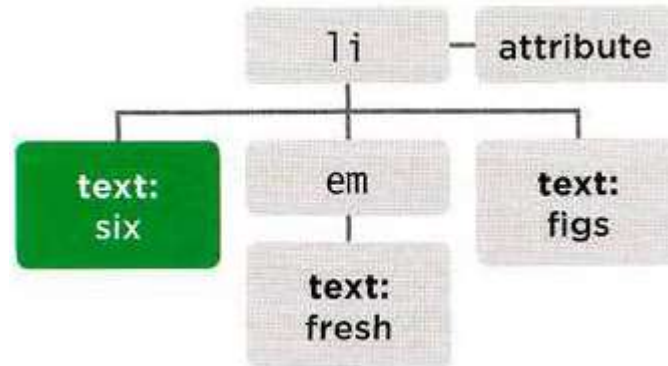


```
// Select the starting point and find its children.
var startItem = document.getElementsByTagName('ul')[0];
var firstItem = startItem.firstChild;
var lastItem = startItem.lastChild;
// Change the values of the children's class attributes.
firstItem.className = 'complete';
lastItem.className = 'cool';
```



Τροποποίηση στοιχείου κειμένου (NODEVALUE)

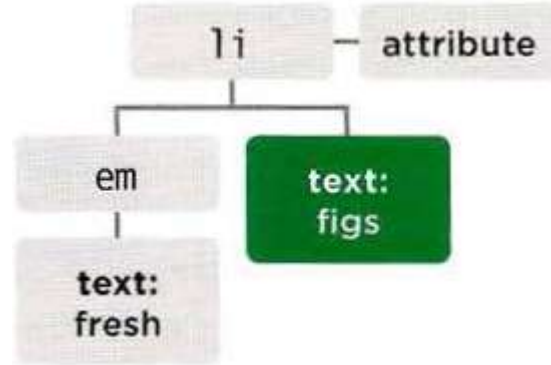
```
<li id="one">six <em>fresh</em> figs</li>
```



- Το πρώτο «παιδί» του στοιχείου `` είναι ένας κόμβος κειμένου, που περιέχει τη λέξη “six”.
- Έχει ένα αδελφό που είναι ένας κόμβος στοιχείου για το ``. Με τη σειρά του, έχει έναν «παιδί» κόμβο κειμένου που περιέχει τη λέξη “fresh”.
- Τέλος, υπάρχει ένας κόμβος κειμένου που κρατά τη λέξη “figs”, ο οποίος είναι «αδελφός» των άλλων δύο.

Τροποποίηση στοιχείου κειμένου (NODEVALUE)

```
<li id="one"><em>fresh</em> figs</li>
```



```
document.getElementById('one').firstChild.nextSibling.nodeValue;
```

1. Ο κόμβος στοιχείου `` επιλέγεται χρησιμοποιώντας τη μέθοδο `getElementById()`.
2. Το πρώτο παιδί `` είναι το στοιχείο ``.
3. Ο κόμβος κειμένου είναι ο επόμενος αδελφός αυτού του στοιχείου ``.
4. Έχετε τον κόμβο κειμένου και έχετε πρόσβαση στο περιεχόμενό του χρησιμοποιώντας την `nodeValue`

Παράδειγμα 8 Τροποποίηση στοιχείου κειμένου nodeValue

```
<body>
  <div id="page">
    <h1 id="header">Εργασίες για το σπίτι</h1>
    <h2>Λίστα Εργασιών</h2>
    <ul>
      <li id="one" class="hot"><em>Ομαδική</em> Έρευνα</li>
      <li id="two" class="hot">Ασκήσεις Μαθηματικών</li>
      <li id="three" class="hot">Ιστοσελίδα</li>
      <li id="four">Ηλεκτρονικό Κύκλωμα</li>
    </ul>
    <script src="js/8node-value.js"></script>
  </div>
</body>
```



Εργασίες για το σπίτι

ΛΙΣΤΑ ΕΡΓΑΣΙΩΝ

Ομαδική Έρευνα

Εργασία Μαθηματικών

Ιστοσελίδα

Ηλεκτρονικό Κύκλωμα

```
// Get second list item
var itemTwo = document.getElementById('two');
// Get its text content
var elText = itemTwo.firstChild.nodeValue;
// Change Ασκήσεις to Εργασία
elText = elText.replace('Ασκήσεις', 'Εργασία');
// Update the list item
itemTwo.firstChild.nodeValue = elText;
```

Παράδειγμα 6 Τροποποίηση στοιχείου κειμένου textContent & InnerText

```
<li id="one" ><em>Ομαδική</ em> Έρευνα</ li>
```

```
Var a = document.getElementById('one').innerText;  
a = “Έρευνα”
```

innerText: Συλλέγει και να ενημερώνει το κείμενο από κάποιο στοιχείο

```
Var a = document.getElementById('one').textContent;  
a = “Ομαδική Έρευνα”
```

textContent: Συλλέγει και να ενημερώνει το κείμενο από κάποιο στοιχείο αγνοώντας κάθε markup μέσα στο στοιχείο. Αντικαθιστά όλο το περιεχόμενό του στοιχείου, μαζί και οποιωνδήποτε markup

***** Διαφοροποίηση Browser**

Διαχείριση περιεχομένου & markup με **innerHTML**

Με την **innerHTML** μπορείτε να έχετε πρόσβαση και να τροποποιήσει το περιεχόμενο οποιουδήποτε στοιχείου, συμπεριλαμβανομένων τυχόν στοιχείων child. Επίσης μπορείτε να προσθέσετε ή να διαγράψετε περιεχόμενο.

```
<li id="one" ><em>Ομαδική</em> Έρευνα</li>
```

```
Var a=document.getElementById('one').innerHTML;  
a = "<em>Ομαδική </em>Έρευνα"
```

```
document.getElementById('one').innerHTML = a;
```

Προσοχή: Η χρήση της πιο πάνω ιδιότητας πρέπει να συνοδεύεται με έλεγχο στοιχείων καθώς δίνει δικαίωμα εισαγωγής ανεπιθύμητου περιεχομένου στη σελίδα μας

Παράδειγμα 10

Τροποποίηση κειμένου και Markup innerHTML

```
<body>
  <div id="page">
    <h1 id="header">Εργασίες για το σπίτι</h1>
    <h2>Λίστα Εργασιών</h2>
    <ul>
      <li id="one" class="hot"><em>Ομαδική</em> Έρευνα</li>
      <li id="two" class="hot">Ασκήσεις Μαθηματικών</li>
      <li id="three" class="hot">Ιστοσελίδα</li>
      <li id="four">Ηλεκτρονικό Κύκλωμα</li>
    </ul>
    <script src="js/10inner-html.js"></script>
  </div>
</body>
```



Ομαδική Έρευνα [Link](#)

Ασκήσεις Μαθηματικών

Ιστοσελίδα

Ηλεκτρονικό Κύκλωμα

```
// Store the first list item in a variable
var firstItem = document.getElementById('one');
// Get the content of the first list item
var itemContent = firstItem.innerHTML;
// Update the content of the first list item so it is a link
firstItem.innerHTML = '<a href=\ "http://www.pashias.net\" >'
+ itemContent + '</a>';
```

Προσθήκη στοιχείου στο DOM Tree

1

Δημιουργία του
στοιχείου κόμβου με
`createElement()`

Αρχικά δημιουργείται ένας νέος κόμβος στοιχείου με τη μέθοδο `createElement()` και αποθηκεύεται σε μια μεταβλητή.

2

Δημιουργία του
στοιχείου κειμένου
με `createTextNode()`

Μετά δημιουργείται ένα νέο στοιχείο κειμένου χρησιμοποιώντας το `createTextNode()`. Επίσης αποθηκεύεται σε μια μεταβλητή.

2

Προσθήκη των
στοιχείων στο DOM
με `appendChild()`

Τέλος κρεμάμε το στοιχείο κειμένου στο νέο στοιχείο και όλο μαζί στο κατάλληλο στοιχείο του DOM Tree ως `child`

Ακολουθεί παράδειγμα

Παράδειγμα 11 Προσθήκη στοιχείου στο DOM Tree

createElement() - createTextNode() - appendChild()

```
<body>
  <div id="page">
    <h1 id="header">Εργασίες για το σπίτι</h1>
    <h2>Λίστα Εργασιών</h2>
    <ul>
      <li id="one" class="hot"><em>Ομαδική</em> Έρευνα</li>
      <li id="two" class="hot">Ασκήσεις Μαθηματικών</li>
      <li id="three" class="hot">Ιστοσελίδα</li>
      <li id="four">Ηλεκτρονικό Κύκλωμα</li>
    </ul>
    <script src="js/lladd-element.js"></script>
  </div>
</body>
```



```
// Create a new element and store it in a variable.
var newEl = document.createElement('li');
// Create a text node and store it in a variable.
var newText = document.createTextNode('Φύλλο Εργασίας C++')
// Attach the new text node to the new element.
newEl.appendChild(newText);
// Find the position where the new element should be added.
var position = document.getElementsByTagName('ul')[0];
// Insert the new element into its position.
position.appendChild(newEl);
```

Διαγραφή στοιχείου από το DOM Tree

1

Αποθήκευση του στοιχείου κόμβου σε μια μεταβλητή.

Αρχικά επιλέγετε το στοιχείο που πρόκειται να αφαιρέσετε και το αποθηκεύετε σε μια μεταβλητή.

2

Αποθήκευση του γονικού κόμβου σε μια μεταβλητή

Στη συνέχεια, βρίσκετε το γονικό στοιχείο και να το αποθηκεύετε σε μια μεταβλητή. (χρησιμοποιήσετε την ιδιότητα **parentNode** αυτού του στοιχείου.

2

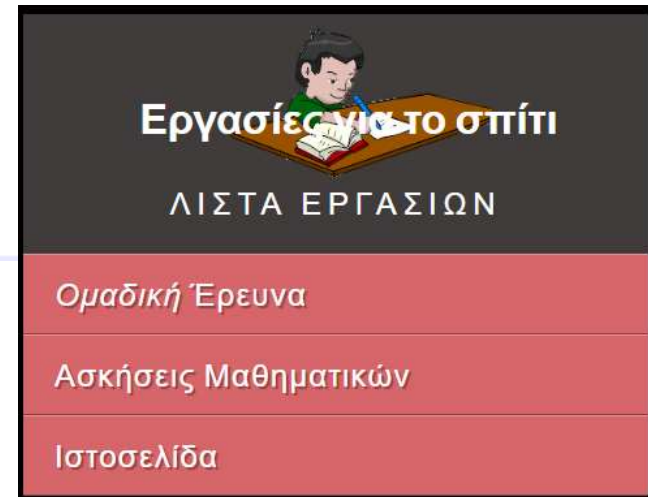
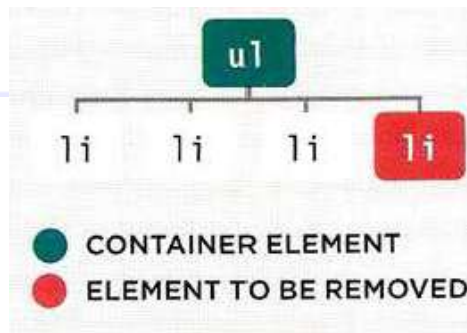
Διαγραφή στοιχείου από το γονικό στοιχείο με **appendChild()**

Χρησιμοποιείτε τη μέθοδο `removeChild()` στο στοιχείο γόνος για να διαγράψετε το στοιχείο που δεν θέλετε πλέον.

Ακολουθεί παράδειγμα

Παράδειγμα 12 Διαγραφή στοιχείου από το DOM

```
<body>
  <div id="page">
    <h1 id="header">Εργασίες για το σπίτι</h1>
    <h2>Λιστα Εργασιών</h2>
    <ul>
      <li id="one" class="hot"><em>Ομαδική</em> Έρευνα</li>
      <li id="two" class="hot">Ασκήσεις Μαθηματικών</li>
      <li id="three" class="hot">Ιστοσελίδα</li>
      <li id="four">Ηλεκτρονικό Κύκλωμα</li>
    </ul>
    <script src="js/12remove-element.js"></script>
  </div>
</body>
```



```
// Store the element to be removed in a variable.
var removeEl = document.getElementsByTagName('li')[3];
// Find the element which contains the element to be removed.
var containerEl = document.getElementsByTagName('ul')[0];
// Remove the element.
containerEl.removeChild(removeEl);
```

innerHTML vs Μεθόδοι DOM

Ιδιότητα innerHTML

❑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ

- Μπορείτε να τη χρησιμοποιήσετε για να προσθέσετε πολλά νέα στοιχεία χρησιμοποιώντας λιγότερο κώδικα από τις μεθόδους χειρισμού DOM.
- Μπορεί να είναι ταχύτερη από τις μεθόδους DOM όταν προσθέτουμε πολλά νέα στοιχεία σε μια ιστοσελίδα.
- Είναι ένας απλός τρόπος για να αφαιρέσετε όλο το περιεχόμενο από ένα στοιχείο (εκχωρώντας την κενή συμβολοσειρά).

❑ ΜΕΙΟΝΕΚΤΗΜΑΤΑ

- Δεν πρέπει να χρησιμοποιείται για την προσθήκη περιεχομένου που έχει προέρχονται από ένα χρήστη, καθώς μπορεί να αποτελέσει σημαντικό ρήγμα ασφάλειας.
- Μπορεί να είναι δύσκολο να απομονώσετε μεμονωμένα στοιχεία που θέλετε να ενημερώσετε σε ένα μεγαλύτερο DOM Tree.
- Οι χειριστές συμβάντων ενδέχεται να μην λειτουργούν πλέον όπως είχαν αρχικά σχεδιαστεί.

innerHTML vs Μεθόδοι DOM

Μέθοδοι DOM

❑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ

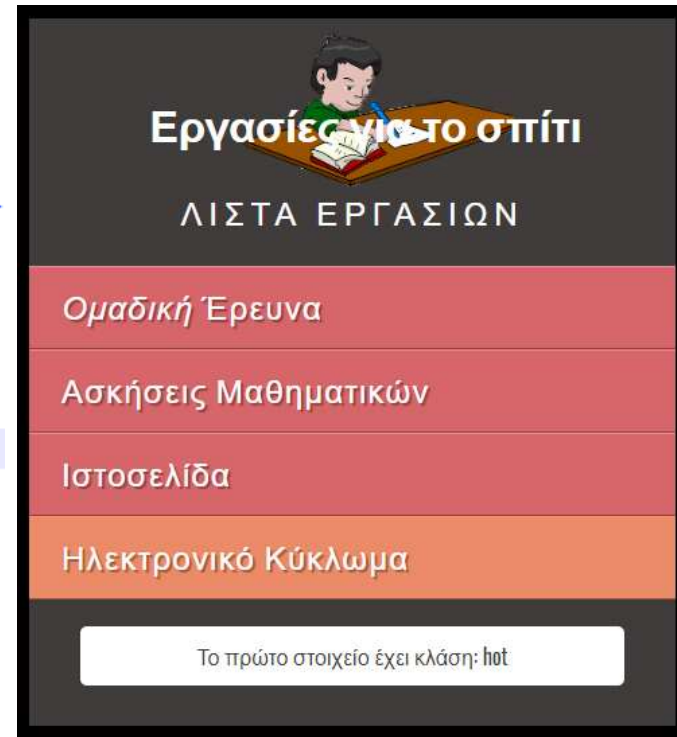
- Είναι κατάλληλες για την αλλαγή ενός στοιχείου από ένα DOM όπου υπάρχουν πολλά αδέλφια.
- Δεν επηρεάζει τους χειριστές συμβάντων.
- Εύκολα επιτρέπει σε μια δέσμη ενεργειών να προσθέσει στοιχεία (όταν δεν θέλετε να αλλάξετε πολλά ταυτόχρονα).

❑ ΜΕΙΟΝΕΚΤΗΜΑΤΑ

- Εάν πρέπει να κάνετε πολλές αλλαγές στο περιεχόμενο μιας σελίδας, είναι πιο αργή από την innerHTML.
- Πρέπει να γράψετε περισσότερο κώδικα για να επιτύχετε το ίδιο σε σύγκριση με την innerHTML.

Παράδειγμα 13 Εντοπισμός ιδιοτήτων

```
<body>
  <div id="page">
    <h1 id="header">Εργασίες για το σπίτι</h1>
    <h2>Λίστα Εργασιών</h2>
    <ul>
      <li id="one" class="hot"><em>Ομαδική</em> Έρευνα</li>
      <li id="two" class="hot">Ασκήσεις Μαθηματικών</li>
      <li id="three" class="hot">Ιστοσελίδα</li>
      <li id="four">Ηλεκτρονικό Κύκλωμα</li>
    </ul>
    <div id="scriptResults">
      <p>The first item has a class name: hot</p>
    </div>
  </div>
  <script src="js/13get-attribute.js"></script>
</body>
```



Εργασίες για το σπίτι

ΛΙΣΤΑ ΕΡΓΑΣΙΩΝ

- Ομαδική Έρευνα
- Ασκήσεις Μαθηματικών
- Ιστοσελίδα
- Ηλεκτρονικό Κύκλωμα

Το πρώτο στοιχείο έχει κλάση: hot

```
var firstItem = document.getElementById('one'); // Get first list item
if (firstItem.hasAttribute('class')) {        // If it has class attribute
  var attr = firstItem.getAttribute('class'); // Get the attribute

  // Add the value of the attribute after the list
  var el = document.getElementById('scriptResults');
  el.innerHTML = '<p>Το πρώτο στοιχείο έχει κλάση: ' + attr + '</p>';
}
```

Παράδειγμα 14 Δημιουργία & τροποποίηση ιδιοτήτων

```
<body>
  <div id="page">
    <h1 id="header">Εργασίες για το σπίτι</h1>
    <h2>Λίστα Εργασιών</h2>
    <ul>
      <li id="one" class="hot"><em>Ομαδική</em> Έρευνα</li>
      <li id="two" class="hot">Ασκήσεις Μαθηματικών</li>
      <li id="three" class="hot">Ιστοσελίδα</li>
      <li id="four">Ηλεκτρονικό Κύκλωμα</li>
    </ul>
  </div>
  <script src="js/14set-attribute.js"></script>
</body>
```



```
// Get the first item
var firstItem = document.getElementById('one');
// Change its class attribute
firstItem.className = 'complete';
// Get fourth item
var fourthItem = document.getElementsByTagName('li').item(3);
// NOTE: The following line should say fourthItem (not el2)
// Add an attribute to it
fourthItem.setAttribute('class', 'cool');
```

Παράδειγμα 15 Ακύρωση ιδιοτήτων

```
<body>
  <div id="page">
    <h1 id="header">Εργασίες για το σπίτι</h1>
    <h2>Λίστα Εργασιών</h2>
    <ul>
      <li id="one" class="hot"><em>Ομαδική</em> Έρευνα</li>
      <li id="two" class="hot">Ασκήσεις Μαθηματικών</li>
      <li id="three" class="hot">Ιστοσελίδα</li>
      <li id="four">Ηλεκτρονικό Κύκλωμα</li>
    </ul>
  </div>
  <script src="js/15remove-attribute.js"></script>
</body>
```



```
// Get the first item
var firstItem = document.getElementById('one');
// If it has a class attribute
if (firstItem.hasAttribute('class')) {
  // Remove its class attribute
  firstItem.removeAttribute('class');
}
```

Άσκηση 1

```
// ADDING ITEMS TO START AND END OF LIST
var list = document.getElementsByTagName('ul')[0]; // Get the <ul> element

// ADD NEW ITEM TO END OF LIST
var newItemLast = document.createElement('li'); // Create element
var newTextLast = document.createTextNode('Διαγώνισμα C++ !!!'); // Create text node
newItemLast.appendChild(newTextLast); // Add text node to element
list.appendChild(newItemLast); // Add element end of list

// ADD NEW ITEM START OF LIST
var newItemFirst = document.createElement('li'); // Create element
var newTextFirst = document.createTextNode('Διαγώνισμα Ιστοσελίδα !!!'); // Create text node
newItemFirst.appendChild(newTextFirst); // Add text node to element
list.insertBefore(newItemFirst, list.firstChild); // Add element to list

var listItems = document.querySelectorAll('li'); // All <li> elements

// ADD A CLASS OF COOL TO ALL LIST ITEMS
var i; // Counter variable
for (i = 0; i < listItems.length; i++) { // Loop through elements
    listItems[i].className = 'cool'; // Change class to cool
}

// ADD NUMBER OF ITEMS IN THE LIST TO THE HEADING
var heading = document.querySelector('h2'); // h2 element
var headingText = heading.firstChild.nodeValue; // h2 text
var totalItems = listItems.length; // No. of <li> elements
var newHeading = headingText + '<span>' + totalItems + '</span>'; // Content
// Update h2 using innerHTML (not textContent) because it contains markup
heading.innerHTML = newHeading;
```

Άσκηση 2

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 </head>
6 <body>
7 <h1 id="et1"> et1: This is a demo about JavaScript </h1>
8 <h1 id="et2"> et2: getElementById command</h1>
9 <h1 id="part1"> part1: You are wellcome</h1>
10
11 <script>
12 var s1=document.getElementById("et1").innerHTML;
13 document.write(s1+"<br/>");
14 var s2=document.getElementById("et2").innerHTML;
15 document.write(s2+"<br/>");
16 var s3=document.getElementById("part1").innerHTML;
17 document.write(s3+"<br/>");
18 var s4=document.getElementById("part1").innerHTML=(s3 + " Add New message");
19 document.write(s4+"<br/>");
20
21
22 </script>
23 <button type="button" onclick="document.getElementById('part1').innerHTML=
24     'And now we have new message!'">New!</button>
25 </body>
26 </html>
```

et1: This is a demo about JavaScript

et2: getElementById command

part1: You are wellcome Add New message

et1: This is a demo about JavaScript

et2: getElementById command

part1: You are wellcome

part1: You are wellcome Add New message

New!