

## The Document Object Model

Αρχικά, ας ανακεφαλαιώσουμε όσα γνωρίζουμε για την HTML, το CSS και τη JavaScript.

Όταν θέλετε να δημιουργήσετε μια εφαρμογή Ιστού, δημιουργείτε ένα έγγραφο HTML γεμάτο με HTML, CSS και JavaScript.

Αυτές οι τρεις τεχνολογίες χρησιμοποιούνται μαζί σαν μια ομάδα, η καθεμία παίζει διαφορετικό ρόλο στη δημιουργία αυτού που βλέπετε στο πρόγραμμα περιήγησής σας.

Η HTML είναι μια markup language που χρησιμοποιείται για τον καθορισμό της δομής και του περιεχομένου της σελίδας. Οι ετικέτες που χρησιμοποιούμε όπως <head>, <body> και <button> έχουν σημασία και το πρόγραμμα περιήγησης θα επεξεργαστεί αυτές τις ετικέτες, αποδίδοντας το περιεχόμενο του εγγράφου στην οθόνη

Παράδειγμα

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>My First Heading</h1>
```

```
<p>My first paragraph.</p>
```

</body>

</html>

## Μνημονικοί Κανόνες

- Όλα τα HTML πρέπει να ξεκινούν με μια δήλωση τύπου εγγράφου:  
<!DOCTYPE html>
- Το ίδιο το έγγραφο HTML αρχίζει με <html> και τελειώνει με </html>.
- Το ορατό τμήμα του εγγράφου HTML βρίσκεται μεταξύ <body> και </body>.
- Μεταξύ των <body> και </body> βάζω διάφορες tags για να το κάνω πιο ευπαρουσίαστο, πχ παράγραφο (<p>κείμενο που θέλεις.</p>), Επικεφαλίδες (<h1> ...</h1>, <h2> ...</h2> κλπ ).
- Σύνδεσμους HTML. Όταν πατάς σε ένα σύνδεσμο μετακινείσαι από μία σελίδα σε μία άλλη. Σύνταξη :  
<a href="https://www.disneylandparis.com/en-int/">Click to visit the Disneeyland</a>
- εικόνες. Για να βάλεις μία εικόνα στη σελίδα σου χρειάζεσαι: α) tags και τη διεύθυνση της εικόνας. Για παράδειγμα  
  
εμφανίζει μια εικόνα Sleeping Beauty Castle.

Κοκ

Αλλά η HTML από μόνη της παράγει μια σελίδα με πολύ βαρετή και γενική εμφάνιση. Το CSS δημιουργήθηκε για να μας επιτρέψει να αλλάξουμε τον τρόπο με τον οποίο αναπαρίστανται οι ετικέτες HTML στο πρόγραμμα περιήγησης.

Χρησιμοποιώντας CSS, μπορείτε να δημιουργήσετε ένα κουμπί που εμφανίζεται σε χρώμα της επιλογής σας ή ένα πλαίσιο εισαγωγής κειμένου με στρογγυλεμένο περίγραμμα, καθιστώντας τα πιο ελκυστικά και μοντέρνα.

Η HTML και η CSS είναι εξαιρετικοί συνεργάτες, αλλά ακόμα και τότε, μια ιστοσελίδα εξακολουθεί να είναι απλώς ένα παθητικό πράγμα. Μόλις το πρόγραμμα περιήγησης επεξεργαστεί το περιεχόμενο και το αποδώσει, δεν μπορείτε να επεξεργαστείτε τη σελίδα με κανέναν τρόπο.

Δεν μπορείτε να αλλάξετε το φόντο και το χρώμα του κειμένου της σελίδας (όπως η μετάβαση σε σκοτεινή λειτουργία, για παράδειγμα). Δεν μπορείτε να δημιουργήσετε εντυπωσιακά κινούμενα σχέδια, να ενεργοποιήσετε το μικρόφωνο ή την κάμερα web ή να ανακτήσετε δεδομένα από άλλο ιστότοπο για εμφάνιση στη σελίδα μετά από ένα κλικ.

Η JavaScript είναι η γλώσσα προγραμματισμού που υποστηρίζεται από το πρόγραμμα περιήγησης και καλύπτει αυτό το κενό που μένει ανοιχτό από το HTML και το CSS.

Εν ολίγοις, η JavaScript δίνει ζωή σε στατικούς ιστότοπους, μετατρέποντάς τους δυναμικούς. Επιτρέπει σε έναν ιστότοπο να αλληλεπιδρά με τον χρήστη, να ανταποκρίνεται στις ενέργειές του και να ενημερώνει δυναμικά το περιεχόμενο χωρίς να απαιτείται επαναφόρτωση της σελίδας.

## ΠΟΥ ΕΙΝΑΙ ΤΟ DOM?

Εδώ είναι.

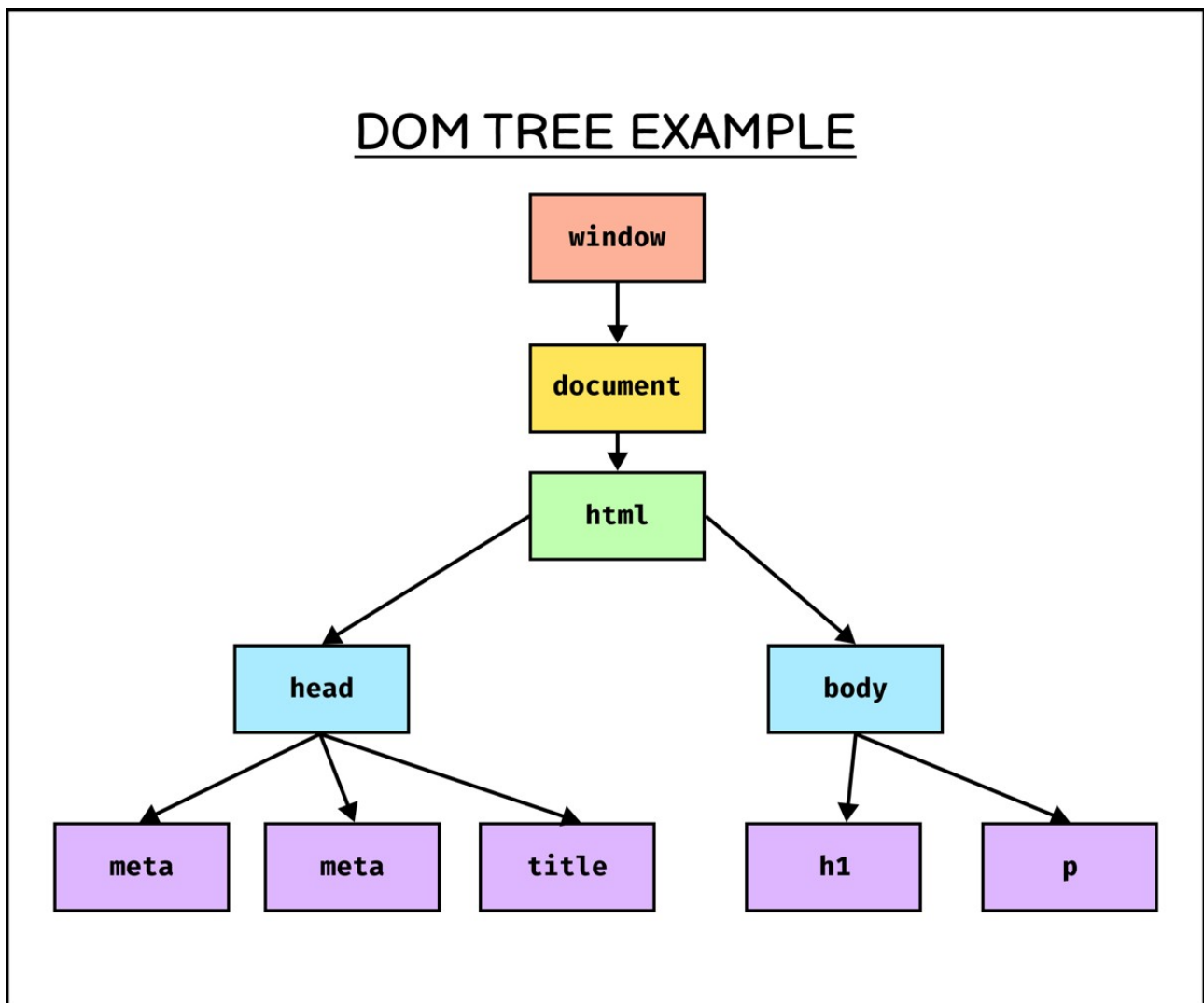
Το DOM είναι μια ιεραρχική δομή που αποτελείται από αντικείμενα που συνθέτουν μια ιστοσελίδα. Στη συνέχεια, τα προγράμματα περιήγησης Ιστού εκθέτουν αυτό το DOM, ώστε να μπορείτε να αλλάξετε τη δομή, το στυλ και το περιεχόμενο της σελίδας χρησιμοποιώντας JavaScript.

Για παράδειγμα, ας υποθέσουμε ότι έχετε το ακόλουθο περιεχόμενο HTML στο έγγραφό σας:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1>Hello World!</h1>
```

```
<p>This is a paragraph element</p>  
</body>  
</html>
```

Το γράφημα DOM που δημιουργήθηκε από το παραπάνω έγγραφο HTML έχει ως εξής



Στο πιο βασικό επίπεδο, ένας ιστότοπος αποτελείται από ένα έγγραφο HTML. Το πρόγραμμα περιήγησης που χρησιμοποιείτε για την προβολή του ιστότοπου είναι ένα πρόγραμμα που ερμηνεύει HTML και CSS και

αποδίδει το στυλ, το περιεχόμενο και τη δομή στη σελίδα που βλέπετε.

Εκτός από την ανάλυση του στυλ και της δομής του HTML και του CSS, το πρόγραμμα περιήγησης δημιουργεί μια αναπαράσταση του εγγράφου που είναι γνωστή ως Μοντέλο Αντικειμένου Εγγράφου. Αυτό το μοντέλο επιτρέπει στην JavaScript να έχει πρόσβαση στο περιεχόμενο κειμένου και στα στοιχεία του εγγράφου του ιστότοπου ως αντικείμενα.

Η JavaScript είναι μια διαδραστική γλώσσα και είναι ευκολότερο να κατανοήσουμε νέες έννοιες κάνοντας πράξη. Ας δημιουργήσουμε έναν πολύ βασικό ιστότοπο. Δημιουργήστε ένα αρχείο index.html και αποθηκεύστε το σε έναν νέο κατάλογο έργο

Πως ο browser μετατρέπει ένα html σε dom :

## HTML

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Example DOM with Multiple Divs</title>
```

```
</head>
```

```
<body>
```

```
<div id="header">
  <h1>Website Header</h1>
  <div class="nav">
    <a href="#home">Home</a>
    <a href="#about">About</a>
    <a href="#contact">Contact</a>
  </div>
</div>
```

```
<div id="main-content">
  <div class="content-section">
    <h2>Section 1</h2>
    <p>This is some content for section 1.</p>
  </div>
```

```
<div class="content-section">
  <h2>Section 2</h2>
  <p>This is some content for section 2.</p>
</div>
```

```
<div class="content-section">
  <h2>Section 3</h2>
  <p>This is some content for section 3.</p>
</div>
</div>
```

```
<div id="footer">
```

```

<p>Website Footer</p>
<div class="social-links">
  <a href="#facebook">Facebook</a>
  <a href="#twitter">Twitter</a>
  <a href="#instagram">Instagram</a>
</div>
</div>
</body>
</html>

```

## DOM

### html

```

└── head
    ├── meta (charset="UTF-8")
    ├── meta (name="viewport", content="width=device-width, initial-
scale=1.0")
    └── title
        └── "Example DOM with Multiple Divs"
└── body
    ├── div (id="header")
    │   ├── h1
    │   │   └── "Website Header"
    │   └── div (class="nav")
    │       ├── a (href="#home")
    │       │   └── "Home"
    │       └── a (href="#about")

```



```

|       |_____ "About"
|
|       |_____ a (href="#contact")
|           |_____ "Contact"
|_____ div (id="main-content")
|   |_____ div (class="content-section")
|   |   |_____ h2
|   |   |   |_____ "Section 1"
|   |   |   |_____ p
|   |   |       |_____ "This is some content for section 1."
|   |_____ div (class="content-section")
|   |   |_____ h2
|   |   |   |_____ "Section 2"
|   |   |   |_____ p
|   |   |       |_____ "This is some content for section 2."
|   |_____ div (class="content-section")
|   |   |_____ h2
|   |   |   |_____ "Section 3"
|   |   |   |_____ p
|   |       |_____ "This is some content for section 3."
|_____ div (id="footer")
|   |_____ p
|   |   |_____ "Website Footer"
|   |_____ div (class="social-links")
|   |   |_____ a (href="#facebook")
|   |   |   |_____ "Facebook"
|   |   |_____ a (href="#twitter")
|   |   |   |_____ "Twitter"
```

```
_____ a (href="#instagram")  
      _____ "Instagram"
```

Γιά να μετατρέψει το html σε dom ο browser θεωρεί ΚΑΘΕ tag ένα object(περίπου όπως τα objects της Java) και ΚΑΘΕ ιδιότητα (attribute) του object είναι περίπου ότι είναι μια μεταβλητή μιας class Java.

Δες για παράδειγμα την έκφραση

```
h1 {  
    color: green;  
}
```

Αυτό το css λέει στο browser ότι θα βλέπει την Επικεφαλίδα h1 θα χρησιμοποιεί πράσινο χρώμα για το κείμενο της κάθε Επικεφαλίδας. Με αυτό το τρόπο μπορώ να δώσω διαφορετική εμφάνιση σε στοιχεία(tags) του.html.

Αν έκανα κάτι αντίστοιχο στη Java θα έγραφα

```
public class h1{  
    String color = "green";  
}
```

Μπορώ να ορίσω τα πάντα για κάθε tag: διαστάσεις,χρώμα,background κλπ.

Όταν γράφω,πχ h1 {  
 color: green;

}

ο browser θα χρησιμοποιεί το πράσινοχρώμα για το κείμενο ΚΑΘΕ tag h1.

Μπορεί όμως κάποιο h1 tag στο html μου να μη το θέλω πράσινο αλλά πχ κόκκινο.

Τότε οφείλω να το δηλώσω το συγκεκριμένο ως εξής :

```
<h1 style="color: red;">I am red</h1>
```

---

## Tables δηλαδή πίνακες

Είναι δεδομένα οργανωμένα σε στήλες και γραμμές.

Παράδειγμα

Όνομα	Διεύθυνση	Τηλέφωνο
Μαρία Καρά	Αθήνα	12345678

Γιά να φτιάξεις ένα πίνακα το κάνεις σε 3 βήματα:

1. Χρειάζεσαι ένα tag με όνομα table : <table> .....</table>

2. Τώρα πρέπει να αρχίσεις να προσθέτεις γραμμές

```
<tr> ....</tr>
```

```
<tr> ... </tr>
```

3. Και τέλος μέσα σε κάθε <tr> ...</tr> (<tr> σημαίνει table row) πρέπει να βάλεις τα cells, τα “κουτιά δηλαδή” που αποτελούν κάθε σειρά.

```
<td>Column 1</td>
```

```
<td>Column 2</td>
```

```
<td>Column 3</td>
```

```
.
```

```
.
```

```
<td>Column 1</td>
```

```
<td>Column 2</td>
```

```
<td>Column 3</td>
```

Ολος ο κώδικας για να φτιάξεις ένα table με 2 γραμμές όπου κάθε μία θα έχει 3 cells :

```
<table>
```

```
<tr>
```

```
<td>Column 1</td>
```

```
<td>Column 2</td>
```

```
<td>Column 3</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Column 4</td>
```

```
<td>Column 5</td>
```

```
<td>Column 6</td>
```

```
</tr>
```

```
</table>
```

Μέσα στα `<td>....</td>` βάζω ότι κείμενο θέλω να εμφανίζεται στο αντιστοιχο cell. Εδώ για παράδειγμα το `table` θα εμφανίζει το κείμενο Column 1 στο πρώτο κουτί της 1ης γραμμής, το Column 2 στο δεύτερο κουτί της 1ης γραμμής κ.ο.κ.

Να πως θα είναι το `table` που έφτιαξα αν δω το `html` με ένα browser :

Column 1	Column 2	Column 3
Column 4	Column 5	Column 6

Μπορώ να προσθέσω/αφαιρέσω γραμμές, κελιά ,να βάλω borders κ.ο.κ.  
Μπορώ επίσης με `css` να ομορφύνω την εμφάνισή του η/και να ορίσω το ύψος ,το πλάτος,το περίγραμμα,το χρώμα κλπ.

Παράδειγμα

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <style>
```

```
  body {
```

```
    text-align: left;
```

```
  }
```

```
h1 {  
    color: green;  
}
```

```
table,
```

```
th,
```

```
td {
```

```
    /* Styling the border. */
```

```
    border: 1.5px solid blue;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
    <h1>Tables are fun</h1>
```

```
    <h2>Add border to table:</h2>
```

```
    <table>
```

```
        <tr>
```

```
            <th>Roll No.</th>
```

```
            <th>Name</th>
```

```
        </tr>
```

```
        <tr>
```

```
            <td>1</td>
```

```
            <td>A_B_C</td>
```

```
        </tr>
```

```
<tr>
  <td>2</td>
  <td>X_Y_Z</td>
</tr>
</table>
</body>

</html>
```

Αν το τρέξεις θα δεις

**Tables are fun**

Blue border

Roll No.	Name
1	text 1
2	text 2

δηλαδή ένα table με 2 σειρές και 2 στήλες, με πλάτος 75% της οθόνης σου και μπλέ περίγραμμα (border).

---

Forms (φόρμες)

Σίγουρα έχεις χρησιμοποιήσει φόρμες πολλές φορές.

Γιά παράδειγμα μπαίνεις σε ένα site και σου ζητάει να κάνεις login.

Μιά τέτοια σελίδα μοιάζει με τη παρακάτω

# Login

First name:

Last name:

Μιά φόρμα είναι ένα σύνολο από html tags οργανωμένα όλα μαζί σε ένα tag `<form>....</form>`.

Συνήθως μιά φόρμα περιέχει οπτικοποιημένους τρόπους για να μπορεί ο χρήστης να στείλει τα στοιχεία που του ζητούνται.

Τέτοι τρόποι-tags- είναι

`<input >` έ να κουτί που γράφουμεότι μας ζητάει

`<label>` → το όνομα του κουτιού που βλέπουμε

`<input type="submit">` → μια ειδική περίπτωση κουμπιού που στέλνει τα δεδομένα που συμπληρώσαμε μόλις πατηθεί.

Δεστε το παρακάτω κώδικα

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```



```
<style>
body {
    text-align: left;
}
h1 {
    color: green;
}
h2 {
    color: blue;;
}

p{
font-family:Arial;
font-size: 14px;
font-style: italic;
color: black;
}

form {
width: 30%;
box-sizing: border-box;
padding: 2rem;
border-radius: 1rem;
border: 1px solid black;
}

</style>
</head>
```

```
<body>
  <h1>Login </h1>
  <form action="action_page.php">
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value=""><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value=""><br><br>
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

και πάμε να εξηγήσουμε τη φόρμα λέξη προς λέξη :

1. <form action="action\_page.php"> λέει στο browser ότι μόλις πατηθεί το κουμπί submit να στείλει όλα τα στοιχεία της φόρμας που συμπληρώσαμε σε ένα αρχείο php που προφανως θα πάρει τα δεδομένα και θα κάνει κάποιο υπολογισμό.

2. <label for="fname">First name:</label><br> λέει στο browser να εμφανίσει το κείμενο First name: πάνω από το κουτί με attributes id="fname" name="fname".

3. <input type="submit" value="Submit"> στείλε τα στοιχεία.

ΕΞΗΓΗΣΕΙΣ.

Όταν ο browser στέλνει τα στοιχεία μιάς φόρμας, παίρνει την τιμή που δώσαμε σε κάθε <input> και σχηματίζει μιά ολόκληρη πρόταση με τα στοιχεία αυτά.

Όπως βλέπεις όμως υπάρχουν περισσότερα του ενός <input> άρα θα πρέπει να υπάρχει ένας τρόπος να ξέρουμε ποιο είναι ποιο.

Αυτό ακριβώς κάνουν οι attributes id και name.

Δίνουμε σε κάθε <input> ένα αναγνωριστικό, το id που χρησιμοποιείται κατά κύριο λόγο από javascript κώδικα, και ένα όνομα, το name που χρησιμοποιείται για να ξέρει αυτός που θα παραλάβει τα στοιχεία (εδώ τπ.php) ποιο είναι ποιο.

Σε μιά εξαπλουστευμένη μορφή, αν στη πιο πάνω φόρμα γράψουμε Julia στο κουτί με ταμπέλα First name και Roberts στο κουτί με ταμπέλα Last name, ο browser θα φτιάξει μια πρόταση του στυλ “action\_page.php?name=Julia&lname=Roberts” έτσι ώστε ο παραλείπτης-το action\_page.php-θα την πάρει, θα την σπάσει σε κομμάτια (parsing), θα βρει τις μεταβλητές name και lname, θα πάρει τη τιμή της κάθε μίας και θα προχωρήσει σε ότι υπολογισμό .