



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



# Εκπαιδευτικό Υλικό της HTML

## Υποκεφάλαιο 3 – Λειτουργίες στην HTML 5

WP3: Εκπαιδευτικό Υλικό του έργου Code4SP

Συντάχθηκε από: 



CITIZENS  
IN POWER



Center for Social  
Innovation



**ZAUG**  
gGmbH



social  
hackers  
academy



Escola Profissional de Espinho

## Υποκεφάλαιο 3: Λειτουργίες στην HTML5





## Τα Χαρακτηριστικά της HTML5

Σε αυτό το υποκεφάλαιο, θα επεξηγηθούν οι λειτουργίες της πέμπτης (και τελευταίας) μεγάλης έκδοσης της HTML.



## Νέοι τύποι εισαγωγής της HTML5

- Η HTML5 εισάγει πολλούς νέους τύπους `< input >` όπως ηλεκτρονικό ταχυδρομείο, ημερομηνία, ώρα, χρώμα, εύρος και ούτω καθεξής, με στόχο τη βελτίωση της εμπειρίας του χρήστη και τη βελτίωση της διαδραστικότητας των φορμών.
- Ωστόσο, εάν ένα πρόγραμμα περιήγησης αδυνατεί να αναγνωρίσει αυτούς τους νέους τύπους εισαγωγής, θα τους θεωρήσει κανονικό πλαίσιο κειμένου.



# Νέοι τύποι εισαγωγής της HTML5

Ακολουθούν ορισμένοι νέοι τύποι εισαγωγής:


- χρώμα
- ημερομηνία
- ημερομηνία-ώρα-τοπική
- Ηλεκτρονικό ταχυδρομείο
- μήνας
- αριθμός
- εύρος
- αναζήτηση
- τηλέφωνο
- ώρα
- Διεύθυνση url
- εβδομάδα

# Νέοι τύποι εισαγωγής της HTML5

## Η εισαγωγή χρώματος (color input)

Επιτρέπει την επιλογή ενός χρώματος από έναν επιλογέα χρωμάτων και δίνει πληροφορίες σχετικά με την τιμή του χρώματος σε δεκαεξαδική μορφή (π.χ. το #000000 είναι το μαύρο, το προεπιλεγμένο χρώμα αν ο χρήστης δεν καθορίσει μια τιμή χρώματος), όπως επαληθεύεται στο σχήμα της επόμενης σελίδας. Θα πρέπει να σημειωθεί ότι η εισαγωγή χρώματος υποστηρίζεται σε όλα τα μεγάλα σύγχρονα προγράμματα περιήγησης (Firefox, Chrome, Opera, Safari (12.1+), Edge (14+)), αλλά δεν υποστηρίζεται από το Microsoft Internet Explorer και παλαιότερες εκδόσεις των προγραμμάτων περιήγησης Apple Safari.

# Νέοι τύποι εισαγωγής της HTML5



The screenshot shows a web browser window with the following elements:

- Address Bar:** www.tutorialrepublic.com diz
- Page Header:** #0400ff
- Page Content:** Select Color:
- Alert Dialog:** OK

The code editor shows the following HTML and JavaScript code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <title>HTML5 Color Input Type</title>
5 <script>
6     function getValue() {
7         var color = document.getElementById("mycolor").value;
8         alert(color);
9     }
10 </script>
11 </head>
12 <body>
13     <form>
14         <label for="mycolor">Select Color:</label>
15         <input type="color" value="#00ff00" id="mycolor">
16         <button type="button" onclick="getValue();">Get Value</button>
17     </form>
18 </body>
19 </html>
```



## Νέοι τύποι εισαγωγής της HTML5

### Εισαγωγή ημερομηνίας

Ο τύπος εισαγωγής ημερομηνίας επιτρέπει στο χρήστη να επιλέξει μια ημερομηνία από ένα αναπτυσσόμενο ημερολόγιο, στο οποίο μπορεί να επιλέξει το έτος, το μήνα και την ημέρα (αλλά όχι την ώρα). Αυτή η δυνατότητα υποστηρίζεται επίσης από τα περισσότερα προγράμματα περιήγησης, εκτός από το Internet Explorer και το Safari.





# Νέοι τύποι εισαγωγής της HTML5

## Εισαγωγή ημερομηνίας

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>HTML5 Date Input Type</title>
<script>
  function getValue() {
    var date = document.getElementById("mydate").value;
    alert(date);
  }
</script>
</head>
<body>
  <form>
    <label for="mydate">Select Date:</label>
    <input type="date" value="2019-04-15" id="mydate">
    <button type="button" onclick="getValue();">Get Value</button>
  </form>
</body>
</html>
```

Select Date:  



## Νέοι τύποι εισαγωγής της HTML5

### Η εισαγωγή της τοπικής ώρας και ημερομηνίας

Δίνει τη δυνατότητα στο χρήστη να επιλέξει την τοπική ημερομηνία και ώρα, συμπεριλαμβανομένου του έτους, του μήνα και της ημέρας, καθώς και την ώρα σε ώρες και λεπτά. Αυτή η εισαγωγή υποστηρίζεται από το Safari, το Firefox και το IE, αλλά όχι από το Chrome, το Edge και το Opera.



# Νέοι τύποι εισαγωγής της HTML5

## Η εισαγωγή της τοπικής ώρας και ημερομηνίας

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>HTML5 Datetime-local Input Type</title>
<script>
  function getValue() {
    var datetime = document.getElementById("mydatetime").value;
    alert(datetime);
  }
</script>
</head>
<body>
  <form>
    <label for="mydatetime">Choose Date and Time:</label>
    <input type="datetime-local" id="mydatetime">
    <button type="button" onclick="getValue();">Get Value</button>
  </form>
</body>
</html>
```

Choose Date and Time:



# Νέοι τύποι εισαγωγής της HTML5

## Η εισαγωγή ηλεκτρονικού ταχυδρομείου (email)

Ο καλύτερος τύπος εισαγωγής για να εισάγουν οι χρήστες τη διεύθυνση ηλεκτρονικού ταχυδρομείου τους, είναι ο τύπος εισαγωγής **email**. Είναι σαν ένας συνηθισμένος τύπος εισαγωγής κειμένου, αλλά εάν εφαρμοστεί σε συνδυασμό με το χαρακτηριστικό **required**, τα προγράμματα περιήγησης μπορούν να αναζητήσουν τα μοτίβα για να διασφαλίσουν ότι θα εισαχθεί μια σωστά διαμορφωμένη διεύθυνση ηλεκτρονικού ταχυδρομείου. Το πεδίο εισαγωγής email μπορεί να διαμορφωθεί για διαφορετικές καταστάσεις επικύρωσης, όταν μια τιμή εισάγεται χρησιμοποιώντας τις **ψευδοκλάσεις (pseudo-classes)** **:valid**, **:invalid** ή **:required**. Αυτός ο τύπος εισαγωγής υποστηρίζεται από όλα τα μεγάλα προγράμματα περιήγησης.



# Νέοι τύποι εισαγωγής της HTML5

## Η εισαγωγή ηλεκτρονικού ταχυδρομείου

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>HTML5 Email Input Type</title>
<style>
  input[type="email"]:valid{
    outline: 2px solid green;
  }
  input[type="email"]:invalid{
    outline: 2px solid red;
  }
</style>
<script>
  function getValue() {
    var email = document.getElementById("myemail").value;
    alert(email);
  }
</script>
</head>
<body>
  <form>
    <label for="myemail">Enter Email Address:</label>
    <input type="email" id="myemail" required>
    <button type="button" onclick="getValue();">Get Value</button>
  </form>
</body>
</html>
```

Enter Email Address:



## Νέοι τύποι εισαγωγής της HTML5

### Η εισαγωγή του μήνα

Η εισαγωγή του μήνα είναι ένα πολύ παρόμοιο χαρακτηριστικό με τα προηγούμενα, καθώς επιτρέπει την επιλογή ενός **μήνα** και ενός έτους από ένα αναπτυσσόμενο ημερολόγιο (με 'EEEE' για το έτος και 'MM' για τον μήνα. Θα πρέπει να σημειωθεί ότι αυτό δεν υποστηρίζεται από τα προγράμματα περιήγησης Firefox, Safari και Internet Explorer. Υποστηρίζεται μόνο από τα προγράμματα περιήγησης Chrome, Edge και Opera.



# Νέοι τύποι εισαγωγής της HTML5

## Η εισαγωγή του μήνα

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>HTML5 Month Input Type</title>
<script>
  function getValue() {
    var month = document.getElementById("mymonth").value;
    alert(month);
  }
</script>
</head>
<body>
  <form>
    <label for="mymonth">Select Month:</label>
    <input type="month" id="mymonth">
    <button type="button" onclick="getValue();">Get Value</button>
  </form>
</body>
</html>
```

Select Month: fevereiro de 2022 



## Νέοι τύποι εισαγωγής της HTML5

### Η εισαγωγή αριθμού

Όσον αφορά τον τύπο εισαγωγής **αριθμού**, χρησιμοποιείται για την εισαγωγή μιας αριθμητικής τιμής. Ο σχεδιαστής ιστοσελίδων μπορεί επίσης να περιορίσει τον χρήστη έτσι ώστε να μπορεί να εισάγει μόνο αποδεκτές τιμές. Για να συμβεί αυτό, θα πρέπει να χρησιμοποιηθούν τα πρόσθετα χαρακτηριστικά **min**, **max** και **step**. Αυτή η δυνατότητα υποστηρίζεται από όλα τα μεγάλα προγράμματα περιήγησης ιστού.





# Νέοι τύποι εισαγωγής της HTML5

## Η εισαγωγή αριθμού

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>HTML5 Number Input Type</title>
<style>
  input[type="number"]:valid{
    outline: 2px solid green;
  }
  input[type="number"]:invalid{
    outline: 2px solid red;
  }
</style>
<script>
  function getValue() {
    var number = document.getElementById("mynumber").value;
    alert(number);
  }
</script>
</head>
<body>
  <form>
    <label for="mynumber">Enter a Number:</label>
    <input type="number" min="1" max="10" step="0.5" id="mynumber">
    <button type="button" onclick="getValue();">Get Value</button>
  </form>
  <p><strong>Note</strong>: If you try to enter the number out of the range (1-10) or
  text character it will show error.</p>
</body>
</html>
```

Enter a Number:

Note: If you try to enter the number out of the range (1-10) or text character it will show error.



## Νέοι τύποι εισαγωγής της HTML5

### Η εισαγωγή του εύρους

Μπορεί να εφαρμοστεί για την καταχώριση μιας αριθμητικής τιμής εντός ενός συγκεκριμένου εύρους. Λειτουργεί με πολύ παρόμοιο τρόπο με την εισαγωγή **αριθμών** όπως είδαμε, αν και παρουσιάζει έναν ευκολότερο τρόπο για την εισαγωγή ενός αριθμού. Αυτός ο τύπος εισαγωγής υποστηρίζεται από όλα τα μεγάλα προγράμματα περιήγησης.



# Νέοι τύποι εισαγωγής της HTML5

## Η εισαγωγή του εύρους

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>HTML5 Range Input Type</title>
<script>
  function getValue() {
    var number = document.getElementById("mynumber").value;
    alert(number);
  }
</script>
</head>
<body>
  <form>
    <label for="mynumber">Select a Number:</label>
    <input type="range" min="1" max="10" step="0.5" id="mynumber">
    <button type="button" onclick="getValue();">Get Value</button>
  </form>
</body>
</html>
```

Select a Number:



## Νέοι τύποι εισαγωγής της HTML5

### Η εισαγωγή αναζήτησης

Είναι κατάλληλο για τη δημιουργία πεδίων της εισαγωγής αναζήτησης. Αξίζει να αναφερθεί ότι, σε ορισμένα προγράμματα περιήγησης (δηλαδή, Chrome και Safari), μόλις ο χρήστης αρχίσει να πληκτρολογεί στο πλαίσιο αναζήτησης, εμφανίζεται ένας μικροσκοπικός σταυρός στη δεξιά πλευρά του πεδίου, ο οποίος μπορεί να χρησιμοποιηθεί για να καθαριστεί ολόκληρο το πεδίο αναζήτησης. Υποστηρίζεται από όλα τα μεγάλα προγράμματα περιήγησης.



# Νέοι τύποι εισαγωγής της HTML5

## Η εισαγωγή αναζήτησης

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>HTML5 Search Input Type</title>
<script>
  function getValue() {
    var term = document.getElementById("mysearch").value;
    alert(term);
  }
</script>
</head>
<body>
  <form>
    <label for="mysearch">Search Website:</label>
    <input type="search" id="mysearch" placeholder="Type something...">
    <button type="button" onclick="getValue();">Get Value</button>
  </form>
</body>
</html>
```

Search Website:

# Νέοι τύποι εισαγωγής της HTML5

## Η εισαγωγή τηλεφώνου

Αυτός ο τύπος εισαγωγής είναι ιδιαίτερα χρήσιμος για την εισαγωγή ενός αριθμού τηλεφώνου. Καθώς τα προγράμματα περιήγησης δεν υποστηρίζουν την επικύρωση εισαγωγής **τηλεφώνου** από προεπιλογή, το χαρακτηριστικό **placeholder** μπορεί να χρησιμοποιηθεί για να βοηθήσει τους χρήστες να εισάγουν τη σωστή μορφή για τον αριθμό τηλεφώνου τους ή να υποδείξουν μια τυπική έκφραση για την επικύρωση της εισαγωγής του χρήστη εφαρμόζοντας το χαρακτηριστικό **pattern**. Αυτή η δυνατότητα δεν υποστηρίζεται από κανένα πρόγραμμα περιήγησης, καθώς οι αριθμοί τηλεφώνου ποικίλλουν πολύ παγκοσμίως.



# Νέοι τύποι εισαγωγής της HTML5

## Η εισαγωγή τηλεφώνου

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>HTML5 Tel Input Type</title>
<script>
  function getValue() {
    var phone = document.getElementById("myphone").value;
    alert(phone);
  }
</script>
</head>
<body>
  <form>
    <label for="myphone">Telephone Number:</label>
    <input type="tel" id="myphone" placeholder="xx-xxxx-xxxx" required>
    <button type="button" onclick="getValue();" >Get Value</button>
  </form>
</body>
</html>
```

Telephone Number:

Get Value



## Νέοι τύποι εισαγωγής της HTML5

### Η εισαγωγή ώρας

Όσον αφορά τον τύπο εισαγωγής **ώρας**, μπορεί να χρησιμοποιηθεί για την εισαγωγή οποιασδήποτε δεδομένης ώρας (ώρες και λεπτά) και το πρόγραμμα περιήγησης μπορεί να χρησιμοποιήσει και τις δύο μορφές ωρών (12 ή 24 ωρών) για την εισαγωγή ώρας, ανάλογα με την περιοχή. Αυτή η εισαγωγή δεν υποστηρίζεται από προγράμματα περιήγησης IE και Safari.





# Νέοι τύποι εισαγωγής της HTML5

## Η εισαγωγή ώρας

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>HTML5 Time Input Type</title>
<script>
  function getValue() {
    var time = document.getElementById("mytime").value;
    alert(time);
  }
</script>
</head>
<body>
  <form>
    <label for="mytime">Select Time:</label>
    <input type="time" id="mytime">
    <button type="button" onclick="getValue();">Get Value</button>
  </form>
</body>
</html>
```

Select Time: -- : -- ⌚



# Νέοι τύποι εισαγωγής της HTML5

## Η εισαγωγή διεύθυνσης url

Όσον αφορά τον τύπο εισαγωγής **διεύθυνσης url**, μπορεί να χρησιμοποιηθεί για την εισαγωγή διευθύνσεων URL ή ιστοσελίδων. Το χαρακτηριστικό **multiple** μπορεί να χρησιμοποιηθεί για την εισαγωγή περισσότερων από μία διευθύνσεων URL. Επιπλέον, εάν το χαρακτηριστικό **required** δεν δύναται να χρησιμοποιηθεί, το πρόγραμμα περιήγησης θα εκτελέσει αυτόματα επικύρωση για να διασφαλίσει ότι μόνο το κείμενο που πληροί την τυπική μορφή για τις διευθύνσεις URL θα μπει στο πλαίσιο εισαγωγής. Όλα τα μεγάλα προγράμματα περιήγησης υποστηρίζουν αυτόν τον τύπο εισαγωγής.



# Νέοι τύποι εισαγωγής της HTML5

## Η εισαγωγή διεύθυνσης url

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>HTML5 URL Input Type</title>
<style>
  input[type="url"]:valid{
    outline: 2px solid green;
  }
  input[type="url"]:invalid{
    outline: 2px solid red;
  }
</style>
<script>
  function getValue() {
    var url = document.getElementById("myurl").value;
    alert(url);
  }
</script>
</head>
<body>
  <form>
    <label for="myurl">Enter Website URL:</label>
    <input type="url" id="myurl" required>
    <button type="button" onclick="getValue();">Get Value</button>
  </form>
  <p><strong>Note</strong>: Enter URL in the form like https://www.google.com</p>
</body>
</html>
```

Enter Website URL:

**Note:** Enter URL in the form like <https://www.google.com>



## Νέοι τύποι εισαγωγής της HTML5

### Η εισαγωγή της εβδομάδας

Τέλος, ο τύπος εισαγωγής **εβδομάδας** επιτρέπει στο χρήστη να επιλέξει μια εβδομάδα και ένα έτος από ένα αναπτυσσόμενο ημερολόγιο. Αυτή η δυνατότητα δεν υποστηρίζεται από Firefox, Safari και IE, αλλά προς το παρόν υποστηρίζεται από τα προγράμματα περιήγησης Chrome, Edge και Opera.



# Νέοι τύποι εισαγωγής της HTML5

## Η εισαγωγή της εβδομάδας

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>HTML5 Week Input Type</title>
<script>
  function getValue() {
    var week = document.getElementById("myweek").value;
    alert(week);
  }
</script>
</head>
<body>
  <form>
    <label for="myweek">Select Week:</label>
    <input type="week" id="myweek">
    <button type="button" onclick="getValue();">Get Value</button>
  </form>
</body>
</html>
```

Select Week:

# Ο Καμβάς στην HTML5

- Αυτή η ενότητα θα είναι χρήσιμη για να μάθετε πώς να σχεδιάζετε γραφικά χρησιμοποιώντας το στοιχείο καμβά (canvas element) στην HTML5. Δημιουργήθηκε αρχικά από την Apple για τα γραφικά στοιχεία του πίνακα ελέγχου Mac OS και για την ενεργοποίηση γραφικών στο Safari. Επιπλέον, υιοθετήθηκε από άλλα προγράμματα περιήγησης, όπως το Firefox, το Google Chrome και το Opera.
- Από προεπιλογή, το στοιχείο `< canvas >` έχει πλάτος 300px και ύψος 150px χωρίς περίγραμμα και περιεχόμενο. Ωστόσο, το προσαρμοσμένο πλάτος και ύψος μπορούν να καθοριστούν χρησιμοποιώντας την ιδιότητα ύψους και πλάτους της CSS.
- Ο καμβάς είναι μια δισδιάστατη τετράπλευρη περιοχή. Οι συντεταγμένες της επάνω αριστερής γωνίας του καμβά είναι (0, 0), οι οποίες προσδιορίζονται ως προέλευση και οι συντεταγμένες της κάτω δεξιάς γωνίας είναι (πλάτος καμβά, ύψος καμβά), όπως φαίνεται με τη χρήση του διαδραστικού εργαλείου που είναι διαθέσιμο [εδώ](#).
- Για να σχεδιάσετε βασικές διαδρομές και σχήματα χρησιμοποιώντας το στοιχείο καμβά της HTML5 που μόλις είδαμε και τη JavaScript, μπορείτε να ρίξετε μια ματιά σε πολλά πρότυπα.

# Ο Καμβάς στην HTML5

Ας δούμε πρώτα το βασικό πρότυπο για τη σχεδίαση διαδρομών και σχημάτων στον δισδιάστατο καμβά της HTML5:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <title>Embedding Canvas Into HTML Pages</title>
6 <style>
7     canvas {
8         border: 1px solid #000;
9     }
10 </style>
11 <script>
12     window.onload = function() {
13         var canvas = document.getElementById("myCanvas");
14         var context = canvas.getContext("2d");
15         // draw stuff here
16     };
17 </script>
18 </head>
19 <body>
20     <canvas id="myCanvas" width="300" height="200"></canvas>
21 </body>
22 </html>
```





## Ο Καμβάς στην HTML5

Όλες οι σειρές εκτός από αυτές από το 7 έως το 11 είναι αρκετά απλές και κατανοητές. Η ανώνυμη συνάρτηση που έχει ενσωματωθεί στο συμβάν `window.onload` θα εκτελεστεί όταν φορτώσει η σελίδα. Μόλις φορτώσει η σελίδα, ο χρήστης μπορεί να έχει πρόσβαση στο στοιχείο `<canvas>` με τη μέθοδο `document.getElementById()`. Αργότερα, ορίζεται ένα πλαίσιο δισδιάστατου καμβά εισάγοντας το 2d στη μέθοδο `getContext()` του αντικειμένου του καμβά.



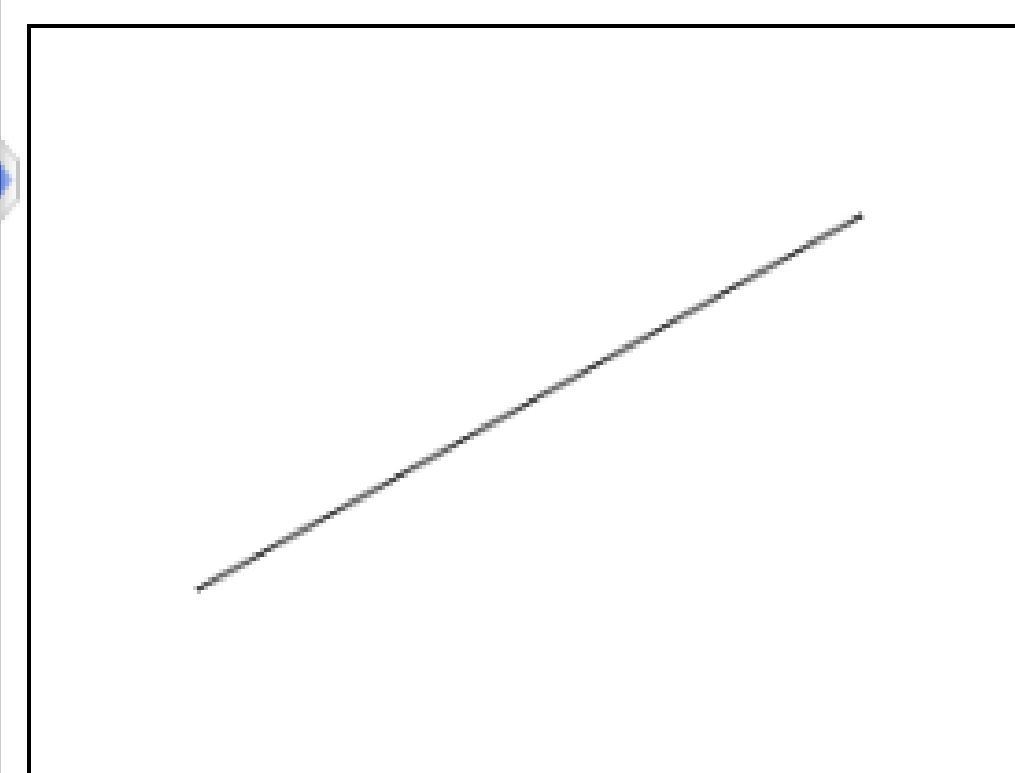


## Ο Καμβάς στην HTML5

Το πρώτο βήμα για να σχεδιάσετε σε καμβά είναι να σχεδιάσετε μια **ευθεία γραμμή**. Οι πιο σημαντικές διαδικασίες που χρησιμοποιούνται για αυτό το σκοπό είναι οι **moveTo()**, **lineTo()** και το **stroke()**. Η μέθοδος **moveTo()** προσδιορίζει τη θέση του κέρσορα σχεδίασης στον καμβά, ενώ η μέθοδος **lineTo ()** χρησιμοποιείται για τον καθορισμό των συντεταγμένων σημείου όπου τελειώνει η γραμμή και τέλος η μέθοδος **stroke()** χρησιμοποιείται για να κάνει τη γραμμή ορατή:

# Ο Καμβάς στην HTML5

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <title>Drawing a Line on the Canvas</title>
6 <style>
7     canvas {
8         border: 1px solid #000;
9     }
10 </style>
11 <script>
12     window.onload = function() {
13         var canvas = document.getElementById("myCanvas");
14         var context = canvas.getContext("2d");
15         context.moveTo(50, 150);
16         context.lineTo(250, 50);
17         context.stroke();
18     };
19 </script>
20 </head>
21 <body>
22     <canvas id="myCanvas" width="300" height="200"></canvas>
23 </body>
24 </html>
```





## Ο Καμβάς στην HTML5

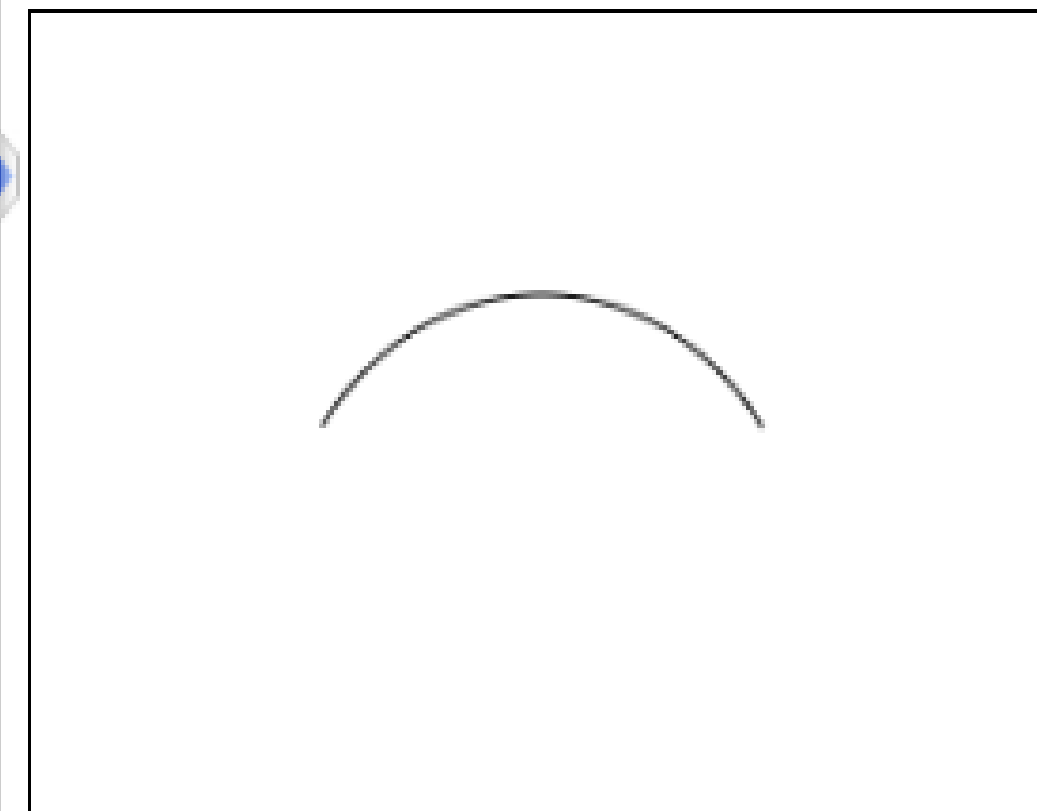
Και πώς μπορούμε να σχεδιάσουμε μια καμπύλη; Μπορούμε να τη σχεδιάσουμε χρησιμοποιώντας απλά τη μέθοδο `arc()`, η οποία συντάσσεται ως εξής:

```
context.arc (centerX, centerY, radius,  
startingAngle, endingAngle, counterclockwise);
```

Στο παράδειγμα της επόμενης διαφάνειας, σχεδιάστηκε μια καμπύλη σε καμβά εισάγοντας έναν κώδικα JavaScript:

# Ο Καμβάς στην HTML5

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Drawing an Arc on the Canvas</title>
<style>
  canvas {
    border: 1px solid #000;
  }
</style>
<script>
  window.onload = function() {
    var canvas = document.getElementById("myCanvas");
    var context = canvas.getContext("2d");
    context.arc(150, 150, 80, 1.2 * Math.PI, 1.8 * Math.PI, false);
    context.stroke();
  };
</script>
</head>
<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
</body>
</html>
```



## Ο Καμβάς στην HTML5

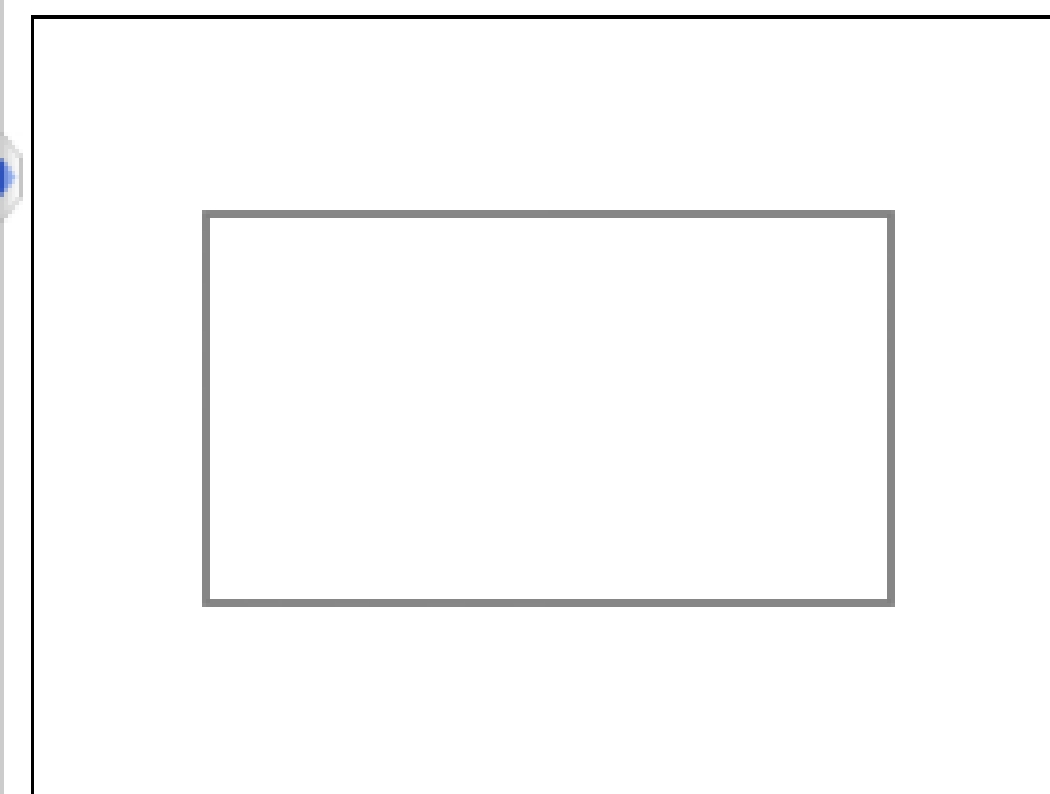
Η κατάλληλη μέθοδος για τον σχεδιασμό ορθογώνιων και τετράγωνων σχημάτων είναι η μέθοδος **rect()**. Συνεπάγεται τέσσερις παραμέτρους: τις χ, ψ θέσεις του ορθογωνίου, το πλάτος και το ύψος του. Η βασική σύνταξη της μεθόδου **rect()** είναι η εξής:

```
context.rect (x, y, width, height);
```

# Ο Καμβάς στην HTML5

Για τον σχεδιασμό ορθογωνίου χρησιμοποιώντας έναν κώδικα JS:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Drawing a Rectangle on the Canvas</title>
<style>
  canvas {
    border: 1px solid #000;
  }
</style>
<script>
  window.onload = function() {
    var canvas = document.getElementById("myCanvas");
    var context = canvas.getContext("2d");
    context.rect(50, 50, 200, 100);
    context.stroke();
  };
</script>
</head>
<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
</body>
</html>
```





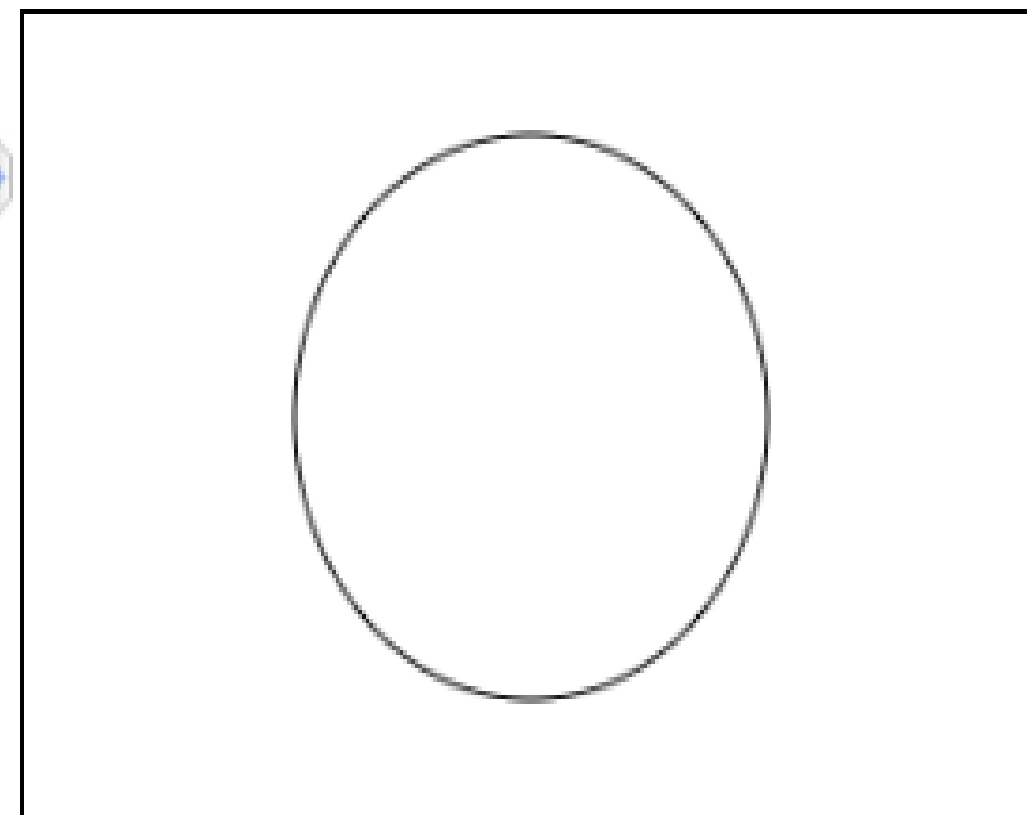
## Ο Καμβάς στην HTML5

Σε αντίθεση με τη μέθοδο `rect()`, δεν υπάρχει συγκεκριμένη διαδικασία για τη σχεδίαση ενός κύκλου. Ωστόσο, αυτό μπορεί να επιτευχθεί δημιουργώντας μια πλήρως κλειστή καμπύλη, χρησιμοποιώντας τη μέθοδο `arc()`. Η σύνταξη για τη σχεδίαση ενός πλήρους κύκλου χρησιμοποιώντας τη μέθοδο `arc()` είναι η εξής:

```
context.arc(centerX, centerY, radius, 0, 2 * Math.PI, false);
```

# Ο Καμβάς στην HTML5

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Drawing a Circle on the Canvas</title>
<style>
  canvas {
    border: 1px solid #000;
  }
</style>
<script>
  window.onload = function() {
    var canvas = document.getElementById("myCanvas");
    var context = canvas.getContext("2d");
    context.arc(150, 100, 70, 0, 2 * Math.PI, false);
    context.stroke();
  };
</script>
</head>
<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
</body>
</html>
```





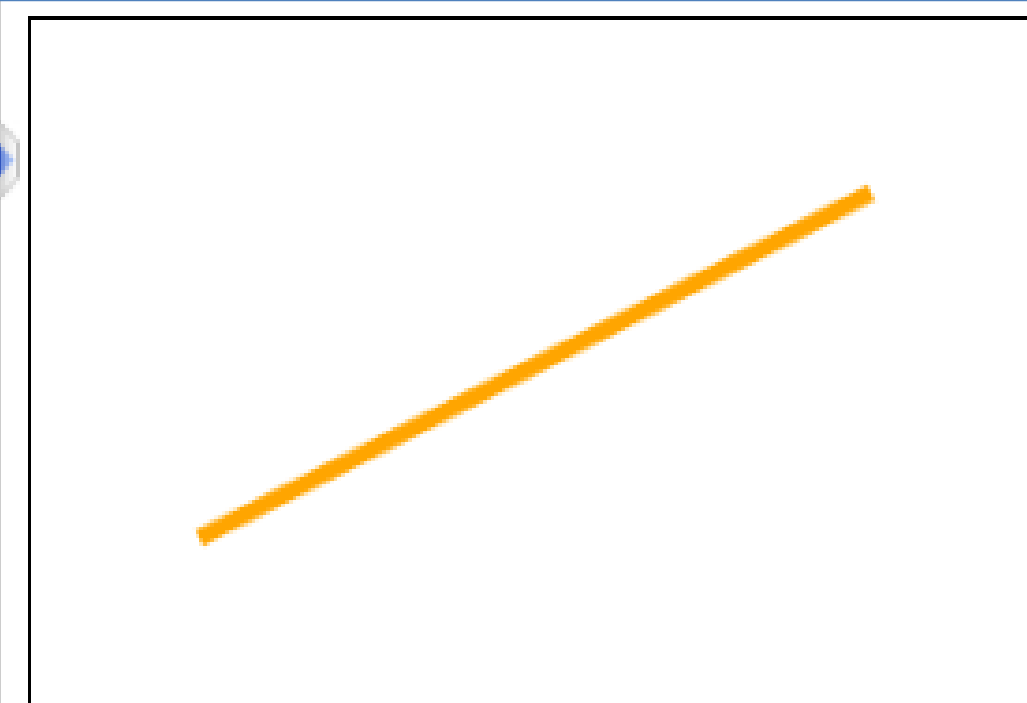


## Ο Καμβάς στην HTML5

Όσον αφορά το **στυλ και το χρώμα σχεδίασης**, το προεπιλεγμένο χρώμα είναι το μαύρο, με πάχος 1 pixel. Ωστόσο, αυτά τα χαρακτηριστικά μπορούν να αλλάξουν χρησιμοποιώντας τις ιδιότητες **strokeStyle** και **lineWidth**, όπως φαίνεται στην επόμενη διαφάνεια.

# Ο Καμβάς στην HTML5

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Setting Stroke Color and Width on the Canvas</title>
<style>
  canvas {
    border: 1px solid #000;
  }
</style>
<script>
  window.onload = function() {
    var canvas = document.getElementById("myCanvas");
    var context = canvas.getContext("2d");
    context.lineWidth = 5;
    context.strokeStyle = "orange";
    context.moveTo(50, 150);
    context.lineTo(250, 50);
    context.stroke();
  };
</script>
</head>
<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
</body>
</html>
```



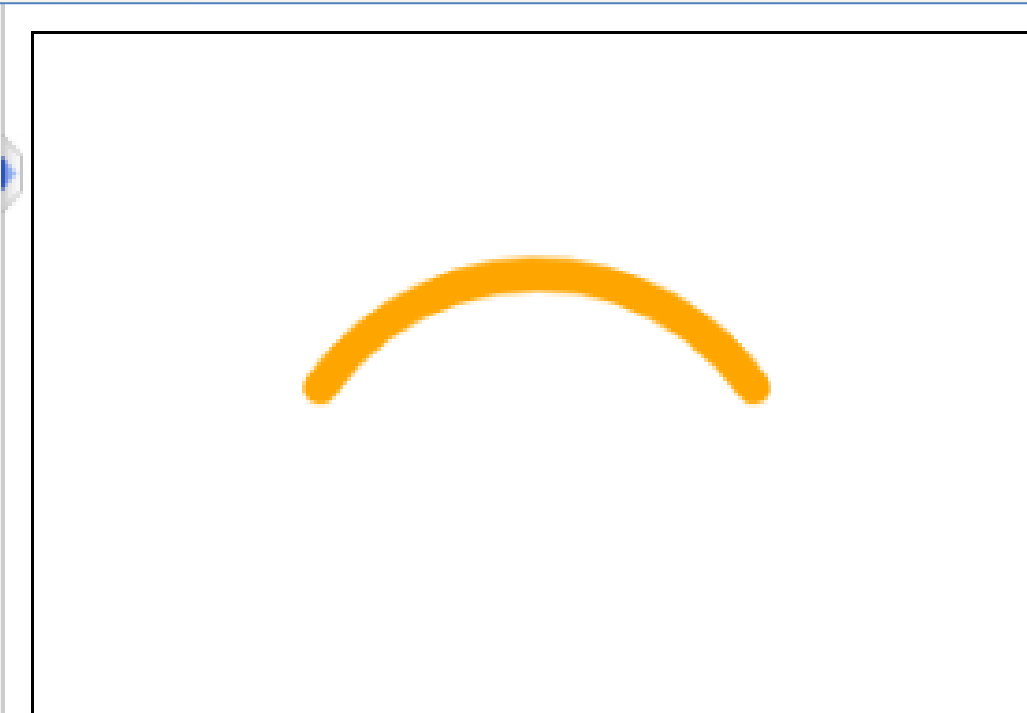


## Ο Καμβάς στην HTML5

Έχετε επίσης τη δυνατότητα να ορίσετε το άκρο του κέρσορα σχεδίασης των γραμμών χρησιμοποιώντας την ιδιότητα **lineCap**, με τρία διαθέσιμα στυλ: επίπεδο, στρογγυλό και τετράγωνο. Αυτό φαίνεται στο παράδειγμα της επόμενης διαφάνειας.

# Ο Καμβάς στην HTML5

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Setting Stroke Cap Style on the Canvas</title>
<style>
  canvas {
    border: 1px solid #000;
  }
</style>
<script>
  window.onload = function() {
    var canvas = document.getElementById("myCanvas");
    var context = canvas.getContext("2d");
    context.lineWidth = 10;
    context.strokeStyle = "orange";
    context.lineCap = "round";
    context.arc(150, 150, 80, 1.2 * Math.PI, 1.8 * Math.PI, false);
    context.stroke();
  };
</script>
</head>
<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
</body>
</html>
```



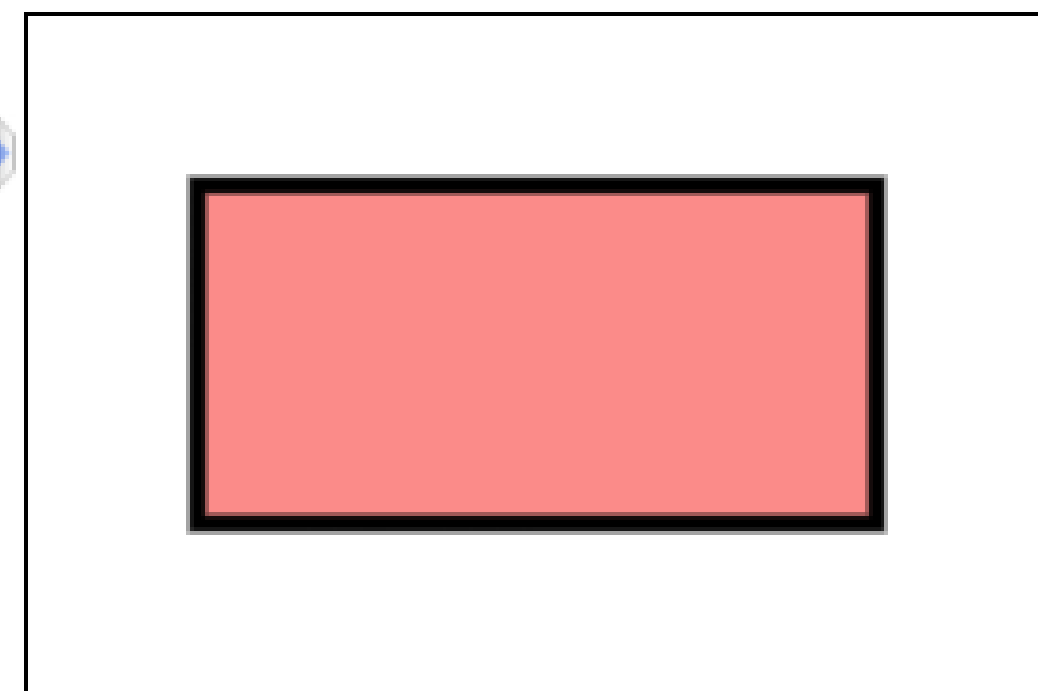


## Ο Καμβάς στην HTML5

Μπορείτε επίσης να γεμίσετε με χρώμα μέσα στα σχήματα του καμβά χρησιμοποιώντας **το fillStyle()**. Η εικόνα της επόμενης διαφάνειας δείχνει πώς να γεμίσετε ένα ορθογώνιο σχήμα με ένα συμπαγές χρώμα. Κατά τη σχεδίαση των σχημάτων σε καμβά, προτείνεται η χρήση της μεθόδου **fill()** πριν από τη μέθοδο **stroke()** για την κατάλληλη απόδοση του τρόπου σχεδίασης.

# Ο Καμβάς στην HTML5

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Filling Color inside a Rectangle on the Canvas</title>
<style>
  canvas {
    border: 1px solid #000;
  }
</style>
<script>
  window.onload = function() {
    var canvas = document.getElementById("myCanvas");
    var context = canvas.getContext("2d");
    context.rect(50, 50, 200, 100);
    context.fillStyle = "#FB8B89";
    context.fill();
    context.lineWidth = 5;
    context.strokeStyle = "black";
    context.stroke();
  };
</script>
</head>
<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
</body>
</html>
```





## Ο Καμβάς στην HTML5

Μπορείτε επίσης να προσθέσετε χρώμα με βαθμιαία αλλαγή (μια ομαλή οπτική μετάβαση από το ένα χρώμα στο άλλο) μέσα στα σχήματα του καμβά. Υπάρχουν δύο τύποι βαθμιαίων αλλαγών εδώ: γραμμικές και ακτινικές.

Η βασική σύνταξη για τη δημιουργία μιας **γραμμικής διαβάθμισης** είναι:

```
var grd = context.createLinearGradient (startX, startY, endX, endY);
```

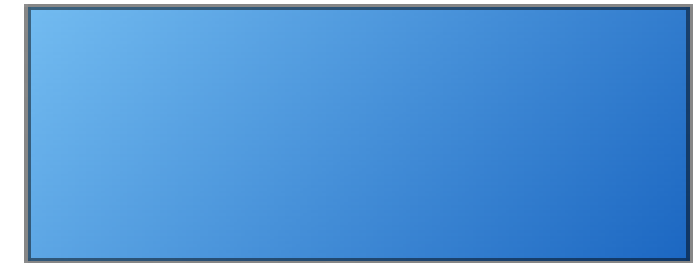
Η βασική σύνταξη για τη δημιουργία μιας **ακτινικής διαβάθμισης** είναι:

```
var grd = context.createRadialGradient (startX, startY, startRadius, endX, endY, endRadius);
```

# Ο Καμβάς στην HTML5

Η παρακάτω εικόνα δείχνει πώς να γεμίσετε ένα χρώμα με γραμμική διαβάθμιση μέσα σε ένα ορθογώνιο χρησιμοποιώντας τη μέθοδο `createLinearGradient ()`:

```
<!DOCTYPE HTML>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Filling Linear Gradient inside Canvas Shapes</title>
<style>
  canvas {
    border: 1px solid #000;
  }
</style>
<script>
  window.onload = function() {
    var canvas = document.getElementById("myCanvas");
    var context = canvas.getContext("2d");
    context.rect(50, 50, 200, 100);
    var grd = context.createLinearGradient(0, 0, canvas.width, canvas.height);
    grd.addColorStop(0, '#8ED6FF');
    grd.addColorStop(1, '#004CB3');
    context.fillStyle = grd;
    context.fill();
    context.stroke();
  };
</script>
</head>
<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
</body>
</html>
```

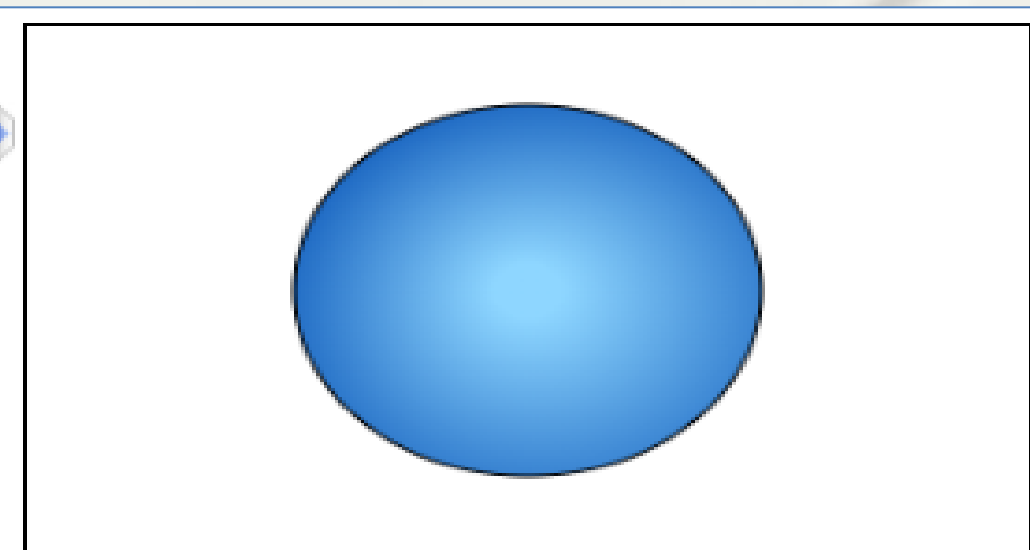




# Ο Καμβάς στην HTML5

Η παρακάτω εικόνα δείχνει πώς να γεμίσετε ένα χρώμα με ακτινική διαβάθμιση μέσα σε έναν κύκλο με τη μέθοδο `createRadialGradient()`:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Filling Radial Gradient inside Canvas Shapes</title>
<style>
  canvas {
    border: 1px solid #000;
  }
</style>
<script>
  window.onload = function() {
    var canvas = document.getElementById("myCanvas");
    var context = canvas.getContext("2d");
    context.arc(150, 100, 70, 0, 2 * Math.PI, false);
    var grd = context.createRadialGradient(150, 100, 10, 160, 110, 100);
    grd.addColorStop(0, '#8ED6FF');
    grd.addColorStop(1, '#004CB3');
    context.fillStyle = grd;
    context.fill();
    context.stroke();
  };
</script>
</head>
<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
</body>
</html>
```





## Ο Καμβάς στην HTML5

Επιπλέον, μπορείτε να **σχεδιάσετε κείμενο** σε καμβά (που περιέχει μόνο χαρακτήρες Unicode), να προσθέσετε **χρώμα και στοίχιση**, καθώς και πινελιές στο κείμενο χρησιμοποιώντας τη μέθοδο **strokeText()**, η οποία θα χρωματίσει την περίμετρο του κειμένου αντί να το γεμίσει .

Ωστόσο, εάν ο σχεδιαστής ιστοσελίδων θέλει να ορίσει τόσο το χρώμα που θα γεμίσει αλλά και της περιμέτρου του κειμένου, μπορεί να χρησιμοποιήσει τις μεθόδους **fillText ()** και **strokeText ()** μαζί. Προτείνεται η χρήση της μεθόδου **fillText ()** πριν από τη μέθοδο **strokeText ()** για την ακριβή απόδοση του τρόπου σχεδίασης.

# Ο Καμβάς στην HTML5

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Adding Stroke to Canvas Text</title>
<style>
  canvas {
    border: 1px solid #000;
  }
</style>
<script>
  window.onload = function() {
    var canvas = document.getElementById("myCanvas");
    var context = canvas.getContext("2d");
    context.font = "bold 32px Arial";
    context.textAlign = "center";
    context.textBaseline = "middle";
    context.strokeStyle = "blue";
    context.strokeText("Code4SP", 150, 100);
  };
</script>
</head>
<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
</body>
</html>
```

Code4SP



# Τα γραφικά SVG στην HTML5

- Αυτή η υποενότητα αποσκοπεί στο να διδάξει τον τρόπο χρήσης των στοιχείων SVG της HTML5 για τη σχεδίαση διανυσματικών γραφικών σε μια ιστοσελίδα. Για να γίνει αυτό, πρέπει πρώτα να ορίσουμε τι είναι τα στοιχεία **SVG**.
- Το **SVG** σημαίνει Scalable Vector Graphics και είναι μια μορφή εικόνας που βασίζεται στην γλώσσα σήμανσης XML που χρησιμοποιείται για τον καθορισμό δισδιάστατων διανυσματικών γραφικών για τον Ιστό. Σε αντίθεση με τις εικόνες ράστερ (π.χ. .jpg, .gif, .png και άλλες δισδιάστατες μορφές), μια διανυσματική εικόνα μπορεί να κλιμακωθεί προς τα πάνω ή προς τα κάτω σε οποιοδήποτε βαθμό χωρίς να χάσει την ποιότητα της εικόνας. Οι διανυσματικές εικόνες αποτελούνται από μια σειρά σχημάτων που βασίζονται στα μαθηματικά, ενώ οι εικόνες ράστερ αποτελούνται από ένα σταθερό σύνολο κουκκίδων (pixel).



# Τα γραφικά SVG στην HTML5

Μια εικόνα SVG δημιουργείται χρησιμοποιώντας μια ακολουθία εντολών που συμβαδίζουν με το σχήμα XML (schema), επομένως, οι **εικόνες SVG** μπορούν να δημιουργηθούν και να τροποποιηθούν με ένα πρόγραμμα επεξεργασίας κειμένου όπως το Notepad. Υπάρχουν πολλά πλεονεκτήματα από τη χρήση εικόνων SVG έναντι άλλων μορφών εικόνας, ως εξής:

- Μπορούν να αναζητηθούν, να ευρετηριαστούν, να δημιουργηθούν σενάρια και να συμπιεστούν.
- Μπορούν να δημιουργηθούν και να τροποποιηθούν χρησιμοποιώντας JavaScript σε πραγματικό χρόνο.
- Μπορούν να εκτυπωθούν με υψηλή ποιότητα σε οποιαδήποτε ανάλυση.
- Μπορούν να γίνουν κινούμενες χρησιμοποιώντας τα ενσωματωμένα στοιχεία κινούμενων γραφικών.
- Μπορούν να περιέχουν υπερσυνδέσμους προς άλλα έγγραφα.

Όλα τα μεγάλα σύγχρονα προγράμματα περιήγησης ιστού (Chrome, Firefox, Safari και Opera), καθώς και ο Internet Explorer 9 και νεότερη έκδοση είναι συμβατά με ενσωματωμένη απόδοση SVG.



# Τα γραφικά SVG στην HTML5

Τα γραφικά SVG μπορούν να ενσωματωθούν απευθείας σε ένα έγγραφο χρησιμοποιώντας το στοιχείο `<svg>` HTML5 (όπως φαίνεται παρακάτω):

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Embedding SVG Into HTML Pages</title>
<style>
  svg {
    border: 1px solid black;
  }
</style>
</head>
<body>
  <svg width="300" height="200">
    <text x="10" y="20" style="font-size:14px;">
      Your browser support SVG.
    </text>
    Sorry, your browser does not support SVG.
  </svg>
</body>
</html>
```

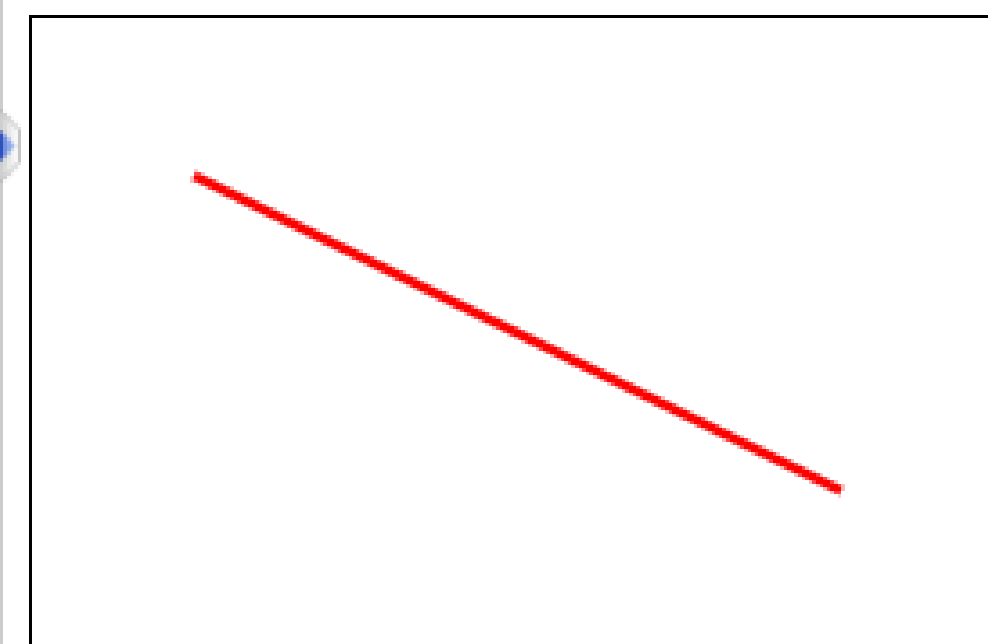
Your browser support SVG.

# Τα γραφικά SVG στην HTML5

Βασικές διαδρομές και σχήματα που βασίζονται σε διανύσματα στις ιστοσελίδες μπορούν να σχεδιαστούν χρησιμοποιώντας το στοιχείο `<svg>` στην HTML5.

Η βασική διαδρομή για να δουλέψουμε με SVG είναι να **σχεδιάσουμε μια ευθεία γραμμή**. Για να συμβεί αυτό, θα πρέπει να χρησιμοποιηθεί το στοιχείο `<line>` σε SVG. Όπως φαίνεται παρακάτω, τα χαρακτηριστικά `x1`, `x2`, `y1` και `y2` του στοιχείου σχεδιάζουν μια γραμμή από `(x1,y1)` έως `(x2,y2)`.

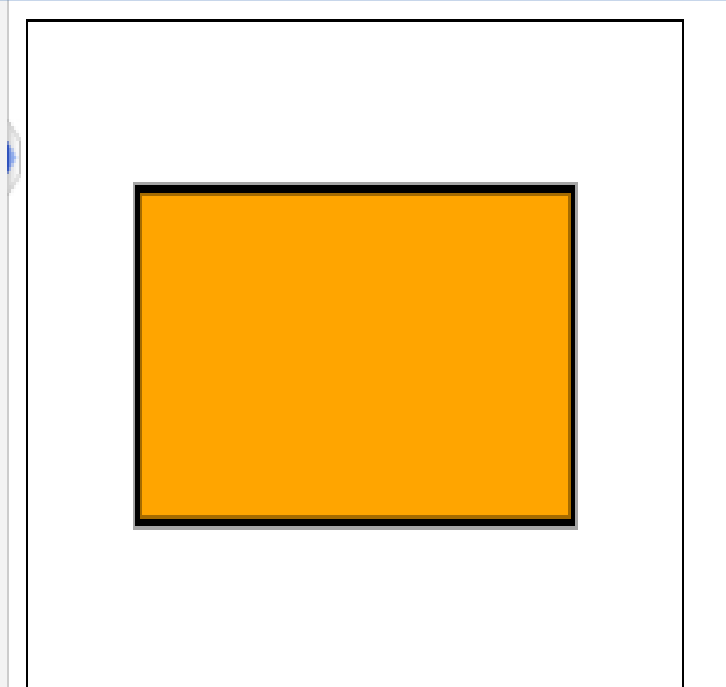
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Create a Line with HTML5 SVG</title>
<style>
  svg {
    border: 1px solid black;
  }
</style>
</head>
<body>
  <svg width="300" height="200">
    <line x1="50" y1="50" x2="250" y2="150" style="stroke:red; stroke-width:3;" />
  </svg>
</body>
</html>
```



# Τα γραφικά SVG στην HTML5

Ο καταλληλότερος τρόπος για τον σχεδιασμό **ορθογωνίων** και **τετραγώνων**, είναι το στοιχείο `< rect >` σε SVG. Τα χαρακτηριστικά `x` και `y` του στοιχείου `< rect >` καθορίζουν τις συντεταγμένες της επάνω αριστερής γωνίας του ορθογωνίου. Τα χαρακτηριστικά `width` και `height` καθορίζουν το πλάτος και το ύψος του σχήματος.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Create a Rectangle with HTML5 SVG</title>
<style>
  svg {
    border: 1px solid black;
  }
</style>
</head>
<body>
  <svg width="300" height="200">
    <rect x="50" y="50" width="200" height="100" style="fill:orange; stroke:black;
stroke-width:3;" />
  </svg>
</body>
</html>
```

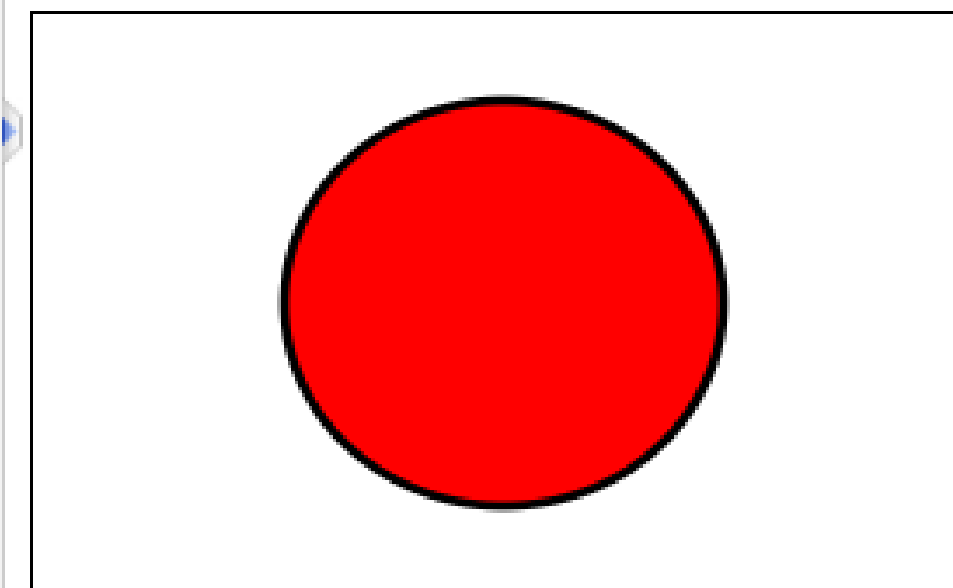




# Τα γραφικά SVG στην HTML5

Για τον σχεδιασμό ενός **κύκλου**, το πιο κατάλληλο στοιχείο SVG είναι το στοιχείο `<circle>`. Τα χαρακτηριστικά `cx` και `cy` του στοιχείου `<circle>` καθορίζουν τις συντεταγμένες του κέντρου του κύκλου και το χαρακτηριστικό `r` προσδιορίζει την ακτίνα του κύκλου. Επίσης, εάν τα χαρακτηριστικά `cx` και `cy` απουσιάζουν ή δεν καθορίζονται, το κέντρο του κύκλου ορίζεται σε (0,0).

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Create a Circle with HTML5 SVG</title>
<style>
  svg {
    border: 1px solid black;
  }
</style>
</head>
<body>
  <svg width="300" height="200">
    <circle cx="150" cy="100" r="70" style="fill:red; stroke:black; stroke-width:3;"
  />
</svg>
</body>
</html>
```





# Τα γραφικά SVG στην HTML5

Έχουμε επίσης τη δυνατότητα να **σχεδιάσουμε κείμενο** σε μορφή SVG. Το κείμενο σε SVG αποδίδεται ως ένα γραφικό, επομένως ο σχεδιαστής Ιστού μπορεί να χρησιμοποιήσει όλο τον μετασχηματισμό γραφικών σε αυτό, ενώ θα εξακολουθεί να λειτουργεί ως κείμενο, ώστε να μπορεί να επιλεγεί και να αντιγραφεί ως κείμενο από τον χρήστη. Τα χαρακτηριστικά x και y του στοιχείου `< text >` προσδιορίζουν τη θέση της επάνω αριστερής γωνίας σε απόλυτες τιμές, αν και τα χαρακτηριστικά dx και dy υποδηλώνουν τη σχετική θέση.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Render Text with HTML5 SVG</title>
<style>
  svg {
    border: 1px solid black;
  }
</style>
</head>
<body>
  <svg width="300" height="200">
    <text x="20" y="30" style="fill:purple; font-size:22px;">
      Welcome to Our Website!
    </text>
    <text x="20" y="30" dx="0" dy="20" style="fill:navy; font-size:14px;">
      Here you will find lots of useful information.
    </text>
  </svg>
</body>
</html>
```

Welcome to Our Website!  
Here you will find lots of useful information.



# Τα γραφικά SVG στην HTML5

Εναλλακτικά, οι σχεδιαστές ιστοσελίδων μπορούν να χρησιμοποιήσουν το στοιχείο `<tspan>` για να μορφοποιήσετε το κείμενο που περιλαμβάνεται σε ένα στοιχείο `<text>`. Το κείμενο περιλαμβάνεται σε ξεχωριστά `tspan`, αλλά μέσα στο ίδιο στοιχείο κειμένου μπορούν να επιλεγούν όλα ταυτόχρονα, όταν κάνετε κλικ και σύρετε για να επιλέξετε το κείμενο. Ωστόσο, τα κείμενα εντός ξεχωριστών στοιχείων `<text>` δεν μπορούν να επιλεγθούν ταυτόχρονα.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Rotate and Render Text with HTML5 SVG</title>
<style>
  svg {
    border: 1px solid black;
  }
</style>
</head>
<body>
  <svg width="300" height="200">
    <text x="30" y="15" style="fill:purple; font-size:22px; transform:rotate(30deg);">
      <tspan style="fill:purple; font-size:22px;">
        Welcome to Our Website!
      </tspan>
      <tspan dx="-230" dy="20" style="fill:navy; font-size:14px;">
        Here you will find lots of useful information.
      </tspan>
    </text>
  </svg>
</body>
</html>
```





# Τα γραφικά SVG στην HTML5

Παρόλο που έχουμε δει τα νέα στοιχεία γραφικών `<canvas>` και `<svg>` στην HTML 5, αυτά τα δύο στοιχεία διαφέρουν αρκετά στη δημιουργία γραφικών υψηλής ποιότητας στο διαδίκτυο.

Στον παρακάτω πίνακα, συνοψίζονται οι διαφορές μεταξύ των δύο, για να βοηθήσει τους μαθητές να τα χρησιμοποιήσουν με τον κατάλληλο, αποτελεσματικό τρόπο:

SVG	Canvas
Βασίζεται σε διανύσματα (που αποτελούνται από σχήματα)	Βασίζεται σε ράστερ (που αποτελούνται από pixel)
Πολλά γραφικά στοιχεία, τα οποία γίνονται μέρος του δέντρου DOM της σελίδας	Μεμονωμένο στοιχείο με παρόμοια λειτουργία με το στοιχείο <code>&lt;img&gt;</code> . Το διάγραμμα canvas μπορεί να αποθηκευτεί σε μορφή PNG ή JPG
Τροποποιείται μέσω σεναρίου και CSS	Τροποποιείται μόνο μέσω σεναρίου
Καλές δυνατότητες απόδοσης κειμένου	Κακές δυνατότητες απόδοσης κειμένου
Έχει καλύτερη απόδοση με μικρότερο αριθμό αντικειμένων ή μεγαλύτερη επιφάνεια ή και τα δύο	Έχει καλύτερη απόδοση με μεγαλύτερο αριθμό αντικειμένων ή μικρότερη επιφάνεια ή και με τα δύο
Καλύτερη δυνατότητα κλιμάκωσης. Μπορούν να εκτυπωθούν σε υψηλή ποιότητα σε οποιαδήποτε ανάλυση. Δεν προκύπτει rixelation (παραμόρφωση των εικονοστοιχείων)	Μικρή δυνατότητα κλιμάκωσης. Δεν είναι κατάλληλο για εκτύπωση σε υψηλότερη ανάλυση. Ενδέχεται να προκύψει rixelation (παραμόρφωση των εικονοστοιχείων)



# Τα γραφικά SVG στην HTML5

Παρόλο που έχουμε δει τα νέα στοιχεία γραφικών `<canvas>` και `<svg>` στην HTML 5, αυτά τα δύο στοιχεία διαφέρουν αρκετά στη δημιουργία γραφικών υψηλής ποιότητας στο διαδίκτυο.

Στον παρακάτω πίνακα, συνοψίζονται οι διαφορές μεταξύ των δύο, για να βοηθήσει τους μαθητές να τα χρησιμοποιήσουν με τον κατάλληλο, αποτελεσματικό τρόπο:

SVG	Canvas
Βασίζεται σε διανύσματα (που αποτελούνται από σχήματα)	Βασίζεται σε ράστερ (που αποτελούνται από pixel)
Πολλά γραφικά στοιχεία, τα οποία γίνονται μέρος του δέντρου DOM της σελίδας	Μεμονωμένο στοιχείο με παρόμοια λειτουργία με το στοιχείο <code>&lt;img&gt;</code> . Το διάγραμμα canvas μπορεί να αποθηκευτεί σε μορφή PNG ή JPG
Τροποποιείται μέσω σεναρίου και CSS	Τροποποιείται μόνο μέσω σεναρίου
Καλές δυνατότητες απόδοσης κειμένου	Κακές δυνατότητες απόδοσης κειμένου
Έχει καλύτερη απόδοση με μικρότερο αριθμό αντικειμένων ή μεγαλύτερη επιφάνεια ή και τα δύο	Έχει καλύτερη απόδοση με μεγαλύτερο αριθμό αντικειμένων ή μικρότερη επιφάνεια ή και με τα δύο
Καλύτερη δυνατότητα κλιμάκωσης. Μπορούν να εκτυπωθούν σε υψηλή ποιότητα σε οποιαδήποτε ανάλυση. Δεν προκύπτει rixelation (παραμόρφωση των εικονοστοιχείων)	Μικρή δυνατότητα κλιμάκωσης. Δεν είναι κατάλληλο για εκτύπωση σε υψηλότερη ανάλυση. Ενδέχεται να προκύψει rixelation (παραμόρφωση των εικονοστοιχείων)



# Ήχος στην HTML5

- Αυτό το υποκεφάλαιο έχει σκοπό να εξηγήσει πώς να ενσωματώσετε τον ήχο σε ένα έγγραφο HTML.
- Καθώς τα προγράμματα περιήγησης ιστού δεν διέθεταν στο παρελθόν ένα ενιαίο πρότυπο για την ενσωμάτωση αρχείων πολυμέσων ως ήχου, η ενσωμάτωση ήχου δεν ήταν εύκολη υπόθεση. Ωστόσο, τώρα υπάρχουν πολλοί τρόποι για να ενσωματώσετε τον ήχο σε μια ιστοσελίδα, από την απλή χρήση ενός απλού συνδέσμου έως τη χρήση του στοιχείου `<audio>` της HTML5. Αυτό το στοιχείο παρέχει έναν τυπικό τρόπο εισαγωγής ήχου σε ιστοσελίδες. Δεδομένου ότι το στοιχείο ήχου είναι κάπως καινούργιο, εκτελείται στα περισσότερα σύγχρονα προγράμματα περιήγησης ιστού.
- Υπάρχουν πολλοί τρόποι εισαγωγής ήχου σε ένα έγγραφο HTML5. Ένα από αυτά είναι η χρήση της προεπιλεγμένης ομάδας στοιχείων ελέγχου του προγράμματος περιήγησης, με μια πηγή που ορίζεται από το χαρακτηριστικό `src`, όπως μπορεί να επαληθευτεί σε [αυτόν τον κώδικα](#), στον οποίο μπορείτε να ακούσετε μερικά πουλιά να κελαηδούν.

# Ήχος στην HTML5

- Ένας άλλος τρόπος μπορεί να επιτευχθεί χρησιμοποιώντας το στοιχείο `<object>`, το οποίο χρησιμοποιείται για την ενσωμάτωση διαφορετικών τύπων αρχείων πολυμέσων. [Αυτό το παράδειγμα](#) ενσωματώνει ένα αρχείο ήχου σε μια ιστοσελίδα ακολουθώντας την προαναφερθείσα μέθοδο. Ωστόσο, το στοιχείο `<object>` δεν υποστηρίζεται ευρέως και εξαρτάται από τον τύπο του αντικειμένου που προστίθεται. Άλλες μέθοδοι όπως το στοιχείο `<audio>` της HTML5 ή οι εξωτερικές συσκευές αναπαραγωγής ήχου HTML5 μπορεί να αποτελούν καλύτερη επιλογή σε πολλές περιπτώσεις.
- Τέλος, το στοιχείο `<embed>` μπορεί επίσης να αποτελέσει τρόπο εισαγωγής μέσων σε ένα έγγραφο HTML, ακολουθώντας [αυτό το παράδειγμα](#). Παρόλο που το στοιχείο `<embed>` υποστηρίζεται σε πολύ μεγάλο βαθμό στα σημερινά προγράμματα περιήγησης και ορίζεται ως το καθιερωμένο στην HTML5, ο ήχος που θα εισαχθεί ενδέχεται να μην αναπαράγεται καθώς μπορεί η μορφή του αρχείου να μην υποστηρίζεται από το πρόγραμμα περιήγησης ή να μην υπάρχει πρόσβαση στις προσθήκες.

## Βίντεο στην HTML5

- Ας μάθουμε τώρα πώς γίνεται η ενσωμάτωση βίντεο σε ένα έγγραφο HTML.
- Όπως και στην περίπτωση του ήχου, τα αρχεία βίντεο ήταν επίσης δύσκολο να εισαχθούν σε μια ιστοσελίδα και για τον ίδιο λόγο (τα προγράμματα περιήγησης ιστού δεν είχαν ένα ενιαίο πρότυπο για τον ορισμό των ενσωματωμένων αρχείων πολυμέσων όπως το βίντεο). Στις επόμενες παραγράφους, θα επεξηγηθούν πολλοί τρόποι εισαγωγής αυτών των αρχείων.





## Βίντεο στην HTML5

Το στοιχείο `<video>`, το οποίο εισήχθη πρόσφατα λειτουργεί στα περισσότερα από τα σύγχρονα προγράμματα περιήγησης. [Αυτό το παράδειγμα](#) εξηγεί πώς μπορείτε να εισαγάγετε ένα βίντεο με απλό τρόπο σε ένα έγγραφο HTML, χρησιμοποιώντας το προεπιλεγμένο σύνολο εντολών του προγράμματος περιήγησης, με μία πηγή που ορίζεται από το χαρακτηριστικό `src`.

Το στοιχείο `<object>` χρησιμοποιείται επίσης για την ενσωμάτωση διαφορετικών τύπων αρχείων πολυμέσων. Ακολουθώντας [αυτό το παράδειγμα](#), είναι εύκολο να καταλάβουμε πως να ενσωματώσουμε ένα βίντεο Flash σε μια ιστοσελίδα (μόνο τα προγράμματα περιήγησης/εφαρμογές που υποστηρίζουν το Flash θα μπορούν να το αναπαράγουν). Πρέπει να σημειωθεί ότι το `<object>` στοιχείο δεν υποστηρίζεται εκτενώς και εξαρτάται πολύ από τον τύπο του ενσωματωμένου αντικειμένου. Άλλες μέθοδοι θα μπορούσαν να αποτελέσουν καλύτερη επιλογή σε πολλές περιπτώσεις, καθώς, συσκευές όπως iPad και iPhone δεν μπορούν να αναπαράγουν βίντεο Flash.

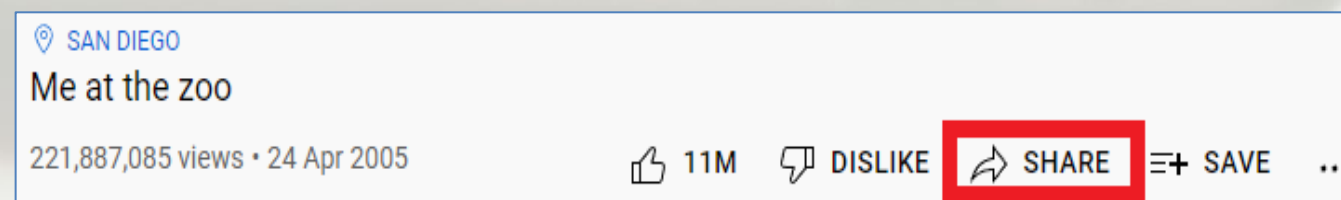


## Βίντεο στην HTML5

Πώς ενσωματώνουμε **βίντεο από το YouTube**; Αυτός είναι ο πιο απλός και συνηθισμένος τρόπος ενσωμάτωσης αρχείων βίντεο σε ιστοσελίδες στις μέρες μας. Ο σχεδιαστής ιστοσελίδων πρέπει απλώς να ανεβάσει το βίντεο στο YouTube και να εισαγάγει έναν κώδικα HTML για να εμφανίσει το βίντεο στην ιστοσελίδα του. Ακολουθεί ένας μικρός οδηγός βήμα προς βήμα:

**Βήμα 1** – Ανεβάστε ένα βίντεο στο YouTube.

**Βήμα 2** – Μετά τη μεταφόρτωση του βίντεο στο YouTube, ο σχεδιαστής ιστού θα πρέπει να αναζητήσει το κουμπί «Κοινοποίηση» (Share), το οποίο βρίσκεται κάτω από το βίντεο στο πρόγραμμα αναπαραγωγής βίντεο της πλατφόρμας, όπως ακριβώς φαίνεται στο παράδειγμα:





## Βίντεο στην HTML5

Όταν κάνετε κλικ στο κουμπί «Κοινοποίηση», θα ανοίξει ένα πλαίσιο για να κοινοποιήσετε το βίντεο που εμφανίζει μερικές ακόμη επιλογές. Τώρα, θα πρέπει να κάνετε κλικ στο κουμπί «**Ενσωμάτωση βίντεο**» (Embed video), το οποίο θα δημιουργήσει τον κώδικα HTML για την απευθείας ενσωμάτωση του βίντεο στην ιστοσελίδα. Για να συμβεί αυτό, ο σχεδιαστής Ιστού θα πρέπει να αντιγράψει και να επικολλήσει αυτόν τον κώδικα στο έγγραφο HTML.

```
Embed Video ×  
  
<iframe width="560" height="315"  
src="https://www.youtube.com/embed/  
jNQXAC9IVRw" title="YouTube video  
player" frameborder="0"  
allow="accelerometer; autoplay;  
clipboard-write; encrypted-media;  
gyroscope; picture-in-picture"  
allowfullscreen></iframe>
```



## Βίντεο στην HTML5

Αυτός ο κώδικας μπορεί να προσαρμοστεί περαιτέρω επιλέγοντας την επιλογή προσαρμογής που δίνεται ακριβώς κάτω από το πλαίσιο εισαγωγής κώδικα ενσωμάτωσης.

Η εισαγωγή ενός βίντεο YouTube σε μια ιστοσελίδα εξηγείται σε [αυτό το παράδειγμα](#).



## Αποθήκευση δεδομένων στον ιστό σε HTML5 (Web Storage)

Αναρωτηθήκατε ποτέ πώς μπορείτε να χρησιμοποιήσετε τη δυνατότητα αποθήκευσης δεδομένων στον ιστό στην HTML5 για την αποθήκευσή τους στο πρόγραμμα περιήγησης του χρήστη; Οι ακόλουθες παράγραφοι θα σας βοηθήσουν να κατανοήσετε πώς γίνεται αυτό.

Πρώτον, είναι σημαντικό να κατανοήσουμε τα θέματα που προκύπτουν με την «αποθήκευση δεδομένων στον ιστό» (web storage).



# Αποθήκευση δεδομένων στον ιστό σε HTML5 (Web Storage)

Με την αποθήκευση δεδομένων στον ιστό , οι διαδικτυακές εφαρμογές μπορούν να αποθηκεύουν δεδομένα τοπικά στο πρόγραμμα περιήγησης του χρήστη. Πριν από την HTML5, τα δεδομένα της εφαρμογής έπρεπε να αποθηκεύονται σε cookies, ενσωματωμένα σε κάθε αίτημα διακομιστή. Η αποθήκευση στον ιστό είναι πιο ασφαλής, ενώ μπορούν να αποθηκευτούν σημαντικές ποσότητες δεδομένων, χωρίς να επηρεάζεται η απόδοση του ιστότοπου.

Οι πληροφορίες που διατηρούνται στο χώρο όπου αποθηκεύονται δεδομένα στον ιστό δεν αποστέλλονται στον διακομιστή ιστού, σε αντίθεση με τα cookies όπου τα δεδομένα παραδίδονται στον διακομιστή με κάθε αίτημα. Επιπλέον, τα cookies επιτρέπουν την αποθήκευση μόνο ενός μικρού όγκου δεδομένων (σχεδόν 4 KB), ενώ ο χώρος αποθήκευσης δεδομένων στον ιστό επιτρέπει έως και 5 MB.



## Αποθήκευση δεδομένων στον ιστό σε HTML5 (Web Storage)

Υπάρχουν δύο τύποι αποθήκευσης δεδομένων στον ιστό:

**Τοπική αποθήκευση περιήγησης (*Local storage*)** — χρησιμοποιεί το αντικείμενο **localStorage** για την αποθήκευση δεδομένων για ολόκληρο τον ιστότοπο σε μια *μόνιμη βάση*. Επομένως, τα αποθηκευμένα τοπικά δεδομένα θα είναι διαθέσιμα την επόμενη ημέρα, την επόμενη εβδομάδα ή το επόμενο έτος, εκτός εάν καταργηθούν.

**Αποθήκευση περιόδου λειτουργίας (*Session Storage*)** — χρησιμοποιεί το αντικείμενο **sessionStorage** για την αποθήκευση δεδομένων σε μια *προσωρινή βάση*, για ένα μόνο παράθυρο ή καρτέλα του προγράμματος περιήγησης. Τα δεδομένα εξαφανίζονται όταν τελειώνει η περίοδος λειτουργίας, για παράδειγμα όταν ο χρήστης κλείνει αυτό το παράθυρο ή την καρτέλα του προγράμματος περιήγησης.



## Αποθήκευση δεδομένων ιστό σε HTML5 (Web Storage)

Όσον αφορά την **τοπική αποθήκευση (local storage)**, το κάθε στοιχείο δεδομένων συλλέγεται σε ένα ζεύγος κλειδιού/τιμής. Το κλειδί προσδιορίζει το όνομα της πληροφορίας (π.χ. «first\_name») και η τιμή είναι η τιμή που σχετίζεται με το ίδιο κλειδί (δηλ. «Peter»). Ο [παρακάτω κώδικας JS](#) εκφράζει τα εξής:

- **ToLocalStorage.setItem** (key, value) αποθηκεύει την τιμή που σχετίζεται με ένα κλειδί.
- **ToLocalStorage.getItem** (key) αποθηκεύει την τιμή που σχετίζεται με το κλειδί.





## Αποθήκευση δεδομένων ιστό σε HTML5 (Web Storage)

Μπορείτε επίσης να αφαιρέσετε ένα συγκεκριμένο στοιχείο από την αποθήκευση, προσθέτοντας το όνομα του κλειδιού στη μέθοδο `removeItem()`, π.χ. `localStorage.removeItem("first_name")`.

Ωστόσο, εάν ο σχεδιαστής ιστοσελίδων θέλει να αφαιρέσει ολόκληρο τον χώρο αποθήκευσης, θα πρέπει να χρησιμοποιήσει τη μέθοδο `clear()`, δηλαδή `localStorage.clear()`. Η μέθοδος `clear()` απλώς διαγράφει όλα τα ζεύγη κλειδιών/τιμών από το `localStorage` ταυτόχρονα, **επομένως πρέπει να χρησιμοποιείται με προσοχή**. Τα δεδομένα που είναι αποθηκευμένα στον ιστό δεν θα είναι προσβάσιμα μεταξύ διαφορετικών προγραμμάτων περιήγησης.

Τέλος, το αντικείμενο `sessionStorage` λειτουργεί με παρόμοιο τρόπο με το `localStorage`, με τη διαφορά ότι αποθηκεύει τα δεδομένα μόνο για μία περίοδο λειτουργίας. Το [παράδειγμα σε αυτό τον σύνδεσμο](#) εξηγεί λεπτομερώς πώς λειτουργεί αυτό.



# Εφαρμογές προσωρινής μνήμης στην HTML5 (Application cache)

Αυτή η ενότητα θα είναι χρήσιμη για να μάθετε τον τρόπο δημιουργίας εφαρμογών εκτός σύνδεσης χρησιμοποιώντας τη **λειτουργία προσωρινής αποθήκευσης στην HTML5**.

Γνωρίζουμε ήδη ότι οι περισσότερες εφαρμογές που παρέχονται μέσω του ιστού δεν λειτουργούν εάν ο σχεδιαστής ιστοσελίδων είναι εκτός σύνδεσης. Ωστόσο, η HTML5 εισήγαγε έναν μηχανισμό προσωρινής αποθήκευσης εφαρμογών που επιτρέπει στο πρόγραμμα περιήγησης να αποθηκεύει αυτόματα το έγγραφο HTML και όλους τους άλλους πόρους που απαιτούνται για να το εμφανίσει σωστά στον τοπικό υπολογιστή, με αυτόν τον τρόπο το πρόγραμμα περιήγησης μπορεί να έχει πρόσβαση στην ιστοσελίδα και τους πόρους της χωρίς σύνδεση στο διαδίκτυο. Αυτό υποστηρίζεται σε όλα τα μεγάλα σύγχρονα προγράμματα περιήγησης ιστού (Firefox, Chrome, Opera, Safari και Internet Explorer 10 και άνω).



# Εφαρμογές προσωρινής μνήμης στην HTML5 (Application cache)

Προκύπτουν πολλά πλεονεκτήματα από τη χρήση αυτής της δυνατότητας:

- **Περιήγηση εκτός σύνδεσης** — Οι επισκέπτες μπορούν να χρησιμοποιήσουν την εφαρμογή ακόμη και όταν δεν είναι συνδεδεμένοι στο διαδίκτυο ή υπάρχουν απροσδόκητες διακοπές στη σύνδεση δικτύου.
- **Βελτίωση απόδοσης** — Οι αποθηκευμένοι πόροι φορτώνονται απευθείας από το μηχάνημα του χρήστη αντί από τον απομακρυσμένο διακομιστή, ώστε οι ιστοσελίδες να φορτώνουν πιο γρήγορα και να έχουν καλύτερη απόδοση.
- **Μείωση αιτημάτων HTTP και φόρτου του διακομιστή** — Το πρόγραμμα περιήγησης πρέπει μόνο να πραγματοποιήσει τη λήψη των ενημερωμένων/τροποποιημένων πόρων από τον απομακρυσμένο διακομιστή, μειώνοντας τα αιτήματα HTTP και εξοικονομώντας πολύτιμο εύρος ζώνης, καθώς μειώνουν επίσης τον φόρτο στον διακομιστή ιστού.



## Εφαρμογές προσωρινής μνήμης στην HTML5 (Application cache)

Υπάρχουν μερικά βήματα που πρέπει να ακολουθήσετε για να αποθηκεύσετε τα αρχεία στην προσωρινή μνήμη για χρήση εκτός σύνδεσης:

**ΒΗΜΑ 1** – Δημιουργήστε ένα αρχείο Cache Manifest. Αυτό είναι ένα ειδικό αρχείο κειμένου που ενημερώνει τα προγράμματα περιήγησης ποια αρχεία πρέπει να αποθηκεύουν (και ποια όχι) και ποια αρχεία πρέπει να αντικαταστήσουν. Ξεκινά πάντα με τις λέξεις **CACHE MANIFEST** (πάντα με κεφαλαία).

# Εφαρμογές προσωρινής μνήμης στην HTML5 (Application cache)

Example	Download
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	

```
1 CACHE MANIFEST
2 # v1.0 : 10-08-2014
3
4 CACHE:
5 # pages
6 index.html
7
8 # styles & scripts
9 css/theme.css
10 js/jquery.min.js
11 js/default.js
12
13 # images
14 /favicon.ico
15 images/logo.png
16
17 NETWORK:
18 login.php
19
20 FALLBACK:
21 / /offline.html
```



# Εφαρμογές προσωρινής μνήμης στην HTML5 (Application cache)

Ας εξηγήσουμε τώρα τον κώδικα που χρησιμοποιήθηκε στην εικόνα της τελευταίας διαφάνειας.

- Πρώτον, είναι σημαντικό να κατανοήσουμε ότι τα αρχεία manifest μπορούν να έχουν τρία διαφορετικά τμήματα: **CACHE** , **NETWORK** και **FALLBACK**.
- Τα αρχεία που παρατίθενται κάτω από την κεφαλίδα του **CACHE:** (ή αμέσως μετά τη γραμμή CACHE MANIFEST) αποθηκεύονται μόνο στην προσωρινή μνήμη μετά την πρώτη τους λήψη.
- Τα αρχεία κάτω από το **NETWORK:** είναι πόροι στη λευκή λίστα που δεν αποθηκεύονται ποτέ στην προσωρινή μνήμη και είναι διαθέσιμοι μόνο στο διαδίκτυο. Σημαίνει ότι οι χρήστες δεν μπορούν ποτέ να έχουν πρόσβαση στη σελίδα **login.php** όταν είναι εκτός σύνδεσης.
- Το **FALLBACK:** καθορίζει εναλλακτικές σελίδες που θα πρέπει να χρησιμοποιεί το πρόγραμμα περιήγησης σε περίπτωση που δεν μπορεί να γίνει η σύνδεση με τον διακομιστή. Κάθε καταχώρηση σε αυτήν την ενότητα παραθέτει δύο URI. Το πρώτο είναι ο κύριος πόρος και το δεύτερο είναι ο εναλλακτικός. Για παράδειγμα, στην περίπτωση της *Εικόνας 98*, η σελίδα **offline.html** θα εμφανιστεί εάν ο χρήστης είναι εκτός σύνδεσης. Επίσης, και οι δύο URI πρέπει να προέρχονται από την ίδια προέλευση με το αρχείο manifest.
- Θα πρέπει να σημειωθεί ότι οι γραμμές που ξεκινούν με το σύμβολο '#' είναι γραμμές με σχόλια.



## Εφαρμογές προσωρινής μνήμης στην HTML5 (Application cache)

**ΒΗΜΑ 2** – Χρησιμοποιήστε το αρχείο cache manifest. Μετά τη δημιουργία του, ο σχεδιαστής ιστού θα πρέπει να ανεβάσει το αρχείο cache manifest στον διακομιστή ιστού. Βεβαιωθείτε ότι ο διακομιστής ιστού είναι ρυθμισμένος έτσι ώστε να υποστηρίζει τα αρχεία manifest με το MIME **type text/cache-manifest**

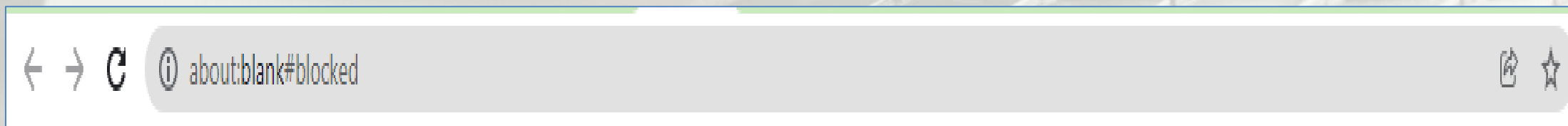
Για να λειτουργήσει το αρχείο cache manifest, ο σχεδιαστής ιστοσελίδων θα πρέπει να το ενεργοποιήσει στις ιστοσελίδες, προσθέτοντας το χαρακτηριστικό manifest στο βασικό στοιχείο, το **< html >**, όπως φαίνεται στην εικόνα στην επόμενη διαφάνεια.



# Εφαρμογές προσωρινής μνήμης στην HTML5 (Application cache)

```
1 <!DOCTYPE html>
2 <html lang="en" manifest="example.appcache">
3 <head>
4   <title>Using the Application Cache</title>
5 </head>
6 <body>
7   <!--The document content will be inserted here-->
8 </body>
9 </html>
```

Εάν ο χρήστης είναι συνδεδεμένος με το διαδίκτυο, το αποτέλεσμα που θα προκύψει από αυτόν τον κώδικα θα είναι το εξής:







# Web Workers στην HTML5

Αυτή η εποενοτητα θα είναι χρησιμη για την εμβάθυνση σε θέματα JS, καθώς θα δείξει στους εκπαιδευόμενους πώς να χρησιμοποιούν τα web workers της HTML5 για την εκτέλεση κώδικα JS. Έτσι, οι εκπαιδευόμενοι μπορούν να ξεναδούν τις οδηγίες εάν αντιμετωπίσουν οποιοσδήποτε δυσκολίες.

Αν κάποιος προσπαθήσει να εκτελέσει εντατικούς, χρονοβόρους και απαιτητικούς υπολογισμούς που απαιτούν εργασίες με χρήση κώδικα JavaScript, πιθανότατα θα παγώσει τις ιστοσελίδες και θα εμποδίσει τους χρήστες από το να κάνουν οτιδήποτε μέχρι να ολοκληρωθεί η εργασία. Γιατί; Λοιπόν, επειδή ο κώδικας JS τρέχει πάντα στο προσκήνιο. Ωστόσο, η HTML5 διαθέτει μια νέα τεχνολογία («web worker») που δημιουργήθηκε για να εκτελεί εργασίες στο παρασκήνιο εκτός από άλλα σενάρια διεπαφής χρήστη, χωρίς να επηρεάζει την απόδοση της σελίδας. Σε αντίθεση με τις κανονικές λειτουργίες JS, το web worker δεν διακόπτει τον χρήστη και η ιστοσελίδα συνεχίζει να λειτουργεί επειδή εκτελεί τις εργασίες στο παρασκήνιο.



# Web Workers στην HTML5

Τα Web workers είναι ιδιαίτερα χρήσιμα για την εκτέλεση μιας χρονοβόρας εργασίας. Έτσι, στο πρώτο παράδειγμα, εκτελείται μια απλή εργασία JS που μετράει από το μηδέν έως το 100 000 (το όνομα του αρχείου θα πρέπει να είναι worker.js), ως εξής:

```
1  var i = 0;
2  function countNumbers() {
3      if(i < 100000) {
4          i = i + 1;
5          postMessage(i);
6      }
7
8      // Wait for sometime before running this script again
9      setTimeout("countNumbers()", 500);
10 }
11 countNumbers();
```

# Web Workers στην HTML5

Έτσι, τώρα που δημιουργήθηκε το αρχείο web worker, ήρθε η ώρα να ξεκινήσουμε το web worker από ένα έγγραφο HTML που εκτελεί τον κώδικα μέσα στο αρχείο με το όνομα «worker.js» στο παρασκήνιο και εμφανίζει σταδιακά το αποτέλεσμα στην ιστοσελίδα. Πρέπει να σημειωθεί ότι ο αριθμός στα δεξιά θα αυξάνεται πάντα μέχρι να φτάσει το 100.000.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <title>Using HTML5 Web Workers</title>
6 <script>
7     if(window.Worker) {
8         // Create a new web worker
9         var worker = new Worker("/examples/js/worker.js");
10
11         // Fire onMessage event handler
12         worker.onmessage = function(event) {
13             document.getElementById("result").innerHTML = event.data;
14         };
15     } else {
16         alert("Sorry, your browser do not support web worker.");
17     }
18 </script>
19 </head>
20 <body>
21     <div id="result">
22         <!--Received messages will be inserted here-->
23     </div>
24 </body>
25 </html>
```

26

## Web Workers στην HTML5

- Τώρα για να εξηγήσουμε τι συμβαίνει στο παράδειγμα της προηγούμενης διαφάνειας, η εντολή **var worker = new Worker("worker.js");** δημιουργεί ένα νέο αντικείμενο web worker, το οποίο χρησιμοποιείται για την επικοινωνία με τον web worker. Όταν ο web worker δημοσιεύει ένα μήνυμα, ενεργοποιεί το πρόγραμμα χειρισμού συμβάντων **onmessage** (γραμμή 14) που επιτρέπει στον κώδικα να λαμβάνει μηνύματα από τον web worker.
- Το στοιχείο **event.data** περιλαμβάνει το μήνυμα που αποστέλλεται από τον web worker. Για την ακρίβεια, ο κώδικας που εκτελεί ένας web worker αποθηκεύεται πάντα σε ένα ξεχωριστό αρχείο JavaScript για να εμποδίσει τον προγραμματιστή ιστού από το να γράψει τον κώδικα του web worker που επιχειρεί να χρησιμοποιήσει παγκόσμιες μεταβλητές ή να ανοίξει απευθείας τα στοιχεία στην ιστοσελίδα.

# Web Workers στην HTML5

Είναι επίσης δυνατό να τερματιστεί ένας τρέχων **web worker** στη μέση της λειτουργίας του, ακολουθώντας το παρακάτω παράδειγμα:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <title>Start/Stop Web Worker in HTML5</title>
6 <script>
7   // Set up global variable
8   var worker;
9
10  function startWorker() {
11    // Initialize web worker
12    worker = new Worker("/examples/js/worker.js");
13
14    // Run update function, when we get a message from worker
15    worker.onmessage = update;
16
17    // Tell worker to get started
18    worker.postMessage("start");
19  }
20
21  function update(event) {
22    // Update the page with current message from worker
23    document.getElementById("result").innerHTML = event.data;
24  }
25
26  function stopWorker() {
27    // Stop the worker
28    worker.terminate();
29  }
30 </script>
31 </head>
```

### Web Worker Demo

# Web Workers στην HTML5

Είναι επίσης δυνατό να τερματιστεί ένας τρέχων **web worker** στη μέση της λειτουργίας, ακολουθώντας το παρακάτω παράδειγμα:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <title>Start/Stop Web Worker in HTML5</title>
6 <script>
7   // Set up global variable
8   var worker;
9
10  function startWorker() {
11    // Initialize web worker
12    worker = new Worker("/examples/js/worker.js");
13
14    // Run update function, when we get a message from worker
15    worker.onmessage = update;
16
17    // Tell worker to get started
18    worker.postMessage("start");
19  }
20
21  function update(event) {
22    // Update the page with current message from worker
23    document.getElementById("result").innerHTML = event.data;
24  }
25
26  function stopWorker() {
27    // Stop the worker
28    worker.terminate();
29  }
30 </script>
31 </head>
```

### Web Worker Demo



## Αποστολή συμβάντων από τον διακομιστή στην HTML5 (Server-Sent Events)

Αυτή η υποενότητα θα είναι χρήσιμη για την κατανόηση του τρόπου χρήσης της δυνατότητας αποστολής συμβάντων από τον διακομιστή (SSE) στην HTML5 για τη δημιουργία μιας μονοκατευθυντικής και μόνιμης σύνδεσης μεταξύ μιας ιστοσελίδας και ενός διακομιστή.



# Αποστολή συμβάντων από τον διακομιστή στην HTML5 (Server-Sent Events)

- Η SSE στην HTML5 είναι ένας καινοτόμος τρόπος επικοινωνίας των ιστοσελίδων με έναν διακομιστή ιστού. Ωστόσο, υπάρχουν ορισμένες περιπτώσεις όπου οι ιστοσελίδες χρειάζονται μια σύνδεση μεγαλύτερης διάρκειας με τον διακομιστή ιστού, για παράδειγμα, σε τιμές μετοχών σε ιστότοπους με χρηματοοικονομικά θέματα όπου οι τιμές ενημερώνονται αυτόματα ή ο χρόνος παιχνιδιών σε διάφορους αθλητικούς ιστότοπους.
- Αυτά γίνονται εφικτά με την SSE στην HTML5, καθώς δίνει τη δυνατότητα σε μια ιστοσελίδα να διατηρεί ανοιχτή σύνδεση με έναν διακομιστή ιστού, με τρόπο που ο διακομιστής Ιστού μπορεί να στείλει μια νέα απάντηση μηχανικά ανά πάσα στιγμή. Σε αυτό το σημείο, δεν χρειάζεται κάθε να επανασυνδέεστε κάθε φορά και να εκτελείτε το ίδιο σενάριο διακομιστή από την αρχή.
- Για να κατανοήσετε καλύτερα τις προαναφερθέντες έννοιες, ανοίξτε ένα αρχείο PHP με το όνομα «[server\\_time.php](#)» και πληκτρολογήστε το ακόλουθο σενάριο σε αυτό. Αυτό το αρχείο θα αναφέρει απλώς την τρέχουσα ώρα του ενσωματωμένου ρολογιού του διακομιστή Ιστού σε τακτά χρονικά διαστήματα:



# Αποστολή συμβάντων από τον διακομιστή στην HTML5 (Server-Sent Events)

```
1  <?php
2  header("Content-Type: text/event-stream");
3  header("Cache-Control: no-cache");
4
5  // Get the current time on server
6  $currentTime = date("h:i:s", time());
7
8  // Send it in a message
9  echo "data: " . $currentTime . "\n\n";
10 flush();
11 ?>
```

Ένα αρχείο PHP είναι ένα αρχείο απλού κειμένου που περιέχει κώδικα γραμμένο στη γλώσσα προγραμματισμού PHP. Δεδομένου ότι η PHP είναι μια γλώσσα προγραμματισμού από την πλευρά του διακομιστή (back-end), ο κώδικας που είναι γραμμένος σε αυτήν εκτελείται στον διακομιστή. Στην πραγματικότητα, ένα αρχείο PHP μπορεί να περιέχει απλό κείμενο, ετικέτες HTML ή κώδικα σύμφωνα με τη σύνταξη της PHP. Η PHP χρησιμοποιείται συνήθως για την ανάπτυξη διαδικτυακών εφαρμογών που υποβάλλονται σε επεξεργασία από μια μηχανή PHP στον διακομιστή Ιστού.



# Αποστολή συμβάντων από τον διακομιστή στην HTML5 (Server-Sent Events)

- Οι δύο πρώτες γραμμές του σεναρίου του PHP ορίζουν δύο πολύ σημαντικές κεφαλίδες. Η πρώτη γραμμή, ορίζει τον τύπο MIME σε ροή κειμένου/συμβάντος (**text/event-stream**), που απαιτείται από το πρότυπο SSE. Η δεύτερη γραμμή ενημερώνει τον διακομιστή Ιστού να απενεργοποιήσει την προσωρινή μνήμη, διαφορετικά το σενάριο που θα εξαχθεί μπορεί να αποθηκευτεί προσωρινά.
- Κάθε μήνυμα που αποστέλλεται μέσω SSE στην HTML5 πρέπει να ξεκινά με το στοιχείο **data:** ακολουθούμενο από το πραγματικό κείμενο του μηνύματος και τη νέα ακολουθία χαρακτήρων σε νέα γραμμή (**\n\n**).
- Και τέλος, η συνάρτηση **flush()** στο PHP έχει χρησιμοποιηθεί για να διασφαλιστεί ότι τα δεδομένα αποστέλλονται αμέσως, αντί να αποθηκεύονται στην προσωρινή μνήμη μέχρι να ολοκληρωθεί ο κώδικας PHP.
- Τώρα όσον αφορά **στον τρόπο επεξεργασίας μηνυμάτων σε μια ιστοσελίδα**, το αντικείμενο **EventSource** χρησιμοποιείται για τη λήψη μηνυμάτων από τα συμβάντα που αποστέλλονται από τον διακομιστή. Στο παράδειγμα που ακολουθεί, οι εκπαιδευόμενοι θα δουν πώς ένα έγγραφο HTML λαμβάνει απλώς την τρέχουσα ώρα που αναφέρεται από τον διακομιστή ιστού και την εμφανίζει στους επισκέπτες της ιστοσελίδας. Για καλύτερη κατανόηση, θα δημιουργηθεί ένα έγγραφο HTML με το όνομα «**demo\_sse.html**» και στη συνέχεια θα τοποθετηθεί στον ίδιο κατάλογο του έργου όπου βρίσκεται το «**server\_time.php**».

# Αποστολή συμβάντων από τον διακομιστή στην HTML5 (Server-Sent Events)

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <title>Using Server-Sent Events</title>
5 <script>
6     window.onload = function() {
7         var source = new EventSource("server_time.php");
8         source.onmessage = function(event) {
9             document.getElementById("result").innerHTML += "New time received
10            from web server: " + event.data + "<br>";
11        };
12    };
13 </script>
14 </head>
15 <body>
16     <div id="result">
17         <!--Server response will be inserted here-->
18     </div>
19 </body>
20 </html>
```



# Γεωτοποθεσία και γεωεντοπισμός μέσω HTML5 (Geolocation)

- Μέσα από αυτή την υποενότητα, οι εκπαιδευόμενοι θα λάβουν μερικές πληροφορίες σχετικά με τον τρόπο χρήσης της δυνατότητας Γεωεντοπισμού μέσω HTML5 για τον εντοπισμό της τοποθεσίας του χρήστη. Αυτή η δυνατότητα επιτρέπει στον προγραμματιστή να ανακαλύψει τις γεωγραφικές συντεταγμένες (γεωγραφικό πλάτος και μήκος) της τρέχουσας τοποθεσίας του επισκέπτη του ιστότοπου. Είναι ιδιαίτερα χρήσιμο για την παροχή της καλύτερης εμπειρίας περιήγησης στον επισκέπτη, καθώς, για παράδειγμα, αυτό το εργαλείο μπορεί να εμφανίζει αποτελέσματα αναζήτησης που βρίσκονται κοντά στην τοποθεσία του χρήστη.
- Η λήψη των πληροφοριών της τοποθεσίας του επισκέπτη του ιστότοπου χρησιμοποιώντας το API γεωγραφικής τοποθεσίας HTML5 δεν είναι δύσκολη. Εκμεταλλεύεται τις τρεις μεθόδους εντός του αντικειμένου `navigator.geolocation` — `getCurrentPosition ()`, `watchPosition ()` και `clearWatch()`.
- Αφού ο χρήστης συμφωνήσει να αφήσει το πρόγραμμα περιήγησης να ενημερώσει τον διακομιστή ιστού σχετικά με την τοποθεσία του (τα προγράμματα περιήγησης δεν θα κοινοποιήσουν την τοποθεσία επισκέπτη με μια ιστοσελίδα εκτός εάν ο χρήστης συμφωνήσει), η διαδικασία γεωεντοπισμού θα πρέπει να πραγματοποιηθεί όπως φαίνεται στην εικόνα στην επόμενη διαφάνεια.

# Γεωτοποθεσία και γεωεντοπισμός μέσω HTML5 (Geolocation)

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Get Visitor's Location Using HTML5 Geolocation</title>
<script>
  function showPosition() {
    if(navigator.geolocation) {
      navigator.geolocation.getCurrentPosition(function(position) {
        var positionInfo = "Your current position is (" + "Latitude: " +
position.coords.latitude + ", " + "Longitude: " + position.coords.longitude + ")";
        document.getElementById("result").innerHTML = positionInfo;
      });
    } else {
      alert("Sorry, your browser does not support HTML5 geolocation.");
    }
  }
</script>
</head>
<body>
  <div id="result">
    <!--Position information will be inserted here-->
  </div>
  <button type="button" onclick="showPosition();">Show Position</button>
</body>
</html>
```

Your current position is (Latitude: 41.0079509, Longitude: -8.6270736)

Show Position



## Γεωτοποθεσία και γεωεντοπισμός μέσω HTML5 (Geolocation)

- Σε περίπτωση που ένας χρήστης δεν θέλει να μοιραστεί τα δεδομένα της τοποθεσίας του με τον ιστότοπο, ο προγραμματιστής μπορεί να παρέχει δύο δευτερεύουσες λειτουργίες όταν θέτει τη **λειτουργία `getCurrentLocation()`**. Η πρώτη λειτουργία χρησιμοποιείται σε περίπτωση που η προσπάθεια γεωεντοπισμού είναι επιτυχής, ενώ η δεύτερη χρησιμοποιείται εάν η προσπάθεια γεωεντοπισμού αποτύχει.

# Γεωτοποθεσία και γεωεντοπισμός μέσω HTML5 (Geolocation)

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Handling the Geolocation Errors and Rejections</title>
<script>
  // Set up global variable
  var result;

  function showPosition() {
    // Store the element where the page displays the result
    result = document.getElementById("result");

    // If geolocation is available, try to get the visitor's position
    if(navigator.geolocation) {
      navigator.geolocation.getCurrentPosition(successCallback, errorCallback);
      result.innerHTML = "Getting the position information...";
    } else {
      alert("Sorry, your browser does not support HTML5 geolocation.");
    }
  };

  // Define callback function for successful attempt
  function successCallback(position) {
    result.innerHTML = "Your current position is (" + "Latitude: " + position.coords.latitude + ", " + "Longitude: " + position.coords.longitude + ")";
  }

  // Define callback function for failed attempt
  function errorCallback(error) {
    if(error.code == 1) {
      result.innerHTML = "You've decided not to share your position, but it's OK. We won't ask you again.";
    } else if(error.code == 2) {
      result.innerHTML = "The network is down or the positioning service can't be reached.";
    } else if(error.code == 3) {
      result.innerHTML = "The attempt timed out before it could get the location data.";
    } else {
      result.innerHTML = "Geolocation failed due to unknown error.";
    }
  }
</script>
</head>
<body>
  <div id="result">
    <!--Position information will be inserted here-->
  </div>
  <button type="button" onclick="showPosition();">Show Position</button>
</body>
</html>
```



## Γεωτοποθεσία και γεωεντοπισμός μέσω HTML5 (Geolocation)

- Υπάρχουν πολλές ενδιαφέρουσες λειτουργίες που μπορούν να εξρευνηθούν με τα δεδομένα γεωεντοπισμού, όπως η εμφάνιση της τοποθεσίας του χρήστη στους Χάρτες Google. Με βάση τα δεδομένα γεωγραφικού πλάτους και μήκους που ανακτώνται μέσω της δυνατότητας γεωεντοπισμού στην HTML5, [αυτό το παράδειγμα](#) δείχνει την τρέχουσα τοποθεσία του χρήστη. Αυτό δείχνει απλώς μια στατική εικόνα που δείχνει την τοποθεσία του χρήστη, αν και ένας διαδραστικός χάρτης Google με δυνατότητα μεταφοράς, μεγέθυνσης/σμίκρυνσης και άλλων λειτουργιών, όπως [δείχνει αυτό το παράδειγμα](#).
- Όλα τα προαναφερθέντα παραδείγματα έχουν βασιστεί στη λειτουργία `getCurrentPosition()`. Ωστόσο, η συνάρτηση γεωεντοπισμού έχει μια άλλη τεχνική, την `watchPosition ()` που επιτρέπει την παρακολούθηση της κίνησης του επισκέπτη εμφανίζοντας την ενημερωμένη θέση καθώς αλλάζει η τοποθεσία. Η `watchPosition ()` έχει τις ίδιες παραμέτρους εισόδου με την `getCurrentPosition()`. Ωστόσο, η `watchPosition()` μπορεί να ενεργοποιήσει την επιτυχή λειτουργία πολλές φορές, όταν λαμβάνει την τοποθεσία για πρώτη φορά και ξανά, κάθε φορά που εντοπίζει μια νέα θέση, όπως [φαίνεται σε αυτό το παράδειγμα](#).



## Σύρσιμο και απόθεση στην HTML5 (Drag and Drop)

- Το σύρσιμο και η απόθεση ενός στοιχείου σε μια άλλη τοποθεσία σε έναν ιστότοπο αποτελεί μια κοινή διαδικασία στην καθημερινή ρουτίνα του Διαδικτύου.
- Η δυνατότητα για σύρσιμο και απόθεση στην HTML5 επιτρέπει αυτή τη διαδικασία, σύροντας και αποθέτοντας.
- Αυτό το απλό [παράδειγμα](#) μεταφοράς και απόθεσης δείχνει στους εκπαιδευόμενους τον τρόπο με τον οποίο μπορούν να προσπαθήσουν να εξοικειωθούν περισσότερο με αυτήν την έννοια. Παρόλο που ο κώδικας φαίνεται να είναι δυσνόητος, είναι αρκετά απλός και λογικός:

## Σύρσιμο και απόθεση στην HTML5 (Drag and Drop)

- Πρώτον, για να έχει ένα στοιχείο τη δυνατότητα μεταφοράς, το χαρακτηριστικό **draggable** πρέπει να είναι **true**:

```
<img draggable= "true">
```

- Στη συνέχεια, θα πρέπει να καθοριστεί τι θα συμβεί μετά τη μεταφορά του στοιχείου. Στο παράδειγμα που δόθηκε παραπάνω, το χαρακτηριστικό **ondragstart** θέτει μια λειτουργία (**drag (event)**) που καθορίζει ποια δεδομένα θα συρθούν. Η διαδικασία **dataTransfer.setData()** ορίζει τον τύπο δεδομένων και την τιμή των δεδομένων που έχουν συρθεί:

```
Functiondrag (ev){ev.dataTransfer.setData("text",ev.target.id);}
```

- Σε αυτό το παράδειγμα, ο τύπος δεδομένων είναι το «text», που είναι η τιμή του αναγνωριστικού του στοιχείου με δυνατότητα μεταφοράς («drag1»).



## Σύρσιμο και απόθεση στην HTML5 (Drag and Drop)

- Το συμβάν **ondragover** ορίζει πού μπορούν να τοποθετηθούν τα δεδομένα που έχουν συρθεί. Ωστόσο, δεν είναι δυνατή η απόθεση δεδομένων/στοιχείων σε άλλα στοιχεία, από προεπιλογή. Για να επιτραπεί μια απόθεση, ο σχεδιαστής ιστοτόπου θα πρέπει να αποτρέψει τον προεπιλεγμένο χειρισμό του στοιχείου, με τη λειτουργία **event.preventDefault()** για το συμβάν **ondragover**:

*event.preventDefault ()*

- Με την απόθεση των δεδομένων που σύρθηκαν, συμβαίνει ένα συμβάν απόθεσης. Στο προηγούμενο παράδειγμα, το χαρακτηριστικό **ondrop** χρησιμοποίησε μια λειτουργία **drop(event)**:

```
function drop(ev) {  
    ev.preventDefault();  
    var data=ev.dataTransfer.getData("text");  
    ev.target.appendChild(document.getElementById(data));  
}
```



## Σύρσιμο και απόθεση στην HTML5 (Drag and Drop)

- ✓ Η λειτουργία `preventDefault()` αποτρέπει τον προεπιλεγμένο χειρισμό των δεδομένων από το πρόγραμμα περιήγησης (η προεπιλογή είναι ανοιχτή ως σύνδεσμος κατά την απόθεση).
- ✓ Ο προγραμματιστής λαμβάνει τα δεδομένα που έχουν συρθεί με την τεχνική `dataTransfer.getData()`. Αυτή η τεχνική θα εμφανίσει τυχόν δεδομένα που ορίστηκαν στον ίδιο τύπο στην τεχνική `setData()`.
- ✓ Τα δεδομένα που σύρθηκαν είναι το αναγνωριστικό του στοιχείου που σύρθηκε («`drag1`»)
- ✓ Ο προγραμματιστής θα πρέπει επίσης να προσαρτήσει το στοιχείο που έχει συρθεί στο στοιχείο απόθεσης.



ΑΣ ΕΞΑΣΚΗΘΟΥΜΕ!

Ανοίξτε τον ακόλουθο σύνδεσμο για να εφαρμόσετε τις έννοιες που μάθατε μέχρι στιγμής:

- <https://www.etutorialspoint.com/index.php/exercise/html-practice-exercises-with-solutions>



**ΕΥΧΑΡΙΣΤΟΥΜΕ!**



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