



# JavaScript Tutorial

## Α μέρος

Καπετανάκη Λένα

site: <https://lenakapetanaki.gr>

YouTube: [youtube.com/c/lenakapetanaki](https://youtube.com/c/lenakapetanaki)

# Προαπαιτούμενα

- ▶ Γνώση HTML 5
- ▶ Βασικές γνώσεις CSS 3

# Απαραίτητο λογισμικό

- ▶ Μπορείς να γράψεις javascript σε έναν απλό text editor (όπως το Notepad++ )
- ▶ Τα αρχεία πρέπει να αποθηκεύονται με κατάληξη .js
- ▶ Development Tools για javascript
  - Brackets
  - Visual Studio Code
  - Sublime
  - Dreamweaver (με συνδρομή)

# Javascript

- ▶ Δημιουργήθηκε από τον Brendan Eich
- ▶ Πρώτα ονόματα Mocha, LiveScript



# Τι είναι javascript?

- ▶ Γλώσσα προγραμματισμού
  - Ελέγχει φόρμες με εισαγωγή στοιχείων χρηστών
  - Κάνει αλλαγές σε σελίδες HTML
  - Επεξεργάζεται και αποθηκεύει δεδομένα
  - Μπορεί να ελέγχει εφαρμογές.
- ▶ Χρησιμοποιείται σε συνδυασμό με την html
- ▶ Προσδίδει διαδραστικότητα και δυναμικότητα σε μια html σελίδα

# Που γράφουμε την javascript?

- ▶ Εσωτερικά σε html αρχεία
  - Μέσα στο tag `<script>..... javascript..... </script>`
  - Το tag `<script>` μπορεί να μπει οπουδήποτε στην σελίδα της html είτε μέσα στο `<head>` είτε στο `<body>`
- ▶ Σε εξωτερικό αρχείο

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Untitled Document</title>
</head>
<body>
  <p> hello world! </p>
  <script>
    document.write("<h1> my first script </h1>");
  </script>
</body>
</html>
```

# Εξωτερικό αρχείο

- ▶ Δημιουργία νέου αρχείου με όνομα script.js
  - Στο νέο αρχείο γράφουμε κατευθείαν javascript

```
document.write("<h1> hello from the javascript sheet</h1>");
```

- ▶ Ένωση του .js αρχείου με το .html αρχείο με ένα script

```
<script src="script.js"> </script>
```



# Δυνατότητες εμφάνισης κειμένου μέσω JS

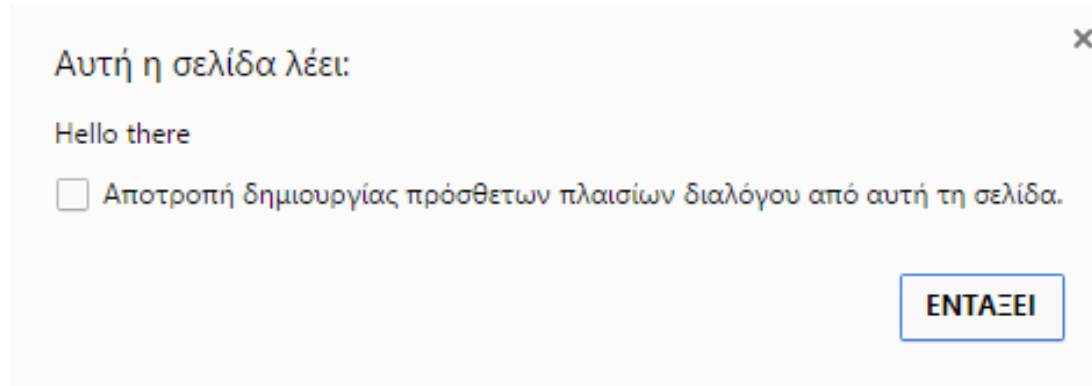
- ▶ Σε ένα pop-up box , χρησιμοποιώντας το **window.alert ()**
- ▶ Σε ένα στοιχείο της HTML, χρησιμοποιώντας το **innerHTML**
- ▶ Στην HTML χρησιμοποιώντας **document.write ()**
- ▶ Στην κονσόλα του προγράμματος περιήγησης, χρησιμοποιώντας το **console.log ()**

# Javascript pop-up windows

## ► Alert Box

- Εμφανίζει κάποιο μήνυμα στην οθόνη. Ο χρήστης πρέπει να πατήσει OK για να συνεχίσει

```
window.alert("Hello there");
```

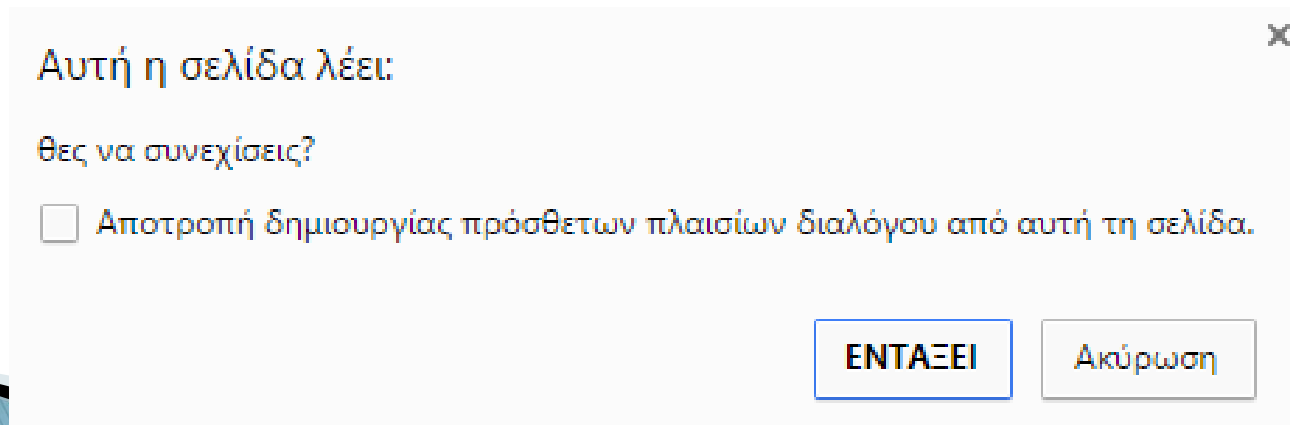


# Javascript pop-up Windows

## ► Confirm Box

- Χρησιμοποιείται για την επιβεβαίωση μιας επιλογής του χρήστη με ok ή cancel

```
window.confirm("Θες να συνεχίσεις?");
```

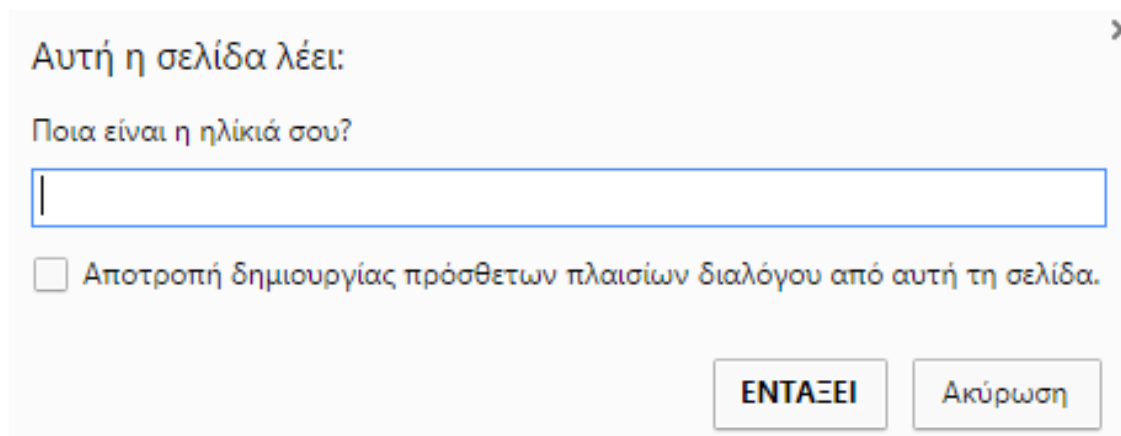


# Javascript pop-up Windows

## ► Prompt Box

- Χρησιμοποιείται προκειμένου να μπορέσει ο χρήστης να εισάγει μια τιμή ένα ένα κείμενο. Ο χρήστης έχει επιλογή OK ή Ακύρωση

```
window.prompt("Ποια είναι η ηλικιά σου?");
```



A screenshot of a JavaScript prompt box. The title bar says "Αυτή η σελίδα λέει:" (This page says:). The main text asks "Ποια είναι η ηλικιά σου?" (What is your age?). Below the text is a single-line text input field. At the bottom, there is a checkbox labeled "Αποτροπή δημιουργίας πρόσθετων πλαισίων διαλόγου από αυτή τη σελίδα." (Prevent additional dialog boxes from this page.). To the right of the checkbox are two buttons: "ΕΝΤΑΞΕΙ" (OK) and "Ακύρωση" (Cancel).

# Πρόσβαση σε στοιχεία της html

- ▶ Για να αποκτηθεί η πρόσβαση σε ένα στοιχείο HTML μέσα από την JavaScript μπορεί να χρησιμοποιηθεί η μέθοδος
  - `document.getElementById ("ονομα_id")`
  - `document.getElementsByClassName()`
  - `document.getElementsByName()`
  - `document.getElementsByTagName()`

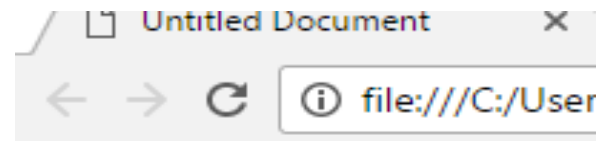
# document.write()

- ▶ Γράφεις μέσα στο αρχείο της html
- ▶ Χρησιμοποιείται για testing κυρίως σκοπούς
- ▶ Χρησιμοποιώντας το document.write () μετά την πλήρη φόρτωση ενός εγγράφου HTML, θα διαγραφούν όλα τα στοιχεία της HTML και θα εμφανιστούν μόνο τα δεδομένα του document.write()

# Document.write(...)

- ▶ Γράφει ένα μήνυμα στην html
- ▶ Χρησιμοποιείται επί το πλείστον για testing λόγους

```
<script>  
document.write("hello world");  
</script>
```



hello world!

**my first script**

# document.getElementById

- ▶ Για να αποκτηθεί η πρόσβαση σε ένα στοιχείο HTML μέσα από την JavaScript μπορεί να χρησιμοποιηθεί η μέθοδος
- ▶ **document.getElementById** (“ονομα\_id”).



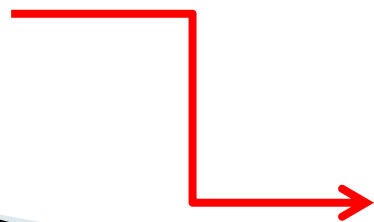
# innerHTML

- ▶ Ένας κοινός τρόπος προβολής δεδομένων σε HTML στοιχεία μέσα από την JS.

```
<p>My First Paragraph.</p>
```

```
<p id="demo"></p>
```

```
<script>  
document.getElementById("demo").innerHTML = 5 + 6;  
</script>
```



My First Paragraph.  
11

# console.log()

- ▶ Χρησιμοποιείται για debugging σκοπούς
- ▶ Εμφανίζει το αποτέλεσμα στο Console του Inspector

# JavaScript Σχόλια

- ▶ Ο κωδικας μετά από διπλές γραμμές `//` ή μεταξύ `/*` και `*/` αντιμετωπίζεται ως σχόλιο.
- ▶ Τα σχόλια αγνοούνται και δεν θα εκτελεστούν

# Javascript Statements

- ▶ Οι προτάσεις στην JS αποτελούνται από Μεταβλητές , Operators, Εκφράσεις, Λέξεις-κλειδιά και Σχόλια.
- ▶ Οι προτάσεις χωρίζονται πάντα με ελληνικό ερωτηματικό (;)

# Javascript Variables

- ▶ Σε μια γλώσσα προγραμματισμού, οι μεταβλητές χρησιμοποιούνται για την αποθήκευση δεδομένων.
- ▶ Η JavaScript χρησιμοποιεί τη λέξη-κλειδί **var** για να δηλώσει τις μεταβλητές.
- ▶ Χρησιμοποιείται το “=” για την εκχώρηση τιμών σε μεταβλητές
- ▶ Σε αυτό το παράδειγμα, το x ορίζεται ως μεταβλητή. Στη συνέχεια στο x εκχωρείται(δίνεται) η τιμή 13

```
var x;  
x = 13;
```

- ▶ Τα ονόματα μπορούν να περιέχουν γράμματα, ψηφία, υπογράμμιση και δολάριο.
- ▶ Τα ονόματα πρέπει να αρχίζουν με ένα γράμμα
- ▶ Τα ονόματα είναι ευαίσθητα σε πεζά (γ και Υ είναι διαφορετικές μεταβλητές)
- ▶ Οι δεσμευμένες λέξεις (όπως λέξεις-κλειδιά JavaScript) δεν μπορούν να χρησιμοποιηθούν ως ονόματα

# Global Scope Variables

- ▶ Οι μεταβλητές που δηλώνονται Globally (εξω από κάθε function) έχουν Global Scope. Οι Global Variables είναι προσβάσιμες από οπουδήποτε σε ένα πρόγραμμα JavaScript.

```
var carName = "Volvo";
```

```
// code here can use carName
```

```
function myFunction() {  
    // code here can also use carName  
}
```

# Function Scope Variables

- ▶ Οι μεταβλητές που δηλώνονται τοπικά (εντός μιας συνάρτησης) έχουν Function Scope.
- ▶ Οι τοπικές μεταβλητές είναι προσπελάσιμες μόνο μέσα από τη Function όπου δηλώνονται.

```
// code here can NOT use carName
```

```
function myFunction() {  
    var carName = "Volvo";  
    // code here CAN use carName  
}
```

```
// code here can NOT use carName
```



# let

- ▶ Η εντολή let δηλώνει μια τοπική μεταβλητή block scope
- ▶ Σε αντίθεση η εντολή var δεν έχει block scope

```
var x = 10;  
// Here x is 10  
{  
  var x = 2;  
  // Here x is 2  
}  
// Here x is 2
```

```
let x = 10;  
// Here x is 10  
{  
  let x = 2;  
  // Here x is 2  
}  
// Here x is 10
```

# const

- ▶ Οι μεταβλητές που ορίζονται με `const` χρησιμοποιούνται όπως αυτές με `let` με την διαφορά ότι δεν μπορούν να εκχωρηθούν εκ νέου
- ▶ ΔΕΝ ορίζει μια σταθερή τιμή. Ορίζει μια σταθερή αναφορά σε μια τιμή.

```
const PI = 3.141592653589793;  
PI = 3.14;           // This will give an error  
PI = PI + 10;        // This will also give an error
```

# JS arithmetic operators

| Operator | Description    |
|----------|----------------|
| +        | Addition       |
| -        | Subtraction    |
| *        | Multiplication |
| /        | Division       |
| %        | Modulus        |
| ++       | Increment      |
| --       | Decrement      |

# JavaScript Assignment Operators

| Operator | Example  | Same As     |
|----------|----------|-------------|
| =        | x = y    | x = y       |
| +=       | x += y   | x = x + y   |
| -=       | x -= y   | x = x - y   |
| *=       | x *= y   | x = x * y   |
| /=       | x /= y   | x = x / y   |
| %=       | x %= y   | x = x % y   |
| <<=      | x <<= y  | x = x << y  |
| >>=      | x >>= y  | x = x >> y  |
| >>>=     | x >>>= y | x = x >>> y |
| &=       | x &= y   | x = x & y   |
| ^=       | x ^= y   | x = x ^ y   |
| =        | x  = y   | x = x   y   |
| **=      | x **= y  | x = x ** y  |

# JavaScript string operators

- ▶ Μπορείς να προσθέσεις και κείμενο με το + operator.

```
<p id="demo"></p>
```

```
<script>  
var txt1 = "Γεια σου!";  
var txt2 = "Τι κάνεις??";  
document.getElementById("demo").innerHTML = txt1 + " " + txt2;  
</script>
```

# JavaScript Data Types

## ▶ Περιέχουν τιμές

- `string`
- `number`
- `boolean`
- `object`
- `function`

## ▶ Δεν περιέχουν τιμές

- `null`
- `undefined`

# Functions

- ▶ Μια συνάρτηση JavaScript είναι ένα τμήμα κώδικα JavaScript, που δεν εκτελείται από την αρχή αλλά μόνο όταν "καλείται".
- ▶ Για παράδειγμα, μια συνάρτηση μπορεί να καλείται όταν συμβαίνει ένα event , όπως όταν ο χρήστης κάνει κλικ σε ένα κουμπί.

```
function name(parameter1, parameter2, parameter3) {  
    code to be executed  
}
```

- ▶ Όταν η JavaScript φτάσει σε εντολή **return**, η function θα σταματήσει να εκτελείται.

```
var x = myFunction(4, 3);  
  
function myFunction(a, b)  
  return a * b;  
}
```

The result in x will be:

12



# Events (1)

- ▶ Ένα συμβάν HTML μπορεί να είναι κάτι που κάνει ο browser ή κάτι που κάνει ένας χρήστης.
- ▶ Ακολουθούν ορισμένα παραδείγματα συμβάντων HTML:
  - Έχει ολοκληρωθεί η φόρτωση μιας ιστοσελίδας HTML
  - Ένα πεδίο *input* HTML έχει αλλάξει
  - Έγινε κλικ σε ένα κουμπί HTML
- ▶ Η JavaScript επιτρέπει να εκτελέσετε κώδικα όταν ανιχνεύονται συμβάντα.
- ▶ Η HTML επιτρέπει την προσθήκη events της JS σαν attributes της HTML.

# Κάποια Events (2)

- ▶ **Onload:** η js τρέχει αφού μια εικόνα ή σελίδα φορτώσει.
- ▶ **OnClick:** η js τρέχει όταν ο χρήστης κάνει κλικ με το ποντίκι
- ▶ **Onmouseover:** η js τρέχει όταν ο χρήστης περάσει το ποντίκι πάνω από ένα στοιχείο
- ▶ **Onchange:** η js τρέχει όταν ένα στοιχείο html αλλάζει
- ▶ **Onmouseout:** η js τρέχει όταν το ποντίκι απομακρίνεται από ένα στοιχείο
- ▶ **Onmouseup:** A mouse button is released
- ▶ **Onkeydown**
- ▶ **Onsubmit:** η js τρέχει όταν ο χρήστης υποβάλλει μια φόρμα

[https://www.w3schools.com/jsref/dom\\_obj\\_event.asp](https://www.w3schools.com/jsref/dom_obj_event.asp)

# Events(3)

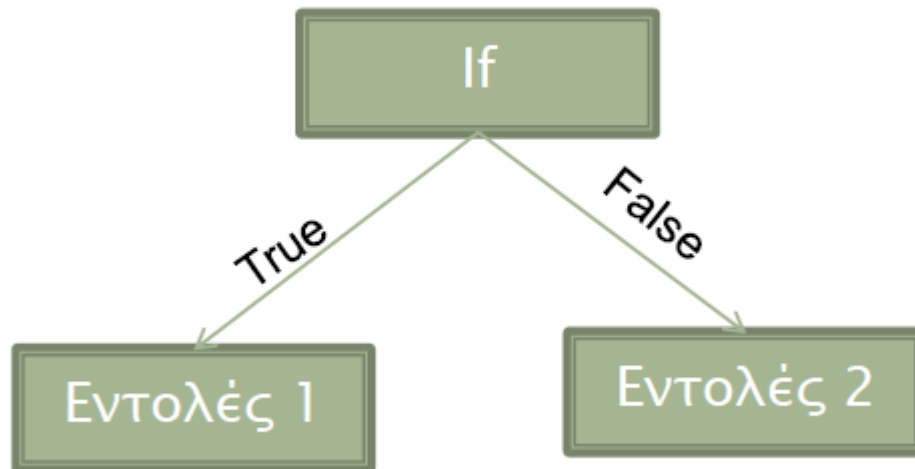
- ▶ Θα αλλάξει το περιεχόμενα της παραγράφου demo στην τρέχουσα ημερομηνία

```
<button  
onclick="document.getElementById('demo').innerHTML=  
Date()">The time is?</button>
```

```
<p id="demo"></p>
```

# IF statement

- ▶ Με την IF statement μπορούμε να πάρουμε μια απόφαση ανάλογα με την τελική τιμή που θα πάρει η δήλωση μας:



```
if (userName == "Bob")  
    alert("Welcome Bob")
```

Ο κώδικας μας ελέγχει αν το userName είναι Bob. Εφόσον είναι Bob τότε εμφανίζει το μήνυμα «Welcome Bob» σε μορφή alert.

Αν το userName είναι διαφορετικό τότε δεν κάνει τίποτε

# If ... else

Ο κώδικας μας ελέγχει αν το `userName` είναι Bob. Εφόσον είναι Bob τότε εμφανίζει το μήνυμα «Welcome Bob» σε μορφή `alert`.

Αν το `userName` είναι διαφορετικό τότε εμφανίζει το “I don’t know you...”

```
if(userName=="Bob"){  
    alert("Welcome Bob");  
} else {  
    alert("I don't know you....");  
}
```

# If.. Else if ... else

```
if(userName=="Bob"){  
    alert("Welcome Bob");  
} else if(userName=="Mpampis") {  
    alert("oh, hello Babis");  
} else{  
    alert("who are you???")  
}
```

Ο κώδικας μας ελέγχει αν το userName είναι Bob. Εφόσον είναι Bob τότε εμφανίζει το μήνυμα «Welcome Bob» σε μορφή alert.

Αν το userName είναι Mpampis εμφανίζει «oh, hello Babis»

Αν το userName είναι διαφορετικό τότε εμφανίζει το "I don't know you..."

$X==Y$  //ίσα?

$!X$  //άρνηση

$X!=Y$  //ανισότητα

$X<Y$  //μικρότερο από

$X<=Y$  //μικρότερο ή ίσο



# Ασκηση

Φτιάξτε ένα script που να αναγνωρίζει τον βαθμό ενός μαθητή.

Αν ο βαθμός είναι μικρότερος από 10 να εμφανίζει μήνυμα: «Sorry»

Αν ο βαθμός είναι μικρότερος ή ίσος από 15 να εμφανίζει το μήνυμα «Fair Enough»

Αν ο βαθμός είναι 16 μέχρι 20 να εμφανίζει «Well Done»

Σε όλες τις άλλες περιπτώσεις να εμφανίζει μήνυμα «Enter number 1-20»

# switch

Η Switch Statement μας δίνει την δυνατότητα να εμφανίσουμε περιπτώσεις (αντίστοιχες του if-else) με έναν πιο εύκολο τρόπο.

Η Switch μπορεί να ελέγξει μόνο δεδομένα.  
Δεν κάνει συγκρίσεις.

Το break ανάμεσα στις περιπτώσεις δουλεύει σαν δικλείδα ασφαλείας, να μην κάνει ολίσθηση από το case "1" στο case "2"

```
switch (myVar) {  
    case "1":  
        do_something;  
        break;  
    case "2":  
        do_something;  
        break;  
    case "3":  
        do_something;  
        break;  
    default:  
        do_something;  
        break;  
}
```

# Ασκηση (με switch)

Φτιάξτε ένα script που να ελέγχει την μάρκα αυτοκινήτου που εισάγει ο χρήστης. Να εμφανίζει τα αντίστοιχα μηνύματα:

| Μάρκα   | Μήνυμα  |
|---------|---------|
| Citroen | French  |
| Toyota  | Japan   |
| BMW     | Germany |
| Kia     | Korea   |
| Bentley | UK      |

# Homework 1

**Δώσε αριθμούς:**

number 1:

number 2:

**Επέλεξε την πράξη που θες να κάνεις (κλικ)**



Εδώ θα δεις το αποτέλεσμα



# Homework 2

- ▶ Δημιουργία calculator που ο χρήστης θα πληκτρολογεί σε 3 inputs, 2 αριθμούς και την πράξη (+, -, \*, /) και θα εμφανίζεται το αποτέλεσμα.

# JavaScript Strings

# String Length

- ▶ `.length` για να βρεις το μήκος μιας συμβολοσειράς χρησιμοποιείς την ιδιότητα

```
<p id="demo"></p>
```

```
<script>
```

```
var txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
```

```
document.getElementById("demo").innerHTML = txt.length;
```

```
</script>
```

26



# Finding a String in a String

- ▶ **indexOf()** επιστρέφει την θέση της πρώτης εμφάνισης ενός string
- ▶ **lastIndexOf()** επιστρέφει την θέση της τελευταίας εμφάνισης ενός string

```
<script>  
var str = "Please locate where 'locate' occurs!";  
var pos = str.indexOf("locate");  
document.getElementById("demo").innerHTML = pos;  
</script>
```

7

# Extracting String Methods

- ▶ **Slice()** εξαγεί ένα τμήμα μιας συμβολοσειράς και επιστρέφει το εξαγόμενο τμήμα σε μια νέα συμβολοσειρά. (δεχεται και αρνητικές τιμές)
- ▶ **Substring()** ακριβώς το ίδιο αλλά δεν δέχεται αρνητικές τιμές
- ▶ Η μέθοδος λαμβάνει 2 παραμέτρους: τη θέση εκκίνησης και την τελική θέση
- ▶ **Substr()** Μοιάζει με το slice() αλλά η δεύτερη παράμετρος δείχνει το μήκος του εξαγόμενου κειμένου

Banana

```
var str = "Apple, Banana, Kiwi";  
var res = str.substr(7, 6);
```

Banana

```
var str = "Apple, Banana, Kiwi";  
var res = str.slice(7, 13);
```

# Replacing String Content

- **Replace()** αντικαθιστά ένα string με ένα άλλο

```
str = "Please visit Microsoft!";  
var n = str.replace("Microsoft", "W3Schools");
```

# Converting to Upper and Lower Case



- ▶ `toUpperCase()`
- ▶ `toLowerCase()`

# JavaScript Number Methods



# toFixed() Method

- ▶ Επιστρέφει μια συμβολοσειρά, με τον αριθμό γραμμένο με έναν καθορισμένο αριθμό δεκαδικών

```
var x = 9.656;  
x.toFixed(0);           // returns 10  
x.toFixed(2);           // returns 9.66  
x.toFixed(4);           // returns 9.6560  
x.toFixed(6);           // returns 9.656000
```

# parseInt() Function

- ▶ Μετατρέπει μια συμβολοσειρά σε ακέραιο
- ▶ Αντοίστοιχα parseFloat()

# Date methods

- ▶ `new Date()` : εμφνίζει τρέχουσα ημερομηνία



# JavaScript Math Object

## ► Math.round()

Επιστρέφει την τιμή του x  
στρογγυλοποιημένη στον κοντινότερο αριθμό

```
Math.round(4.7);    // returns 5
```

```
Math.round(4.4);    // returns 4
```

## ► Math.sqrt()

Επιστρέφει την τετραγωνική ρίζα του x

- ▶ **Math.random()** Επιστρέφει έναν τυχαίο πραγματικό αριθμό από το 0 έως το 1
- ▶ **Math.floor()** Επιστρέφει το ακέραιο κομμάτι ενός δεκαδικού

#### Example

```
Math.floor(Math.random() * 10);    // returns a random integer from 0 to 9
```

Try it Yourself »

#### Example

```
Math.floor(Math.random() * 11);    // returns a random integer from 0 to 10
```

Try it Yourself »

# JS Regular Expressions

- ▶ Είναι μια ακολουθία χαρακτήρων που σχηματίζει ένα πρότυπο αναζήτησης
- ▶ Όταν αναζητάς δεδομένα σε ένα κείμενο, μπορείς να χρησιμοποιήσεις αυτό το πρότυπο αναζήτησης για να περιγράψεις αυτό που αναζητάς.
- ▶ Μπορεί να είναι ένας μόνος χαρακτήρας ή ένα πιο πολύπλοκο μοτίβο.

# Syntax

```
/pattern/modifiers;
```

## Example

```
var patt = /w3schools/i;
```

- ▶ /w3school/ → Είναι μια regular expression
- ▶ i → κάνει την αναζήτηση case-insensitive

# Regular Expression Patterns

## Expression

## Description

`[abc]`

Find any of the characters between the brackets

`[0-9]`

Find any of the digits between the brackets

`(x|y)`

Find any of the alternatives separated with `|`

# JS Arrays

- ▶ Οι πίνακες χρησιμοποιούνται για την αποθήκευση πολλαπλών τιμών σε μία μόνο μεταβλητή
- ▶ Ένας πίνακας είναι μια ειδική μεταβλητή, η οποία μπορεί να κατέχει περισσότερες από μία τιμές τη φορά.

# Δημιουργία Array

- ▶ Ο ευκολότερος τρόπος για την δημιουργία ενός πίνακα

Syntax:

```
var array_name = [item1, item2, ...];
```

Example

```
var cars = ["Saab", "Volvo", "BMW"];
```

# JavaScript Array Methods

- ▶ Μετατροπή Arrays σε Strings
  - **toString()** μετατρέπει έναν πίνακα σε ένα string με τις τιμές του πίνακα χωρισμένες με κομμάτι
  - **join()** μετατρέπει έναν πίνακα σε ένα string αλλά αντίθετα από το **toString()** μπορείς να επιλέξεις ποιος θα είναι ο separator



# Popping and Pushing

- ▶ **pop()** διαγράφει το τελευταίο στοιχείο από έναν πίνακα

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
var x = fruits.pop();           // the value of x is "Mango"
```

- ▶ **push()** προσθέτει ένα καινούριο στοιχείο στον πίνακα (στο τέλος του)

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
var x = fruits.push("Kiwi");    // the value of x is 5
```

- ▶ **shift()** λειτουργεί σαν το pop() αλλά διαγράφει το πρώτο στοιχείο του πίνακα και μετακινεί όλα τα άλλα μια θέση μπροστά

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
var x = fruits.shift();    // the value of x is "Banana"
```

- ▶ **concat()** δημιουργεί έναν καινούριο πίνακα συγχωνεύοντας υπάρχοντες πίνακες

```
var arr1 = ["Cecilie", "Lone"];  
var arr2 = ["Emil", "Tobias", "Linus"];  
var arr3 = ["Robin", "Morgan"];  
var myChildren = arr1.concat(arr2, arr3);    // Concatenates arr1 with arr2 and arr3
```

- ▶ **splice()** μπορεί να χρησιμοποιηθεί για να προστεθούν καινούρια στοιχεία σε έναν πίνακα

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.splice(2, 0, "Lemon", "Kiwi");
```

- ▶ Η πρώτη παράμετρος (2) δείχνει την θέση όπου τα νέα στοιχεία θα προστεθούν
- ▶ Η δεύτερη παράμετρος (0) ορίζει πόσα στοιχεία θα πρέπει να μετακινηθούν
- ▶ Οι υπόλοιπες παράμετροι ("Lemon" , "Kiwi") ορίζουν τα στοιχεία που θα προστεθούν

# Sorting an Array

- ▶ **sort()** ταξινομεί αλφαβητικά έναν πίνακα
- ▶ **reverse()** ταξινομεί με φθίνουσα σειρά έναν πίνακα

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.sort();           // First sort the elements of fruits  
fruits.reverse();        // Then reverse the order of the elements
```

---

- ▶ **forEach()** καλεί μια συνάρτηση για κάθε στοιχείο του πίνακα

```
var txt = "";  
var numbers = [45, 4, 9, 16, 25];  
numbers.forEach(myFunction);  
  
function myFunction(value, index, array) {  
    txt = txt + value + "<br>";  
}
```

- ▶ **filter()** δημιουργεί έναν νέο πίνακα με τα στοιχεία του πίνακα που ικανοποιούν μια συνθήκη

```
var numbers = [45, 4, 9, 16, 25];  
var over18 = numbers.filter(myFunction);  
  
function myFunction(value, index, array) {  
    return value > 18;  
}
```