



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Προγραμματισμός στο Διαδίκτυο

JavaScript

Μανώλης Μαραγκουδάκης

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αιγαίου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

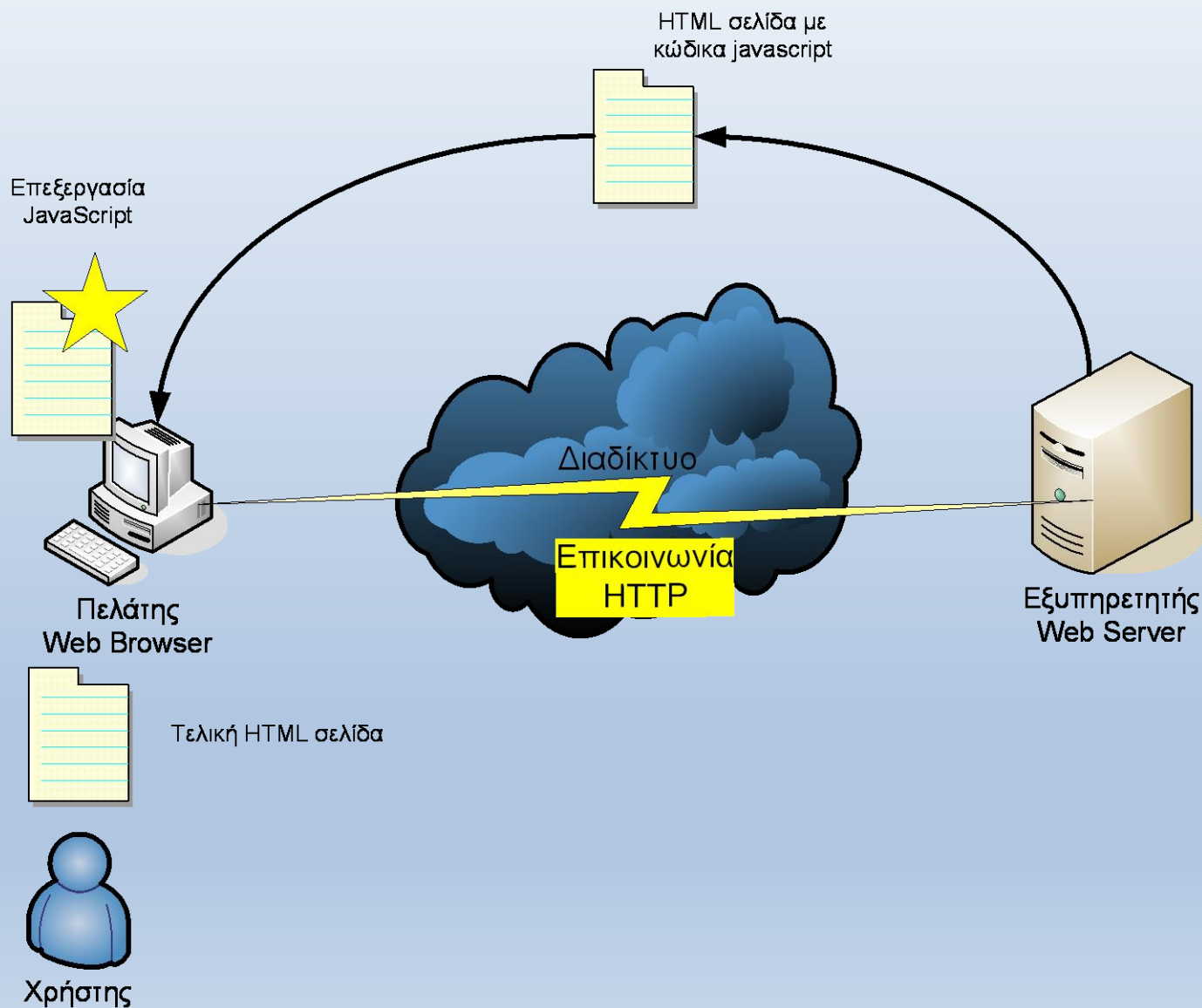
Προγραμματισμός στο Διαδίκτυο

Ενότητα 4 - JavaScript

Μανώλης Μαραγκουδάκης

Πανεπιστήμιο Αιγαίου
Τμήμα Μηχανικών Πληροφοριακών και
Επικοινωνιακών Συστημάτων

Προγραμματισμός στην πλευρά του πελάτη

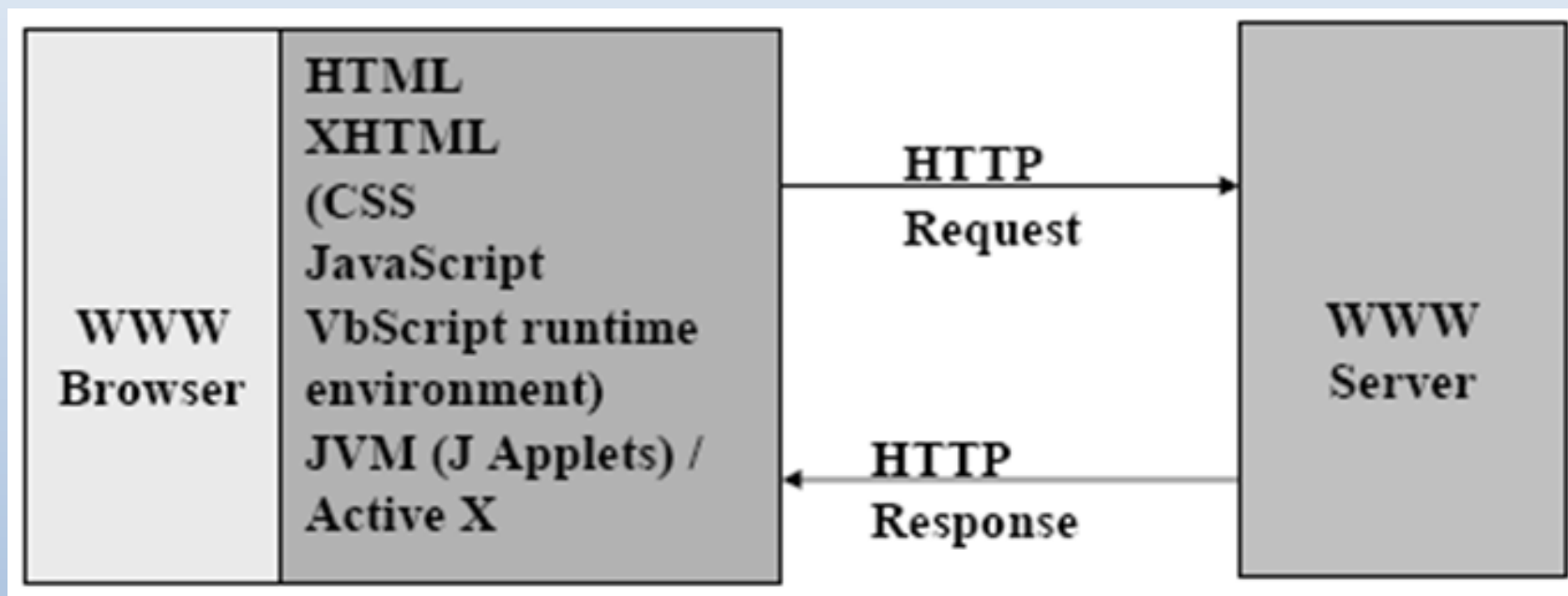


Προγραμματισμός στην πλευρά του πελάτη



- Τα ενεργά στοιχεία (κώδικας) «κατεβαίνουν» στον web browser μαζί με τη στατική πληροφορία (HTML)
- Ο Browser πρέπει να μπορεί να εντοπίσει και να εκτελέσει τα ενεργά στοιχεία – προγράμματα
- Δύο κατηγορίες τεχνολογιών:
 1. Ο κώδικας είναι προ-μεταγλωττισμένος, ο browser τον «ερμηνεύει» (εκτελεί), π.χ. Java Applets
 2. Κώδικας σεναρίου (script), ο browser τον επεξεργάζεται και αν είναι συντακτικά σωστός τον εκτελεί, π.χ. Javascript, Vbscript.

Προγραμματισμός στην πλευρά του πελάτη



Προγραμματισμός στην πλευρά του πελάτη: Πλεονεκτήματα/Μειονεκτήματα



- **Κατάλληλη τεχνολογία όταν:**
 - Ελέγχεται η είσοδος του χρήστη (form validation)
 - Απαιτούνται πολύπλοκες λειτουργίες στον πελάτη (πχ. διάδραση)
- **Πλεονεκτήματα:**
 - Ταχύτερες εφαρμογές λόγω τοπικής εκτέλεσης
 - Δυνατότητα εκτέλεσης χωρίς συνεχή σύνδεση server/client
 - Δυνατότητες, ευελιξία, ευχρηστία και καλαισθησία ιστοσελίδων
- **Μειονεκτήματα:**
 - Ο κώδικας είναι ορατός
 - Δεν μπορεί να έχει πρόσβαση στο τοπικό file system για λόγους ασφάλειας (εξαιρέση τα μπισκότα)
 - Δεν τρέχουν σε όλους τους clients (browsers)

Προγραμματισμός στην πλευρά του πελάτη: Υπάρχουσες Τεχνολογίες



- **Active-X controls:** προγράμματα που τρέχουν σε browser και δουλεύουν μόνο με Microsoft Internet Explorer σε Windows (τεχνολογία Microsoft)
- **Flash: τεχνολογία για δημιουργία animation εφέ σε ιστοσελίδες.**
 - Απαιτεί λιγότερο bandwidth από ένα video που πρέπει να γίνει «download» πλαίσιο με πλαίσιο (δουλεύει ικανοποιητικά και σε αργές Internet συνδέσεις).
 - Απαιτείται ειδικό λογισμικό εγκατεστημένο στον client (plug-in στον browser). Τέτοια plug-ins είναι δωρεάν διαθέσιμα για τους περισσότερους browsers . Ωστόσο δεν μπορεί να υποτεθεί ότι όλοι οι χρήστες «βλέπουν» τα flash animations.
 - Για το σχεδιασμό Flash animations απαιτείται ειδικό λογισμικό.
 - Αν και τα Flash animations μπορεί να είναι εντυπωσιακά, μπορεί να είναι και ενοχλητικά (π.χ. κάποιος θέλει να αναζητήσει κάποια συγκεκριμένη πληροφορία και θα πρέπει να περιμένει να «κατέβει» το animation.
- **VBScript:** μια γλώσσα βασισμένη στη Visual Basic, με παρόμοιες δυνατότητες με τη Javascript, αλλά δουλεύει μόνο στον Internet Explorer.
- **Java Applets**
- **Javascript**

JavaScript



- Γλώσσα σεναρίου (script language)
- Αναπτύχθηκε αρχικά από τη Netscape (αρχικό όνομα **Livescript**) – καμιά ουσιαστική σχέση με την Java
- Τρέχει σε όλους τους browsers (cross-platform)
- Γράφεται απευθείας μέσα σε HTML έγγραφα με ή χωρίς χρήση ειδικού λογισμικού (HTML authoring tools, script editors)
- Πλεονεκτήματα/Μειονεκτήματα client-side programming
 - Αντίδραση/επεξεργασία στις κινήσεις του χρήστη (user actions)
 - Επεξεργασία/έλεγχος δεδομένων εισόδου του χρήστη (user input)
 - Δυναμική εμφάνιση (συγγραφή) HTML
 - Βελτίωση ευχρηστίας, ευελιξίας πλοήγησης
 - Εύκολη στην εκμάθηση και χρήση – όχι όμως πλήρης γλώσσα

Η Javascript είναι γλώσσα σεναρίου



- Μια γλώσσα script επιτρέπει να γράψουμε ένα πρόγραμμα που δεν χρειάζεται να περάσει από τη διαδικασία της μεταγλώττισης (μετάφραση κώδικα σε γλώσσα μηχανής – εκτελέσιμο αρχείο), όπως συμβαίνει με άλλες γλώσσες προγραμματισμού (Java, Fortran, ...)
- Ο κώδικας ενός script “διερμηνεύεται” άμεσα, από την εφαρμογή που το εκτελεί (π.χ. browser)
- Τυχόν συντακτικά λάθη θα γίνουν αντιληπτά όταν “τρέξει” το script. Στην περίπτωση της Javascript, ο χειρισμός των σφαλμάτων γίνεται από τον browser που εκτελεί τον Javascript κώδικα
- Διαφορά Μεταγλώττισης – Διερμηνείας (Compilation - Interpretation)

Χρήση της JavaScript



- Ο JavaScript κώδικας μπορεί να αντιδράσει σε συμβάντα (events), δηλαδή να εκτελεστεί όταν κάτι συμβεί, π.χ. όταν:
 - Μια σελίδα φορτωθεί
 - Όταν ο χρήστης κάνει κλικ σε ένα πλήκτρο
 - Όταν ο χρήστης επιλέξει κάποιο στοιχείο μιας λίστας
- Ο JavaScript κώδικας μπορεί να διαβάσει αλλά και να μεταβάλει το περιεχόμενο ενός HTML στοιχείου (HTML elements), π.χ. να διαβάσει το κείμενο που έχει γράψει ο χρήστης σε ένα πεδίο κειμένου (textfield)
- Εμφάνιση πληροφοριών / μηνυμάτων σε πλαίσια διαλόγου (dialog boxes), στη μπάρα κατάστασης ενός παραθύρου (status bar), κ.ο.κ.
- Ελέγχει δεδομένα που δίνονται σε μία φόρμα πριν αυτή γίνει submit

Το πρώτο σενάριο JavaScript



```
<script language="JavaScript">  
    document.write("Hello World with JavaScript");  
</script>
```

Απλά προγράμματα Javascript



```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3 <html lang="EN" dir="ltr" xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5 <meta http-equiv="content-type" content="text/xml; charset=utf-8" />
6 <title>HelloWorld.html</title>
7 <script type = "text/javascript">
8 //
9 // Hello, world!
10 alert("Hello, World!");
11 //]]&gt;
12 &lt;/script&gt;
13 &lt;/head&gt;
14 &lt;body&gt;
15 &lt;/body&gt;
16 &lt;/html&gt;</pre></div><div data-bbox="759 172 916 687" data-label="Text"><p>Είναι καλό να μπαίνει για το validation. Δηλώνει πως το περιεχόμενο είναι κειμενική πληροφορία και δεν θα πρέπει να εκλαμβάνεται ως XHTML</p></div><div data-bbox="306 732 683 947" data-label="Image"><img alt="A screenshot of a JavaScript Alert dialog box. The title bar says 'JavaScript Alert' and the message text says 'Hello, World!'. There is an 'OK' button at the bottom right."/><p>JavaScript Alert</p><p>Hello, World!</p><p>OK</p></div><div data-bbox="914 964 941 987" data-label="Page-Footer"><p>11</p></div>
```

Τοποθέτηση JavaScript κώδικα



- Head section
 - Η ετικέτα head φορτώνεται πρώτα
 - Μπορεί να καθυστερεί το «κατέβασμα» της σελίδας
 - Κατάλληλο για τοποθέτηση συναρτήσεων
- Body Section
 - Εκτελούνται όταν η σελίδα φορτώνεται
 - Η εκτέλεση μπορεί να απαιτεί την πλήρη φόρτωση της σελίδας
- External script

```
<script language="JavaScript" src="global.js"></script>
```
- Επιτρέπονται πολλαπλά «scripts» σε ένα html αρχείο

Τοποθέτηση στο Head



```
<html><head><title>My Page</title>
<script language="javascript" type="text/javascript">
function myFunction() {
    alert('Hello world');
}
</script>
</head>
<body>
<a href="javascript:myFunction();">Click here</a>
</body>
</html>
```

Τοποθέτηση στο Body



```
<html><head><title>My Page</title>
</head>
<body>
<a href="javascript:myFunction();">Click here</a>
<script language="javascript" type="text/javascript">
function myFunction() {
    alert('Hello world');
}
</script>
</body>
</html>
```


Μεταβλητές



```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3 <html lang="EN" dir="ltr" xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5 <meta http-equiv="content-type" content="text/xml; charset=utf-8" />
6 <title>prompt.html</title>
7 <script type = "text/javascript">
8 //<![CDATA[
9 var person = "";
10 person = prompt("What is your name?");
11 alert("Hi");
12 alert(person);
13 //]]>
14 </script>
15 </head>
16 <body>
17 </body>
18 </html>
```

JavaScript

What is your name?

OK Cancel

Μεταβλητές



- Ξεκινούν με γράμμα ή “_”
- Case sensitive
- Όχι κενά και δεσμευμένες λέξεις της γλώσσας
- Ορισμός
`var name (var optional)`
- Ανάθεση τιμής
`name = “Alex”`
`mynumber = 15`
- Τοποθέτηση σε κείμενο
`“My name is ” + name`

Μεταβλητές & τύποι δεδομένων



- Αριθμοί : ακέραιοι ή κινητής υποδιαστολής
- Boolean : true / false
- Strings : ανάμεσα σε μονά ή διπλά εισαγωγικά
- Αντικείμενα (objects)
- Null : διαφορετικός από τη μηδενική τιμή
 - Σημαίνει ότι ΔΕΝ έχει τιμή
- Undefined / Empty : η μεταβλητή δεν υπάρχει ή υπάρχει αλλά δεν έχει κάποια τιμή
- NaN = Not a Number : όταν το αποτέλεσμα μιας αριθμητικής πράξης δεν ορίζεται

! Δεν είναι απαραίτητο να δηλώνεται ο τύπος πριν την ανάθεση τιμής

Κατανοώντας τους τύπους των μεταβλητών



- Η JavaScript δεν θεωρεί απαραίτητο να της πεις αν μια μεταβλητή είναι π.χ. κείμενο ή αριθμός.
 - Αυτό όμως μπορεί να προκαλέσει προβλήματα.

```
<script type = "text/javascript">
//
var x = 5;
var y = 3;
var sum = x + y;
alert(x + "plus" + y + "equals" + sum);
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="495 339 840 516" data-label="Image"><img alt="Screenshot of a JavaScript alert dialog box titled '[JavaScript Application]'. The message inside says '5 plus 3 equals 8'. There is an 'OK' button at the bottom."/></div><div data-bbox="517 583 940 895" data-label="Text"><pre>&lt;script type = "text/javascript"&gt;
//<![CDATA[
var x = prompt("first number:");
var y = prompt("second number:");
var sum = x + y;
alert(x + " plus " + y + " equals " + sum);
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="88 686 379 974" data-label="Image"><img alt="A sequence of three browser dialog boxes. The first is a 'prompt' dialog for 'first number:' with '3' entered. The second is a 'prompt' dialog for 'second number:' with '5' entered. The third is an 'alert' dialog showing '3 plus 5 equals 35'."/></div><div data-bbox="920 964 947 987" data-label="Page-Footer"><p>18</p></div>
```

Κατανοώντας τους τύπους των μεταβλητών



- Η πληροφορία που αποθηκεύεται στον ΗΥ, από αρχεία μέχρι email είναι μια ακολουθία 0 και 1.
 - 01000001 μπορεί να σημαίνει διάφορα πράγματα:
 - Το 65, ή το A ή ένα χρώμα, ή ένα κομμάτι από τραγούδι, κτλ.
 - Η JavaScript «μαντεύει» τον τύπο μιας μεταβλητής από τα συμφραζόμενα
 - Συνήθως μαντεύει σωστά

Κατανοώντας τους τύπους των μεταβλητών

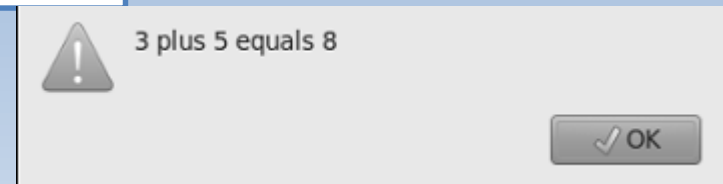


- Αν πάλι δεν μαντέψει σωστά, πρέπει εμείς να τη βοηθήσουμε με συναρτήσεις μετατροπής

<i>Function</i>	<i>From</i>	<i>To</i>	<i>Example</i>	<i>Result</i>
<code>parseInt()</code>	String	Integer	<code>parseInt("23")</code>	23
<code>parseFloat()</code>	String	Floating point	<code>parseFloat("21.5")</code>	21.5
<code>toString()</code>	Any variable	String	<code>myVar.toString()</code>	<i>varies</i>
<code>eval()</code>	Expression	Result	<code>eval("5 + 3")</code>	8
<code>Math.ceil()</code>	Floating point	Integer	<code>Math.ceil(5.2)</code>	6
<code>Math.floor()</code>			<code>Math.floor(5.2)</code>	5
<code>Math.round()</code>			<code>Math.round(5.2)</code>	5

Το προηγούμενο παράδειγμα σωστά αναδιατυπωμένο:

```
var x = prompt("first number:");  
var y = prompt("second number:");  
var sum = parseFloat(x) + parseFloat(y);  
  
alert(x + " plus " + y + " equals " + sum);
```



Τελεστές



Ισότητα	==
Ανισότητα	!=
Λογικό “and”	&&
Λογικό “or”	
Λογικό “not”	!
Ένωση strings	+
Μοναδιαία αύξηση/ μείωση	++ / --
Ανάθεση	=, +=, -=, *=, /=

If ... else



```
if (condition)
{ statements1; }
else
{ statements2; }
```

! else if

! varname = (condition)?trueVal:falseVal

myCar = (ferrari>porche) ? "RedFerrari" : "ColPorche"

Switch



```
switch (expression) {  
    case label1:  
        statements1;  
        break;  
    case label2:  
        statements2;  
        break;  
    default: statements;  
}
```

For



```
for (initial; condition; incr)
{ statements; }
```

```
for (i=0;i<10;i++)
{ document.write(i) }
```

Do ... while



while (condition)

{ statements }

do

{ statements; }

while (condition)

do

**{ document.write(i + "
")**

i=i+1 }

while (i < 10)

```
<script type = "text/javascript">
//<![CDATA[
// from while.html
answer = "-99";
while (answer != "5"){
answer = prompt("What is 3 + 2?");
if (answer == "5"){
alert("great!");
} else {
alert("try again...");
} // end if
} // end while
//]>
</script>
```

Έξοδος από βρόχο ... σχόλια



- `break` :
 - εγκαταλείπει την εκτέλεση του βρόχου
- `continue` :
 - εγκαταλείπει την απλή επανάληψη του βρόχου
- Σχόλια
 - //
 - /* ... */

Ορισμός συναρτήσεων



JavaScript

```
function mfun(arg1,arg2)  
{ statements; }
```

```
function noarg()  
{ document.write(15) }
```

```
function print(msg)  
{ document.write(msg) }
```

```
function sum(a, b)  
{ c = a + b  
  return c }
```

Κλήση συναρτήσεων



```
noarg()
```

```
print("Hello")
```

```
mySum = sum(23, 7)
```

```
alert("This is alert")
```



- Η JavaScript χρησιμοποιεί **try-catch** blocks

```
function message() {  
  try {  
    adddler("Welcome guest!");  
  }  
  catch(err) {  
    txt="There was an error on this page.\n\n";  
    txt+="Error description: " + err.message + "\n\n";  
    txt+="Click OK to continue.\n\n";  
    alert(txt);  
  }  
}
```

- Η εντολή **throw** μας επιτρέπει να δημιουργήσουμε ένα custom error.

JavaScript Objects



- Υπάρχουν διάφοροι τύποι αντικειμένων
 - Εγγενή της γλώσσας (π.χ. Math, Date, String, Array)
 - Browser objects (π.χ. Window, Navigator, History)
 - HTML DOM objects (π.χ. Document, Elements, Attributes, Events)
 - HTML objects (form, canvas, image, table, anchors)
 - User-defined objects
 - Τα αντικείμενα έχουν:
 - Ιδιότητες : *objectName.propertyName*
 - Μεθόδους : *objectName.methodName()*
- ! Case sensitive**

User-defined αντικείμενα (1/2)



Απ' ευθείας δημιουργία ενός αντικειμένου:

```
person=new Object();  
person.firstname="John";  
person.lastname="Doe";  
person.age=50;  
person.eyecolor="blue";
```

ή

```
person={firstname:"John", lastname:"Doe",  
age:50, eyecolor:"blue"};
```

User-defined αντικείμενα (2/2)



Με χρήση ενός κατασκευαστή αντικειμένων:

```
function person(firstname,lastname,age,eyecolor) {  
    this.firstname=firstname;  
    this.lastname=lastname;  
    this.age=age;  
    this.eyecolor=eyecolor;  
}
```

Δημιουργία στιγμιότυπων αντικειμένων:

```
var myFather=new person("John","Doe",50,"blue");  
var myMother=new person("Sally","Rally",48,"green");
```

Προσθήκη μεθόδων



```
function person(firstname,lastname,age,eyecolor) {  
    this.firstname=firstname;  
    this.lastname=lastname;  
    this.age=age;  
    this.eyecolor=eyecolor;  
  
    this.changeLname=changeLname;  
    function changeLname(name)  
    {  
        this.lastname=name;  
    }  
}
```

Παράδειγμα κλήσης: `myMother.changeLname("Doe");` 33

Διάσχιση ιδιοτήτων ενός αντικειμένου



Με χρήση ενός for-loop:

```
var person={fname:"John",lname:"Doe",age:25};
```

```
for (x in person) {  
    txt=txt + person[x];  
}
```

Βασικά Αντικείμενα της JavaScript

Το αντικείμενο Math (object Math)

Το αντικείμενο Math



- Το αντικείμενο Math διαθέτει ιδιότητες και μεθόδους που χρησιμοποιούνται για μαθηματικούς υπολογισμούς, π.χ. για να υπολογίσουμε την τετραγωνική ρίζα ενός αριθμού ή ένα εκθετικό
- Οι ιδιότητες του αντικειμένου Math αποτελούν κάποιες προκαθορισμένες μαθηματικές τιμές, π.χ.
 - `document.write (Math.E) //` Τυπώνει τη σταθερά Euler (≈ 2.7118)
 - `document.write (Math.PI); //` Τυπώνει την τιμή του π (≈ 3.14159)
 - `document.write (Math.SQRT(2)); //` Τυπώνει (≈ 1.414)

Μέθοδοι του αντικειμένου Math



Μέθοδος	Περιγραφή
<code>abs(x)</code>	Υπολογίζει την απόλυτη τιμή του x
<code>exp(x)</code>	Υπολογίζει το e^x
<code>log(x)</code>	Υπολογίζει το φυσικό λογάριθμο του x
<code>sqrt(x)</code>	Υπολογίζει την τετραγωνική ρίζα του x
<code>round(x)</code>	Επιστρέφει τον ακέραιο πλησιέστερο στον x
<code>floor(x)</code>	Επιστρέφει τον αμέσως μικρότερο ακέραιο από τον x
<code>ceil(x)</code>	Επιστρέφει τον αμέσως μεγαλύτερο ακέραιο από τον x
<code>max(x, y)</code>	Επιστρέφει τον μεγαλύτερο αριθμό από τους x, y
<code>min(x, y)</code>	Επιστρέφει τον μικρότερο αριθμό από τους x, y
<code>pow(x, y)</code>	Υπολογίζει το x^y
<code>random()</code>	Επιστρέφει έναν τυχαίο αριθμό μεταξύ 0 και 1

Ασκηση:



- Φτιάξτε ένα πρόγραμμα που προσομοιώνει το ρίξιμο ενός ζαριού:
 - Προτεινόμενη λύση:
 - Δημιουργούμε ένα τυχαίο αριθμό από 0 έως 1
 - Τον πολλαπλασιάζουμε με το 6 (από 0...5.99 αλλά ποτέ 6)
 - Τον στρογγυλοποιούμε στον πλησιέστερο ακέραιο

```
<script type = "text/javascript">
//
var number = Math.random();
alert(number);
var biggerNumber = number * 6;
alert(biggerNumber);
var dice = Math.ceil(biggerNumber);
alert(dice);
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="430 540 717 699" data-label="Image"><img alt="Screenshot of a JavaScript Alert dialog box showing the number 0.2603881233371794."/></div><div data-bbox="436 709 721 894" data-label="Image"><img alt="Screenshot of a JavaScript Alert dialog box showing the number 1.5623287400230765."/></div><div data-bbox="721 635 1000 816" data-label="Image"><img alt="Screenshot of a JavaScript Alert dialog box showing the number 2 and a checkbox for 'Prevent this page from creating additional dialogs.'"/></div><div data-bbox="921 964 947 987" data-label="Page-Footer"><p>39</p></div>
```

Το αντικείμενο Date (object Date)

Το αντικείμενο Date



- Το αντικείμενο Date δίνει τη δυνατότητα να χειριστούμε τιμές ημερομηνίας/ώρας ή να ανακτήσουμε συγκεκριμένες τιμές που θα χρησιμοποιήσουμε αργότερα στα script μας
- Μπορούμε, για παράδειγμα, να ανακτήσουμε την τρέχουσα ημερομηνία του συστήματος όπου εκτελείται το script, να υπολογίσουμε πόσος χρόνος έχει παρέλθει από μια συγκεκριμένη ώρα ως την τρέχουσα ώρα, χρονικές διαφορές μεταξύ δύο ημερομηνιών, να μορφοποιήσουμε την εμφάνιση της ημερομηνίας/ώρας με βάση τις προτιμήσεις μας, κλπ
- Η τρέχουσα ώρα του συστήματος ανακτάται εύκολα:

```
var d = new Date(); // Αποθηκεύουμε την τρέχουσα ημ/νία στη μεταβλητή d  
document.write (d); // Θα τυπώσει κάτι σαν Tue Mar 17 14:43:04 UTC+0200  
2014
```

Μέθοδοι του αντικειμένου Date



Μέθοδος	Περιγραφή
<code>getDay()/</code> <code>getDate()/</code> <code>getMonth()/</code> <code>getFullYear()</code>	Επιστρέφει την ημέρα της εβδομάδας σε αριθμό (0-6) / ημέρα του μήνα/ μήνα (0-11) / έτος για μια συγκεκριμένη ημερομηνία (τετραψήφιος)
<code>setDate()/</code> <code>setMonth()/</code> <code>setFullYear()</code>	Ορίζει την ημέρα του μήνα/ μήνα / έτος μιας ημερομηνίας
<code>getTime()</code>	Επιστρέφει τα msec που έχουν περάσει από την 1/1/1970 (χρησιμοποιείται για να υπολογίσουμε διαφορές μεταξύ ημερομηνιών)

Μέθοδοι του αντικειμένου Date



Μέθοδος	Περιγραφή
<code>getHours()/</code> <code>getMinutes()/</code> <code>getSeconds ()</code>	Επιστρέφει την ώρα (0-23) / λεπτά (0-59) / δευτερόλεπτα (0-59)
<code>setHours()/</code> <code>setMinutes()/</code> <code>setSeconds ()</code>	Ορίζει την ώρα / λεπτά / δευτερόλεπτα

Αντικείμενο Date



Ποιο θα είναι το αποτέλεσμα της εκτέλεσης του παρακάτω σεναρίου;

```
<script language="javascript">
var d = new Date()
document.write("The date is: " + d.getDate() + "/" + (d.getMonth() + 1) + "/" +
  d.getFullYear() + "<br>")
document.write("The time is: " + d.getHours() + ":" + d.getMinutes() + ":" +
  d.getSeconds() + "<br>")
var weekday = new Array("Sunday", "Monday", "Tuesday", "Wednesday",
  "Thursday", "Friday", "Saturday")
document.write("Today is " + weekday[d.getDay()] + "<br>")
d.setFullYear("1990")
document.write(d + "<br>")
</script>
```

Αλλάζει το
έτος της
ημ/νίας d σε
1990

Επιστρέφει 0 αν είναι
Κυριακή, 1 αν είναι
Δευτέρα, κοκ

Πίνακες ή Διατάξεις (arrays) στη Javascript

Ορισμός ενός πίνακα



```
var arrayname = new Array (στοιχείο1, στοιχείο2, ..., στοιχείοN);
```

```
var people = new Array("Μάκης", "Τάκης", "Σάκης");
```

Όνομα πίνακα

Στοιχεία πίνακα

Μάκης

Τάκης

Σάκης

Πίνακας 'people'

people[0]

people[1]

people[2]

```
document.write(people[1]); // Θα τυπώσει 'Τάκης'
```

```
people[2] = people[0]; // Θα καταχωρήσει 'Μάκης' στη θέση του 'Σάκης'
```

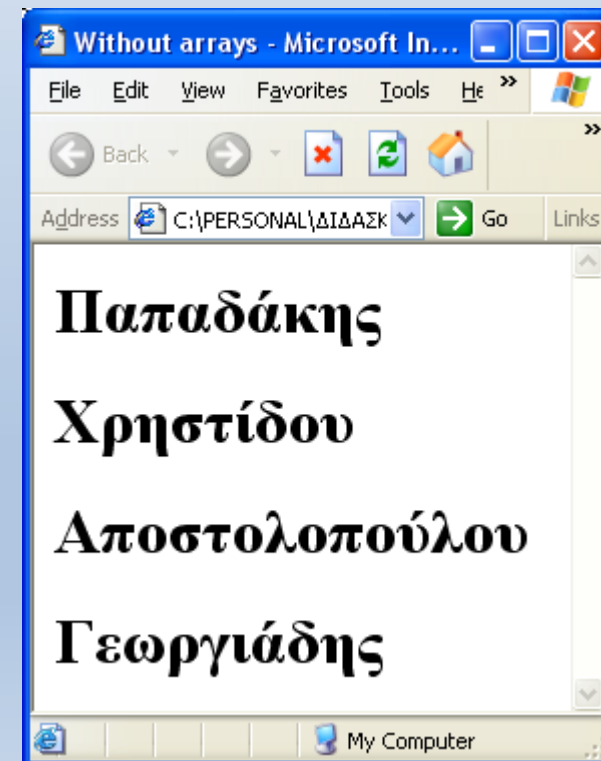

Παράδειγμα (χωρίς χρήση πινάκων)



Αποθήκευση ομοειδών δεδομένων σε διαφορετικές μεταβλητές

```
<script language="JavaScript">
  var student1 = "Παπαδάκης";
  var student2 = "Χρηστίδου";
  var student3 = "Αποστολοπούλου";
  var student4 = "Γεωργιάδης";
  document.write("<h1>" + student1 + "</h1>");
  document.write("<h1>" + student2 + "</h1>");
  document.write("<h1>" + student3 + "</h1>");
  document.write("<h1>" + student4 + "</h1>");
</script>
```

Εκτύπωση κάθε μεταβλητής με ξεχωριστή εντολή



Παράδειγμα (με χρήση πινάκων)



Αποθήκευση ομοειδών δεδομένων
(φοιτητών) σε πίνακα 4 στοιχείων

```
<script language="JavaScript">
```

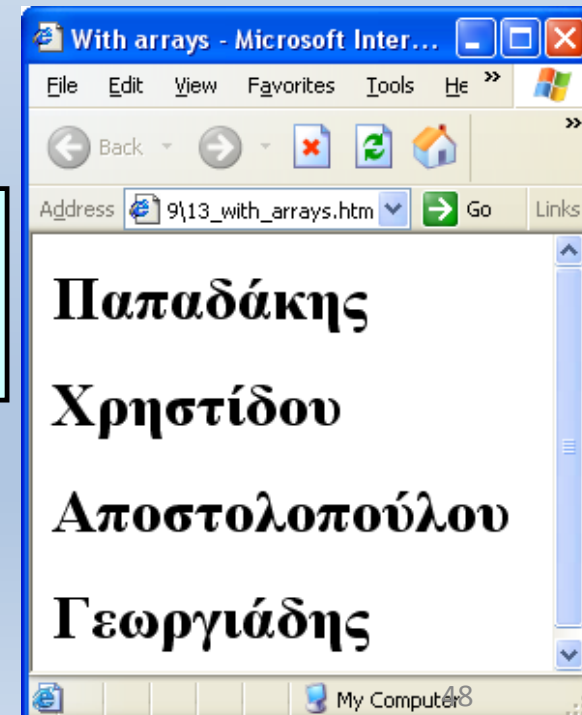
```
var students = new Array ("Παπαδάκης", "Χρηστίδου",  
"Αποστολοπούλου", "Γεωργιάδης");
```

```
for (i=0; i < students.length; i++)
```

```
document.write("<h1>" + students[i] + "</h1>");
```

```
</script>
```

Το `students.length` δίνει το
πλήθος των στοιχείων του πίνακα
`students`



Ορισμός στοιχείων πίνακα κατά την εκτέλεση (έπειτα από τον ορισμό του πίνακα)



- Τα στοιχεία ενός πίνακα δεν δίνονται απαραίτητα κατά τον ορισμό του πίνακα
- Ορισμός πίνακα με δέσμευση χώρου 4 στοιχείων:
 - `var students = new Array(4);`
- Ορισμός πίνακα με απροσδιόριστο μελλοντικό πλήθος στοιχείων:
 - `var students = new Array();`
 - `students[0] = “Μάκης”` // Τα στοιχεία δίνονται αργότερα
 - `students[1] = “Τάκης”`
 - `students[2] = “Σάκης”`
 - `document.write(“Πλήθος = ” + students.length);` // Θα τυπώσει ‘3’

Παράδειγμα



- Distance calculator

	<i>0) Indianapolis</i>	<i>1) New York</i>	<i>2) Tokyo</i>	<i>3) London</i>
0) Indianapolis	0	648	6476	4000
1) New York	648	0	6760	3470
2) Tokyo	6476	6760	0	5956
3) London	4000	3470	5956	0

```
cityName = new Array("Indianapolis", "New York", "Tokyo", "London");
distance = new Array (
  new Array (0, 648, 6476, 4000),
  new Array (648, 0, 6760, 3470),
  new Array (6476, 6760, 0, 5956),
  new Array (4000, 3470, 5956, 0)
);
```

```
function getCity(){
  var theCity = "";
  var cityMenu = "Please choose a city by typing a
  number: \n";
  cityMenu += "0) Indianapolis \n";
  cityMenu += "1) New York \n";
  cityMenu += "2) Tokyo \n";
  cityMenu += "3) London \n";
  theCity = prompt(cityMenu);
  return theCity;
}
```

```
function main(){
  var output = "";
  var from = getCity();
  var to = getCity();
  var result = distance[from][to];
  output = "The distance from " + cityName[from];
  output += " to " + cityName[to];
  output += " is " + result + " miles.";
  alert(output);
}
main();
```

Μέθοδοι του αντικειμένου Array



Μέθοδος	Περιγραφή
<code>concat()</code>	Συνενώνει τα στοιχεία 2 ή περισσότερων πινάκων σε ένα νέο πίνακα
<code>join()</code>	Δημιουργεί μια συμβολοσειρά που περιέχει όλα τα στοιχεία ενός πίνακα, χωρίζοντάς τα με ένα διαχωριστικό
<code>sort()</code>	Ταξινομεί αλφαβητικά τα στοιχεία ενός πίνακα
<code>reverse()</code>	Αντιστρέφει τη σειρά των στοιχείων ενός πίνακα (το πρώτο τελευταίο, το δεύτερο προ-τελευταίο, κοκ)
<code>pop()</code>	Αφαιρεί το τελευταίο στοιχείο από ένα πίνακα και επιστρέφει το διαγραμμένο στοιχείο
<code>push()</code>	Προσθέτει ένα στοιχείο στο τέλος ενός πίνακα και επιστρέφει το νέο μέγεθος του πίνακα

Μέθοδοι του αντικειμένου Array



Μέθοδος	Περιγραφή
<code>shift()</code>	Αφαιρεί το πρώτο στοιχείο από ένα πίνακα και επιστρέφει το διαγραμμένο στοιχείο
<code>unshift()</code>	Προσθέτει ένα στοιχείο στην αρχή ενός πίνακα και επιστρέφει το νέο μέγεθος του πίνακα
<code>slice()</code>	Εξάγει ένα συγκεκριμένο τμήμα ενός πίνακα και επιστρέφει το τμήμα αυτό ως ένα νέο πίνακα
<code>splice()</code>	Διαγράφει στοιχεία από ένα πίνακα ή αντικαθιστά στοιχεία σε ένα πίνακα

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.shift();
```



Orange,Apple,Mango

```
var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];  
var citrus = fruits.slice(1, 3);
```



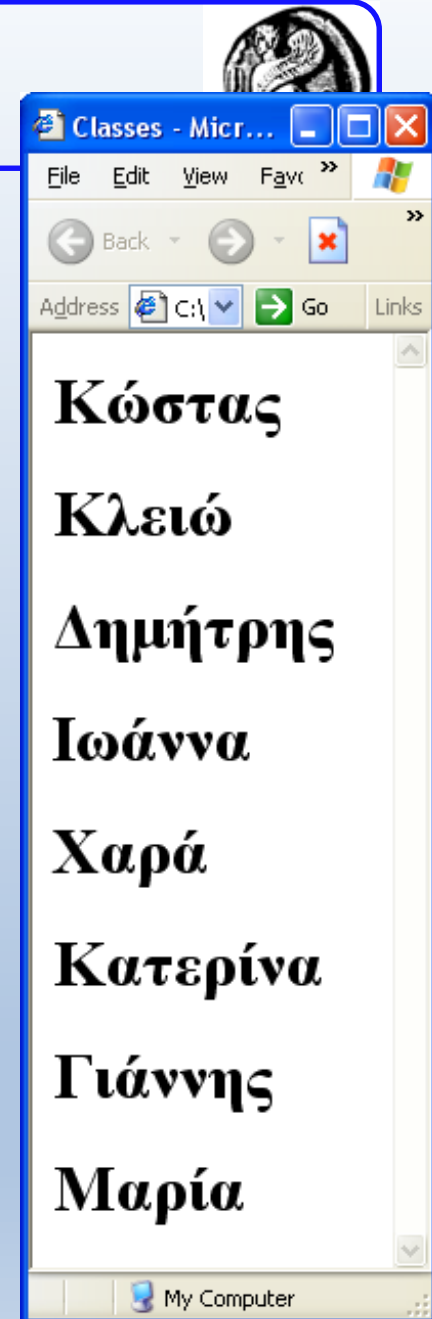
Orange,Lemon

Δεν περιλαμβάνει το στοιχείο στη θέση αυτή!

Παράδειγμα: Μέθοδος concat

```
<script language="JavaScript">
  var class_a = new Array ("Κώστας", "Κλειώ");
  var class_b = new Array ("Δημήτρης", "Ιωάννα", "Χαρά",
    "Κατερίνα");
  var class_c = new Array ("Γιάννης", "Μαρία");
  var everybody = class_a.concat(class_b, class_c);
  for (i=0; i < everybody.length; i++)
    document.write("<h1>" + everybody[i] +
      "</h1>");
</script>
```

Συνενώνει τα στοιχεία των πινάκων
class_a, class_b και class_c,
αποθηκεύοντάς τα στον πίνακα
everybody



Δημιουργία βασικών αντικειμένων



- Δήλωση

```
var critter = new Object();  
critter.name = "Milo";  
critter.age = 5;  
alert("the critter's name is " + critter.name);
```

- Ιδιότητες

```
var critter = new Object();  
  
critter.name = "Milo";  
critter.age = 5;  
  
critter.talk = function(){  
msg = "Hi! My name is " + this.name;  
msg += " and I'm " + this.age;  
alert(msg);  
};  
  
critter.talk();
```


Reusable Objects



```
function Critter(IName, IAge){
  this.name = IName;
  this.age = IAge;
  this.talk = function(){
    msg = "Hi! My name is " + this.name;
    msg += " and I'm " + this.age;
    alert(msg);
  } // end talk
} // end Critter class

function main(){
  critterA = new Critter("Alpha", 1);
  critterB = new Critter("Beta", 2);

  critterB.name = "Charlie";
  critterB.age = 3;
  critterA.talk();
  critterB.talk();
} // end main

main();
```

Οι κλάσεις στην
JavaScript είναι
εμπλουτισμένες
συναρτήσεις

Εισαγωγή στη JSON



- JavaScript Object Notation (JSON)
- Καθολικά πλέον αποδεκτός τρόπος ανταλλαγής δεδομένων μεταξύ Web προγραμμάτων
- Εναλλακτικός τρόπος αναπαράστασης της XML

```
var critter = {  
  "name": "George",  
  "age": 10  
};
```

Μοιάζει με πίνακα αλλά
δεν έχει αριθμητικούς
δείκτες αλλά ιδιότητες,
άρα μοιάζει περισσότερο
με Object

```
alert(critter["name"]);  
Ιδίο με:  
alert(critter.age);
```

JSON



```
var distance = {  
  "Indianapolis" :  
    { "Indianapolis": 0,  
      "New York": 648,  
      "Tokyo": 6476,  
      "London": 4000 },  
  "New York" :  
    { "Indianapolis": 648,  
      "New York": 0,  
      "Tokyo": 6760,  
      "London": 3470 },  
  "Tokyo" :  
    { "Indianapolis": 6476,  
      "New York": 6760,  
      "Tokyo": 0,  
      "London": 5956 },  
  "London" :  
    { "Indianapolis": 4000,  
      "New York": 3470,  
      "Tokyo": 5956,  
      "London": 0 },  
};
```

```
alert(distance["Indianapolis"]["London"]);
```

```
alert(distance.Indianapolis.Tokyo);
```

```
alert(distance["London"].Tokyo);
```

Strings στη Javascript

String object



- Ιδιότητες
 - length
- Μέθοδοι
 - `bold()`, `blink()`, `charAt()`, `indexOf()`, `split()`, `substring()`, `toLowerCase()`, `toUpperCase()`, ...
 - `indexOf(str)` : επιστρέφει την πρώτη εμφάνιση του `str`
 - `substring(a,b)` : τους χαρακτήρες από το a^{th} μέχρι και το $b-1^{\text{th}}$ (η αρίθμηση από το 0)

Μέθοδοι του String αντικειμένου



Method	Description
<u>charAt()</u>	Returns the character at the specified index
<u>charCodeAt()</u>	Returns the Unicode of the character at the specified index
<u>concat()</u>	Joins two or more strings, and returns a copy of the joined strings
<u>fromCharCode()</u>	Converts Unicode values to characters
<u>indexOf()</u>	Returns the position of the first found occurrence of a specified value in a string
<u>lastIndexOf()</u>	Returns the position of the last found occurrence of a specified value in a string
<u>match()</u>	Searches for a match between a regular expression and a string, and returns the matches
<u>replace()</u>	Searches for a match between a substring (or regular expression) and a string, and replaces the matched substring with a new substring
<u>search()</u>	Searches for a match between a regular expression and a string, and returns the position of the match
<u>slice()</u>	Extracts a part of a string and returns a new string
<u>split()</u>	Splits a string into an array of substrings
<u>substr()</u>	Extracts the characters from a string, beginning at a specified start position, and through the specified number of character
<u>substring()</u>	Extracts the characters from a string, between two specified indices
<u>toLowerCase()</u>	Converts a string to lowercase letters
<u>toUpperCase()</u>	Converts a string to uppercase letters
<u>valueOf()</u>	Returns the primitive value of a String object

Παράδειγμα `indexOf`, `match`



- `var str="Hello world, welcome to the universe.";`
`var n=str.indexOf("welcome");`
- `n → 13`
- `var str="Hello world!";`
- `document.write(str.match("world") + "
");`
- output: world
- `document.write(str.match("World") + "
");`
- output: null

Παράδειγμα **replace**, **split**



- `str="Please visit Microsoft!"`
`var str1=str.replace("Microsoft","W3Schools");`
- `var monthString =`
`"Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec";`
`var comma = ",";`
`var arrayOfStrings = monthString.split(comma);`
- The array has 12 elements: Jan / Feb / Mar / Apr /
May / Jun / Jul / Aug / Sep / Oct / Nov / Dec

Αφαίρεση κενών από ένα string



```
var names = "Harry Trump ;Fred Barney; Helen  
Rigby ; Bill Abel ;Chris Hand ";
```

```
print(names);
```

```
var re = /\s*;\s*/;
```

```
var nameList = names.split(re);
```

```
print(nameList);
```

1° print: Harry Trump ;Fred Barney; Helen Rigby
; Bill Abel ;Chris Hand

2° print: Harry Trump,Fred Barney,Helen
Rigby,Bill Abel,Chris Hand

Παράδειγμα **slice**



```
var str1 = "The morning is upon us.";
```

```
var str2 = str1.slice(4, 11);
```

```
alert(str2); // OUTPUT: morning
```

```
var str2 = str1.slice(4, -2);
```

```
alert(str2); // OUTPUT: morning is upon u
```

Παράδειγμα **substr**



```
var str = "abcdefghij";
```

```
document.writeln("(1,2): " + str.substr(1,2)); // (1,2): bc
```

```
document.writeln("(-3,2): " + str.substr(-3,2)); // (-3,2): hi
```

// If start is negative, substr uses it as a character index from the end of the string.

```
document.writeln("(-3): " + str.substr(-3)); // (-3): hij
```

```
document.writeln("(1): " + str.substr(1)); // (1): bcdefghij
```

```
document.writeln("(-20, 2): " + str.substr(-20,2)); // (-20, 2): ab
```

```
document.writeln("(20, 2): " + str.substr(20,2)); // (20, 2):
```

Παράδειγμα **substring**



```
var anyString = "Mozilla";  
// Displays "Moz"  
document.writeln(anyString.substring(0,3));  
document.writeln(anyString.substring(3,0)); // If indexA is larger  
than indexB, then the effect of substring is as if the two  
arguments were swapped;  
// Displays "lla"  
document.writeln(anyString.substring(4,7));  
document.writeln(anyString.substring(7,4));  
// Displays "Mozill"  
document.writeln(anyString.substring(0,6));  
// Displays "Mozilla"  
document.writeln(anyString.substring(0,7));  
document.writeln(anyString.substring(0,10));
```

Συμβάντα (events) και χειριστές συμβάντων (event handlers) στη Javascript

Συμβάντα και χειριστές συμβάντων



- Τι είναι ένα **συμβάν** (event);
Κάτι που λαμβάνει χώρα όταν ο χρήστης εκτελεί μια ενέργεια
 - Κλικ σε ένα πλήκτρο
 - Αλλαγή των περιεχομένων ενός στοιχείου μιας φόρμας
 - Μετακίνηση του δείκτη του ποντικιού πάνω από ένα σύνδεσμο (link)
- **Χειριστής συμβάντος** (event handler):
 - Μια δεσμευμένη λέξη της Javascript που χρησιμοποιείται για το χειρισμό ενός συμβάντος, όταν αυτό λαμβάνει χώρα
 - Με τους χειριστές συμβάντων εισάγουμε στις σελίδες μας ένα βαθμό διαλογικότητας & διαδραστικότητας (μια σειρά από εντολές εκτελούνται ως αποτέλεσμα μιας ενέργειας του χρήστη)
- Που τοποθετούνται οι event handlers:
 - Σε στοιχεία φορμών
 - Σε ετικέτες δημιουργίας συνδέσμων (<a>)
 - Στην αρχική ετικέτα του HTML εγγράφου (<BODY>)

Χειριστές συμβάντων σε στοιχεία HTML φόρμες



`<html>` HTML φόρμα που ονομάζεται 'myform'

`<body>`

`<form name="myform">`

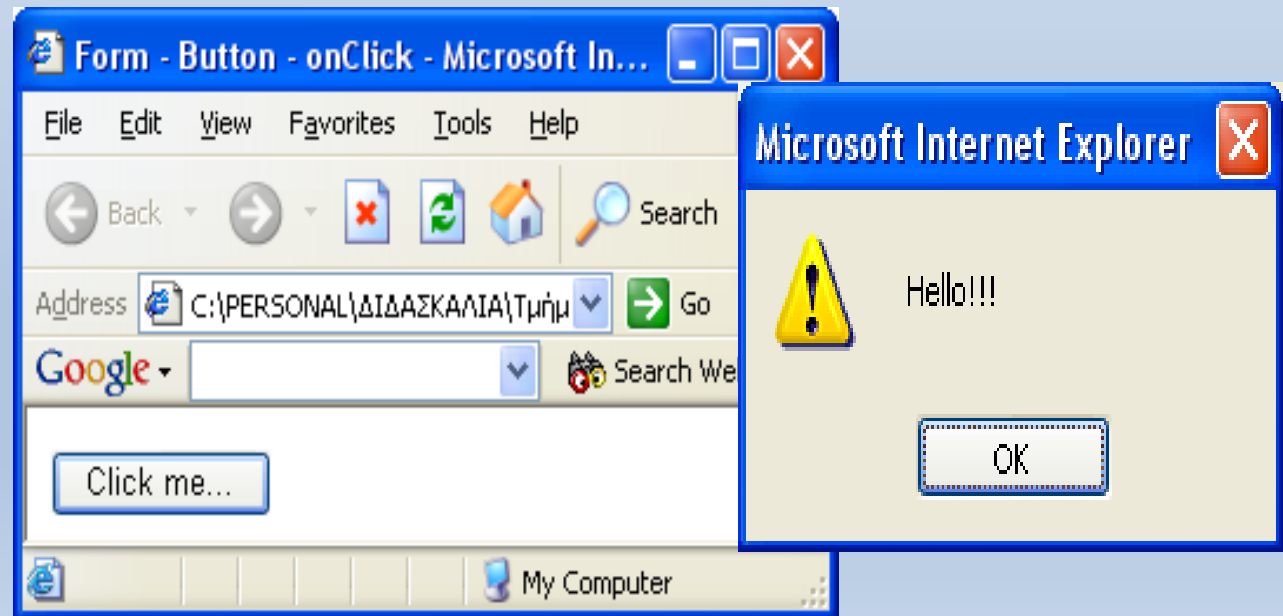
`<input type="button" value="Click me..." onClick="alert('Hello!!!');">`

`</form>`

`</body>`

`</html>`

Εισαγωγή στην HTML φόρμα ενός πλήκτρου που γράφει 'Click me'



Διαθέσιμοι χειριστές συμβάντων



Χειριστής συμβάντος	Ενεργοποιείται όταν ...
onClick	ο χρήστης κάνει κλικ σε ένα στοιχείο (πλήκτρο, σύνδεσμο, ...)
onmouseover	ο χρήστης τοποθετήσει τον δείκτη (cursor) του ποντικιού πάνω σε ένα στοιχείο
onmouseout	ο χρήστης απομακρύνει τον δείκτη του ποντικιού από ένα στοιχείο
onload	ολοκληρωθεί η «φόρτωση» (άνοιγμα) μιας ιστοσελίδας από τον browser
onunload	ο χρήστης «φύγει» από την τρέχουσα σελίδα

Διαθέσιμοι χειριστές συμβάντων



Χειριστής συμβάντος	Ενεργοποιείται όταν ...
onChange	ο χρήστης αλλάξει τα περιεχόμενα ενός στοιχείου μιας φόρμας (π.χ. πληκτρολογήσει κάτι σε ένα textfield ή επιλέξει άλλη επιλογή σε μια λίστα)
onSubmit	ο χρήστης υποβάλλει (στέιλει) μια φόρμα
onError	συμβεί κάποιο σφάλμα κατά τη φόρτωση της σελίδας ή κάποιας εικόνας
onResize	ο χρήστης αλλάξει το μέγεθος του παραθύρου

- Ακόμη: onDragDrop, onMove, onSelect, κ.α.

```
<a href="http://www.aegean.gr" onMouseOver="alert('Cursor over the link!');" onMouseOut="alert('Cursor out of link!');"> University of Aegean</a>
```

Διαθέσιμοι χειριστές συμβάντων



<code>onAbort</code>	<code></code>
<code>onChange</code>	<code><SELECT></code> , <code><INPUT TYPE=text file></code> , <code><TEXTAREA></code>
<code>onError</code> , <code>onLoad</code>	<code></code> , <code><BODY></code> , <code><FRAMESET></code>
<code>onMouseOut</code> , <code>onMouseOver</code>	<code><A HREF></code> , <code><AREA HREF></code>
<code>onReset</code> , <code>onSubmit</code>	<code><FORM></code>
<code>onSelect</code>	<code><INPUT TYPE=text></code> , <code><TEXTAREA></code>
<code>onUnload</code>	<code><BODY></code> , <code><FRAMESET></code>

Διαθέσιμοι χειριστές συμβάντων



<code>onBlur, onFocus</code>	<code><BODY>, <FRAMESET>, <INPUT TYPE=button checkbox file password radio reset submit text>, <TEXTAREA>, <SELECT></code>
<code>onClick</code>	<code><INPUT TYPE=button checkbox radio reset submit>, <A HREF></code>

Γεγονότα



Συμβάν	Πότε παράγεται	Ποιο control αφορά
onChange	Τροποποιείται κείμενο, φεύγει το focus	Select, text
onClick/Dblick	Γίνεται κλικ	Button, checkbox, link, radio
onError	Συμβαίνει λάθος κατά τη φόρτωση μιας εικόνας	Image
onFocus	Δέχεται το focus	Button, checkbox, radio, select, text
onKeyDown/Press/Up	Κάποιο πλήκτρο πιέζεται, αφήνεται, ..	Document, image, link, textarea
onLoad/Unload	Φορτώθηκε το έγγραφο ή μια εικόνα	Image, window
onMouseDown/Up/Over/Out	Πιέζεται, αφήνεται, μετακινείται ο mouse cursor	Button, document, link
onSubmit	Η φόρμα γίνεται submit	

Επικύρωση Εισόδου-Regular Expressions



```
<script type = "text/javascript">
function validate(){
name = document.getElementById("txtName").value;
email = document.getElementById("txtEmail").value;
phone = document.getElementById("txtPhone").value;

errors = "";
//check name - It simply needs to exist
if (name == ""){
errors += "please supply a name \n";
} // end if
//check email
emailRE = /^[^.\+@.\-]{2,4}$/;
if (email.match(emailRE)){

} else {
errors += "please check email address \n";
} // end if
//check phone number
phoneRE = /^(\d{3}\) *\d{3}-\d{4}$/;
if (phone.match(phoneRE)){
} else {
errors += "please check phone #\n";
} // end phone if
//check for errors
if (errors == ""){
alert ("now processing data");
//process the form
} else {
alert(errors);
} // end if
} // end function
```

Validation With Regular Expressions

Name	<input type="text"/>
Email	<input type="text" value="bobatnoplac.net"/>
Phone Number	<input type="text" value="(ddd) ddd-dddd"/>
	<input type="text" value="(000) 000-000"/>
	<input type="button" value="submit"/>



Operator	Description	Sample Pattern	Matches	Doesn't Match
<code>.</code> (period)	Any single character except newline	<code>.</code>	E	\n
<code>^</code>	Beginning of string	<code>^a</code>	Apple	Banana
<code>\$</code>	End of string	<code>a\$</code>	Banana	Apple
<code>[characters]</code>	Any of a list of characters in braces	<code>[abcABC]</code>	A	D
<code>[char range]</code>	Any character in the range	<code>[a-zA-Z]</code>	F	9
<code>\d</code>	Any single numerical digit	<code>\d\d\d-\d\d\d\d</code>	123-4567	The-thing
<code>\b</code>	A word boundary	<code>\bthe\b</code>	The	Theater
<code>+</code>	One or more occurrences of the previous character	<code>\d+</code>	1234	Text
<code>*</code>	Zero or more occurrences of the previous character	<code>[a-zA-Z]*</code> <code>d*</code>	B17, g	7
<code>{digit}</code>	Repeat preceding character <i>digit</i> times	<code>\d{3}-\d{4}</code>	123-4567	999-99-9999
<code>{min, max}</code>	Repeat preceding character at least <i>min</i> but not more than <i>max</i> times	<code>.{2,4}</code>	Ca, com, info	watermelon
<code>(pattern segment)</code>	Store results in pattern	<code>^(.)\.*\1\$</code>	gig, wallow	Bobby

Match αν ξεκινάει και τελειώνει με τον ίδιο χαρακτήρα

Document Object Model (DOM)

DOM - Document Object Model



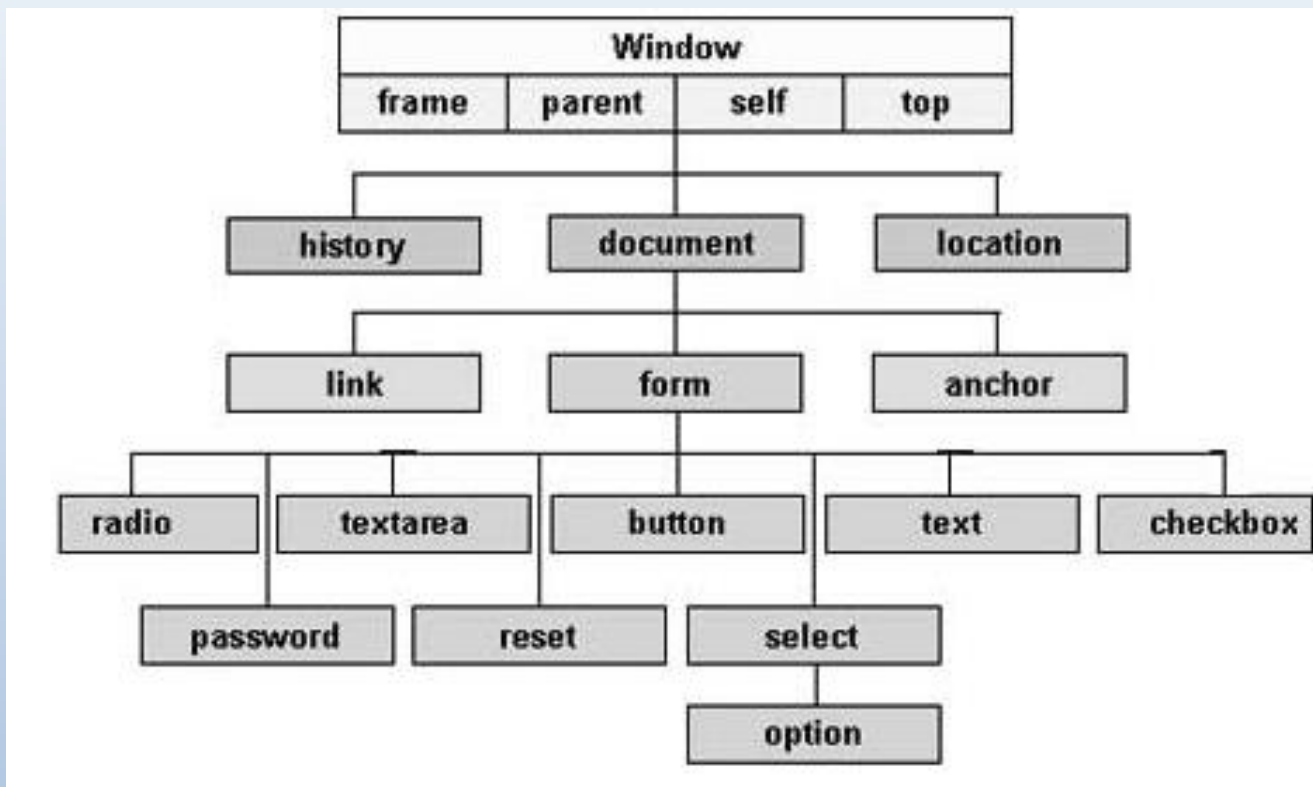
- Το DOM είναι μια πλατφόρμα και μια ανεξάρτητη από γλώσσα προγραμματισμού διεπαφή που επιτρέπει στα προγράμματα και τα scripts να έχουν πρόσβαση δυναμικά και να ενημερώνουν το περιεχόμενο, τη δομή, και το ύφος ενός εγγράφου.
- Το HTML DOC αποτελείται από μία δεντρική δομή όπου στοιχεία ενσωματώνονται σε άλλα στοιχεία.
- Τα στοιχεία αυτά μπορούν να προσπελαστούν από το DOM δέντρο.

DOM - Document Object Model



- Παρέχει ένα ιεραρχικό μοντέλο για την απεικόνιση των αντικειμένων της σελίδας.
- Επιτρέπει την πρόσβαση στα αντικείμενα καθώς και την αλλαγή των ιδιοτήτων τους.
- Ξεκινάει από ένα βασικό αντικείμενο “**window**” (το παράθυρο του browser), που περιέχει το αντικείμενο “document” (η σελίδα HTML).
- Κάτω από το αντικείμενο **document** τοποθετούνται σε ιεραρχική δομή όλα τα αντικείμενα της σελίδας.
- Κάθε αντικείμενο έχει ιδιότητες (όπως χρώμα, μέγεθος, θέση ...)
- Μπορούμε να αναφερθούμε στα αντικείμενα της σελίδας (κλαδιά), καλώντας κάθε αντικείμενο, από την ρίζα έως το κλαδί και χωρίζοντας τα ονόματά τους με τελείες.

Ιεραρχική δομή αντικειμένων DOM



Υπάρχουν διάφορα DOMs:

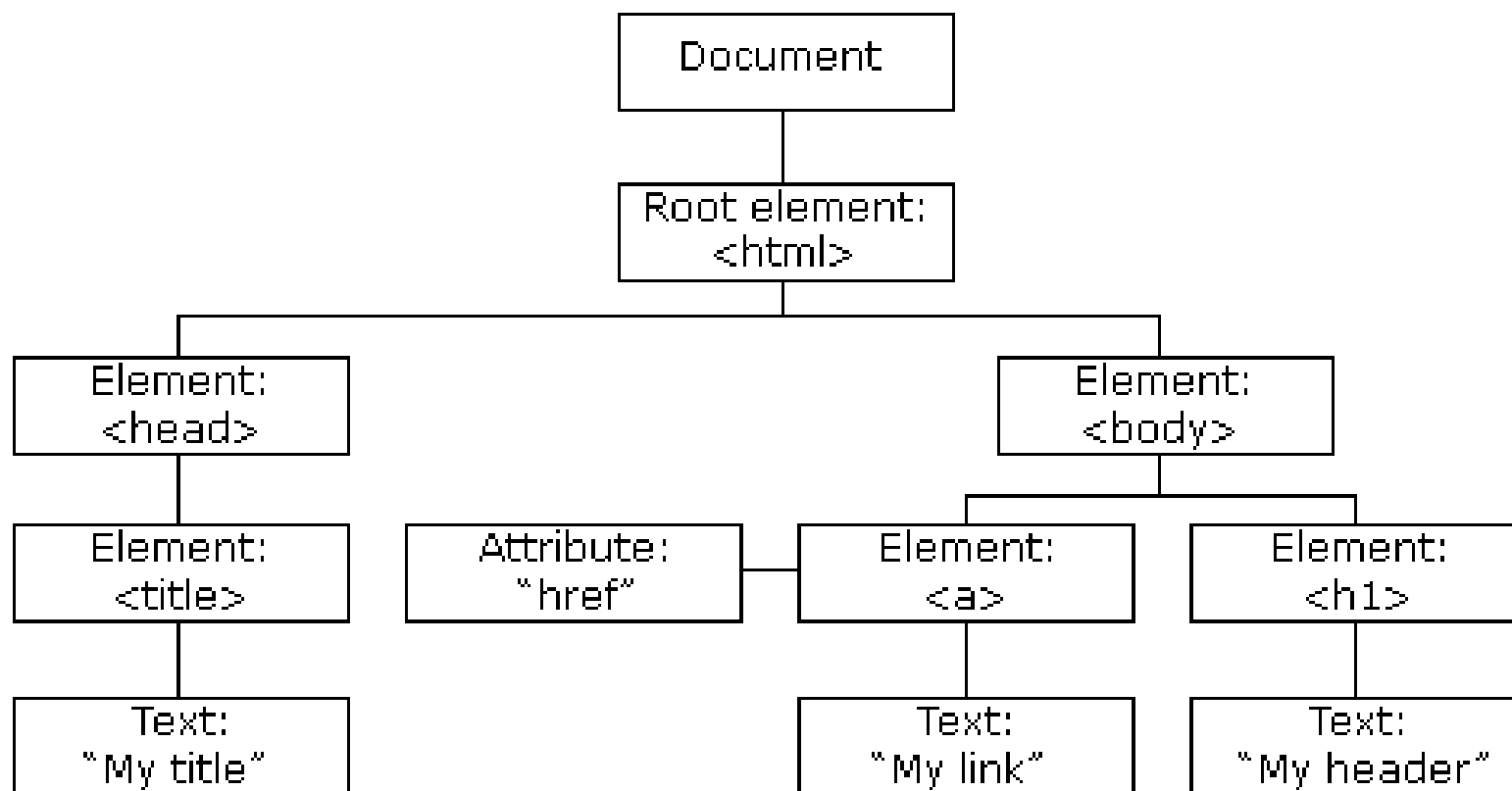
- Το Legacy DOM
- Το W3C DOM
- Το IE4 DOM

Παράδειγμα χρήσης DOM αντικειμένου



- Το αντικείμενο του παραθύρου (window) έχει ενδεικτικά:
 - **Ιδιότητες**: το url του εγγράφου (location) , το όνομα του παραθύρου (name), ένα μήνυμα που θα εμφανιστεί στη μπάρα κατάστασης του παραθύρου (status), το πλάτος (width), το ύψος (height) κ.α.
 - **μεθόδους** : open() και close().
 - Με την εντολή :
`window.open ("http://www.aegean.gr", "my_window")`
δημιουργείται και ανοίγει ένα στιγμιότυπο (instance) παραθύρου, που δείχνει στη διεύθυνση "http://www.aegean.gr" και έχει όνομα "my_window".

HTML DOM



Βασική προσπέλαση HTML στοιχείων (Legacy DOM)



Προσπέλαση με name ή id

- `document.form1`
- `document.form1.fname`
- `document.form1.fname.value`

Προσπέλαση με index σε πίνακα

- `document.form[1]`
- `document.form[1].element[1]`

Προσπέλαση με συνδυασμό των άνω

- `document.form[0].fname`
- `document.form1.element[3]`

Προσπέλαση HTML στοιχείων στο W3C DOM



- Μέθοδος `getElementById`
- Ιδιότητα `innerHTML`

Παράδειγμα

```
<html>
```

```
< body>
```

```
    < p id="intro">Hello World!</p>
```

```
< script>
```

```
    var txt=document.getElementById("intro").innerHTML;  
    document.write(txt);
```

```
< /script>
```

```
< /body>
```

```
< /html>
```

Παράδειγμα



```
<body>
<h1>Inner HTML Demo</h1>
<form action = "">
<fieldset>
<label>Please type your name</label>
<input type = "text"
id = "txtName" />
<button type = "button"
onclick = "sayHi()">
Click Me
</button>
</fieldset>
</form>
<div id = "divOutput">
Watch this space.
</div>
</body>
```

Inner HTML Demo

Please type your name

Nick is a very nice name.

```
<script type = "text/javascript">
//

function sayHi(){
txtName =
document.getElementById("txtName");
divOutput =
document.getElementById("divOutput");
name = txtName.value;
<divOutput.innerHTML = "&lt;em&gt;" + name
+ "&lt;/em&gt;";
divOutput.innerHTML += " is a very nice
name.";
}
//]]&gt;
&lt;/script&gt;</pre><p>Βαζουμε παντα ενα \ πριν απο καθε /</p></div>
```

HTML DOM Navigation

Κόμβοι (Nodes)



- Ένα HTML document αποτελείται από κόμβους διαφόρων τύπων.
- Οι κόμβοι μπορούν να περιέχουν άλλους κόμβους

```
<p style="special">This is text.</p>
```

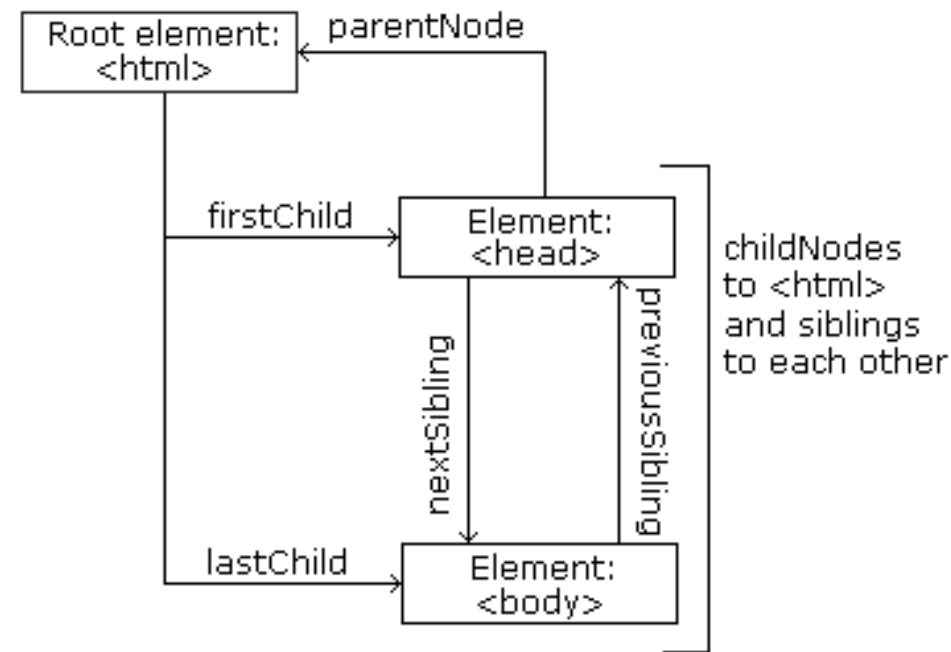
- **Element node**—HTML tag
- **Text node**— κείμενο στο document. Περιέχεται μέσα σε element nodes.
- **Attribute node**—παρέχει περισσότερες πληροφορίες για ένα element.

Συσχετίσεις κόμβων



```
<html>
  <head>
    <title>DOM Tutorial</title>
  </head>

  <body>
    <h1>DOM Lesson one</h1>
    <p>Hello world!</p>
  </body>
</html>
```



Πλοήγηση μεταξύ κόμβων



Χρησιμοποιούμε τις ιδιότητες ενός κόμβου:

- `parentNode`
- `childNodes[nodenum]`
- `firstChild`
- `lastChild`
- `nextSibling`
- `previousSibling`

Child Nodes and Node Values



```
<html>
< body>
< p id="intro">Hello World!</p>

< script>
var
txt=document.getElementById("intro").childNodes[0].nodeValue;
// Αντί για το στοιχείο childNodes[0] θα μπορούσαμε να
χρησιμοποιήσουμε την ιδιότητα firstChild
document.write(txt);
< /script>

< /body>
< /html>
```

Ιδιότητες nodeName , nodeValue, nodeType



- Η ιδιότητα **nodeName** προσδιορίζει το όνομα ενός κόμβου.
 - Το nodeName είναι read-only
 - Το nodeName για ένα element node είναι το ίδιο με το tag name
 - Το nodeName για ένα attribute node είναι το attribute name
 - Το nodeName για ένα text node είναι πάντα #text
 - Το nodeName για ένα document node είναι πάντα #document
- Η ιδιότητα **nodeValue** προσδιορίζει την τιμή ενός κόμβου.
 - Το nodeValue για element nodes είναι απροσδιόριστο
 - Το nodeValue για text nodes είναι το ίδιο το text
 - Το nodeValue για attribute nodes είναι το attribute value
- Η ιδιότητα **nodeType** επιστρέφει τον τύπο του κόμβου (είναι read only).
 - Οι πιο σημαντικοί τύποι είναι : Element (1), Attribute (2), Text (3), Comment (8), Document (9)

DOM Root Nodes



- `document.documentElement`
 - Όλο το document
- `document.body`
 - Το body του document

Το αντικείμενο window

Το αντικείμενο 'window'



- Το αντικείμενο **window** (παράθυρο) δημιουργείται για κάθε παράθυρο browser που εμφανίζεται στην οθόνη
- Το παράθυρο μπορεί να είναι κύριο παράθυρο της εφαρμογής, ένα παράθυρο με μια ομάδα πλαισίων ή ένα μεμονωμένο πλαίσιο ή ένα νέο παράθυρο που δημιουργήθηκε από κώδικα Javascript
- Το αντικείμενο window επιτρέπει την εμφάνιση μηνυμάτων, εισαγωγή πληροφοριών και επιβεβαίωση ενεργειών από το χρήστη, άνοιγμα νέων παραθύρων, κλπ
- Οι μέθοδοι **alert()** και **prompt()** του αντικειμένου window επιτρέπουν να εμφανίσουμε ένα μήνυμα ή να ζητήσουμε από το χρήστη να δώσει κάποια είσοδο, π.χ.
 - `window.alert("Hello World!!!")`
 - `var input = window.prompt("Insert text...", "");`

Ιδιότητες του αντικειμένου window



Ιδιότητα	Περιγραφή
name	Το όνομα του παραθύρου
closed	Δείχνει εάν ένα παράθυρο είναι κλειστό
opener	Το όνομα του παραθύρου που άνοιξε αυτό το παράθυρο
status	Κείμενο που θα εμφανιστεί στη μπάρα κατάστασης
length	Αριθμός πλαισίων (frames) που υπάρχουν στο παράθυρο
self	Αναφέρεται στο ίδιο το παράθυρο
parent	Αναφέρεται σε ένα παράθυρο που περιλαμβάνει πλαίσια

frames[], είναι ένας πίνακας που περιέχει όλα τα πλαίσια μιας ιστοσελίδας, όπου για παράδειγμα το `parent.frames[0]` είναι το πρώτο παράθυρο πλαίσιο κ.ο.κ.

Αναφορές σε άλλα αντικείμενα



- **document**, είναι το τρέχον έγγραφο ιστοσελίδας.
- **history**, περιέχει μια λίστα από τα URLs που έχουμε επισκεφθεί.
- **location**, περιέχει πληροφορίες για τη θέση (URL) του HTML εγγράφου που εμφανίζεται μέσα στο παράθυρο της ιστοσελίδας.
- **screen**, περιέχει πληροφορία για την οθόνη που απεικονίζει το παράθυρο.
- **navigator**, περιέχει πληροφορίες σχετικά με την έκδοση του φυλλομετρητή.

Μέθοδοι του αντικειμένου window



Μέθοδος	Περιγραφή
<code>alert()</code>	Εμφανίζει ένα μήνυμα στο χρήστη (συνήθως κάποια «προειδοποίηση») με ένα OK πλήκτρο
<code>prompt()</code>	Εμφανίζει ένα παράθυρο μέσω του οποίου ο χρήστης εισάγει κάποια είσοδο
<code>confirm()</code>	Εμφανίζει ένα μήνυμα επιβεβαίωσης (ο χρήστης πρέπει να πατήσει OK ή Cancel για να προχωρήσει)
<code>open()</code>	Ανοίγει ένα νέο παράθυρο του browser
<code>close()</code>	Κλείνει ένα νέο παράθυρο του browser

Μέθοδοι του αντικειμένου window



Μέθοδος	Περιγραφή
<code>setInterval()</code>	Καλεί μια συνάρτηση σε τακτά χρονικά διαστήματα (in ms)
<code>clearInterval()</code>	Ακυρώνει μια κλήση που έγινε μέσω της μεθόδου <code>setInterval()</code>
<code>setTimeout()</code>	Καλεί μια συνάρτηση μια φορά, αφότου περάσει κάποιο συγκεκριμένο χρονικό διάστημα σε milliseconds
<code>clearTimeout()</code>	Ακυρώνει μια κλήση που έγινε μέσω της μεθόδου <code>setTimeout()</code>

Μέθοδος confirm()



```
<script language="JavaScript">  
  var ok = window.confirm("Ονομάζεσαι Γιώργος;");  
  if (ok)  
    window.alert('Γεια σου Γιώργο!');  
  else  
    window.alert('Δεν ξέρω πως ονομάζεσαι...');  
</script>
```



Μέθοδοι setInterval()/clearInterval()

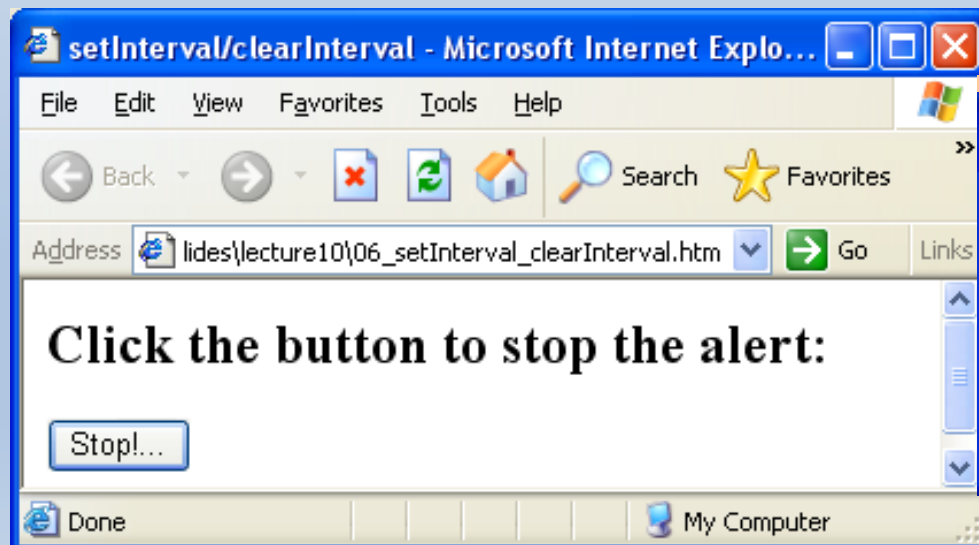


```
<body onLoad="iritation();">
```

```
<h2>Click the button to stop the alert: </h2>
```

```
<input type="button" value="Stop!..." onClick="stop_iritation();">
```

```
</body>
```



Μέθοδοι setInterval()/clearInterval()



```
<script language="JavaScript">
```

```
var repeat;
```

```
function show_message() {  
    window.alert ('Running for ever!!!');  
}
```

```
function irritation() {  
    repeat = window.setInterval("show_message()", 10000);  
}
```

```
function stop_iritation() {  
    window.clearInterval(repeat);  
    window.alert ('stopped!!!');  
}
```

```
</script>
```

Η συνάρτηση `show_message` θα καλείται **κάθε 10 sec** (σε msec)

Η κλήση της μεθόδου εκχωρείται σε μια μεταβλητή για μελλοντική αναφορά

Ακυρώνεται η περιοδική κλήση της μεθόδου

Μέθοδοι setTimeout()/clearTimeout()

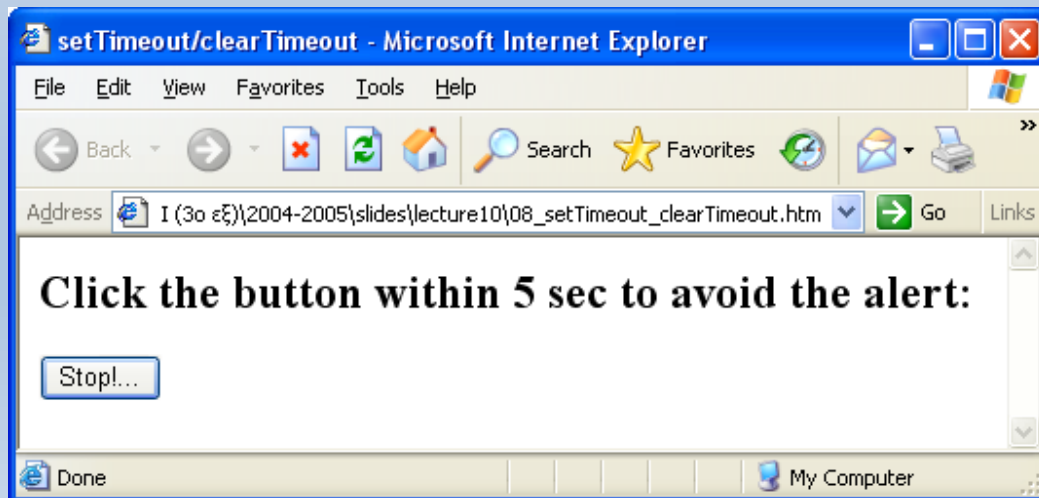


```
<body onLoad="iritation();">
```

```
<Click the button within 5 sec to avoid the alert: </h2>
```

```
<input type="button" value="Stop!..." onClick="stop_iritation();">
```

```
</body>
```



Μέθοδοι setTimeout()/clearTimeout()



```
<script language="JavaScript">
```

```
var repeat;
```

```
function irritation() {
```

```
    repeat = window.setTimeout("show_message()", 5000);
```

```
}
```

```
function show_message() {
```

```
    window.alert ('You are too slow!!!');
```

```
}
```

```
function stop_iritation() {
```

```
    window.clearTimeout(repeat);
```

```
}
```

```
</script>
```

Η μέθοδος `show_message` θα κληθεί **μία μόνο** φορά, μετά από **5 sec**

Ακυρώνεται η μοναδική κλήση της μεθόδου `show_message`

**Το αντικείμενο *element*
(*element object*)**

element object



- Στο HTML DOM, το **element object** αναπαριστά ένα HTML element.
- Τα element objects μπορούν να έχουν **child nodes** των τύπων: element nodes, text nodes, ή comment nodes.
- Ένα **NodeList object** αναπαριστά μια λίστα κόμβων
- Τα Elements μπορεί να έχουν επίσης attributes

Ιδιότητες



Property	Description
<u>element.attributes</u>	Returns a NamedNodeMap of an element's attributes
<u>element.childNodes</u>	Returns a NodeList of child nodes for an element
<u>element.className</u>	Sets or returns the class attribute of an element
<u>element.firstChild</u>	Returns the first child of an element
<u>element.id</u>	Sets or returns the id of an element
<u>element.innerHTML</u>	Sets or returns the content of an element
<u>element.lastChild</u>	Returns the last child of an element
<u>element.nodeName</u>	Returns the name of an element
<u>element.nodeType</u>	Returns the node type of an element
<u>element.nodeValue</u>	Sets or returns the value of an element
<u>element.parentNode</u>	Returns the parent node of an element
<u>element.previousSibling</u>	Returns the previous element at the same node tree level
<u>element.title</u>	Sets or returns the title attribute of an element

Μέθοδοι



Method	Description
<code>element.appendChild()</code>	Adds a new child node, to an element, as the last child node
<code>element.getAttribute()</code>	Returns the specified attribute value of an element node
<code>element.getAttributeNode()</code>	Returns the specified attribute node
<code>element.getElementsByTagName()</code>	Returns a collection of all child elements with the specified tagname
<code>element.hasAttribute()</code>	Returns true if an element has the specified attribute, otherwise false
<code>element.hasAttributes()</code>	Returns true if an element has any attributes, otherwise false
<code>element.hasChildNodes()</code>	Returns true if an element has any child nodes, otherwise false
<code>element.insertBefore()</code>	Inserts a new child node before a specified, existing, child node
<code>element.removeAttribute()</code>	Removes a specified attribute from an element
<code>element.removeAttributeNode()</code>	Removes a specified attribute node, and returns the removed node
<code>element.removeChild()</code>	Removes a child node from an element
<code>element.replaceChild()</code>	Replaces a child node in an element
<code>nodelist.item()</code>	Returns the node at the specified index in a NodeList

**Το αντικείμενο document
(document object)**

Ιδιότητες του αντικειμένου document



Ιδιότητες	Περιγραφή
bgColor	Το χρώμα που έχει το φόντο του εγγράφου
alinkColor	Το χρώμα των συνδέσμων
vlinkColor	Το χρώμα των συνδέσμων που έχει επισκεφθεί ένας χρήστης
referrer	Το όνομα του εγγράφου από το οποίο δημιουργήθηκε το έγγραφο
lastModified	Η ημερομηνία τελευταίας τροποποίησης του εγγράφου
title	Ο τίτλος του εγγράφου, δηλαδή το κείμενο της ετικέτας <TITLE> της HTML
URL	Η URL διεύθυνση της σελίδας HTML

Ιδιότητα bgColor (deprecated)



```
<script language="JavaScript">
function change_color (newColor) {
    document.bgColor = newColor;
}
</script>
```

```
<body>
<input type="button" value="Light Blue" onClick="change_color('lightblue')">
<input type="button" value="Orange" onClick="change_color('orange')">
<input type="button" value="Yellow" onClick="change_color('yellow')">
<input type="button" value="Green" onClick="change_color('green')">
</body>
```


Αντικείμενα παιδιά



- ***forms[]***, είναι ένας πίνακας που περιέχει όλες τις φόρμες που υπάρχουν μέσα στο έγγραφο.
- ***anchors[]***, είναι ένας πίνακας που περιέχει όλες τις συνδέσεις τύπου άγκυρας που υπάρχουν μέσα στο έγγραφο.
- ***images[]***, είναι ένας πίνακας που περιέχει όλες τις εικόνες που υπάρχουν μέσα στο έγγραφο.
- ***links[]***, είναι ένας πίνακας που περιέχει όλες τις συνδέσεις που υπάρχουν μέσα στο έγγραφο.

Μέθοδοι του αντικειμένου document



Μέθοδος	Περιγραφή
<code>open()</code>	Ανοίγει ένα νέο έγγραφο, του οποίου τα περιεχόμενα μπορείτε να δημιουργήσετε με τις εντολές <code>write()</code> και <code>writeln()</code>
<code>close()</code>	Κλείνει ένα έγγραφο που έχετε προηγουμένα ανοίξει με την εντολή <code>open()</code>
<code>write()</code>	Γράφει μια συμβολοσειρά σε ένα HTML έγγραφο
<code>writeln()</code>	Γράφει μια συμβολοσειρά (κείμενο) σε ένα HTML έγγραφο τερματίζοντας τη γραμμή με τον χαρακτήρα νέας γραμμής (αλλαγή γραμμής)

Εύρεση HTML Elements



Method	Description
<code>document.getElementById()</code>	Finding an element by element id
<code>document.getElementsByTagName()</code>	Finding elements by tag name
<code>document.getElementsByClassName()</code>	Finding elements by class name
<code>document.forms[]</code>	Finding elements by HTML element objects

Παραδείγματα



Finding HTML Elements by Id

```
var x=document.getElementById("intro");
```

Finding HTML Elements by Tag Name

```
var x=document.getElementById("main");  
var y=x.getElementsByTagName("p");
```

Finding HTML Elements by Class Name

```
document.getElementsByClassName("intro");
```

Finding HTML Elements by HTML Object Collections

```
var x = document.getElementById("frm1");  
var txt = "";  
for (var i=0; i<x.length; i++) {  
    txt = txt + x.elements[i].value + "<br>";  
}  
document.getElementById("demo").innerHTML=txt;
```

HTML DOM Nodelist



- Η μέθοδος `getElementsByTagName()` επιστρέφει ένα **node list**.
- Ένα `nodelist` είναι ένας πίνακας από κόμβους

Παράδειγμα

```
var x=document.getElementsByTagName("p");  
for (i=0;i<x.length;i++)  
{  
    document.write(x[i].innerHTML);  
    document.write("<br />");  
}
```

Μεταβολή HTML Elements



Method	Description
<i>element.innerHTML=</i>	Changing the inner HTML of an element
<i>element.attribute=</i>	Changing the attribute of an HTML element
<i>element.setAttribute(attribute, value)</i>	Changing the attribute of an HTML element
<i>element.style.property=</i>	Changing the style of an HTML element

Παραδείγματα



Changing HTML Content

```
<body>
< h1 id="header">Old Header</h1>
< script>
var element=document.getElementById("header");
element.innerHTML="New Header";
< /script>
< /body>
```

Changing the Value of an Attribute

```
<body>
< img id="image" src="smiley.gif">
< script>
document.getElementById("image").src="landscape.jpg";
< /script>
< /body>
```

Παραδείγματα



Changing HTML Style

```
<body>  
< p id="p2">Hello World!</p>  
< script>  
document.getElementById("p2").style.color="blue";  
< /script>  
< p>The paragraph above was changed by a script.</p>  
< /body>
```


Προσθήκη και Διαγραφή Elements



Method	Description
<code>document.createElement()</code>	Create an HTML element
<code>document.removeChild()</code>	Remove an HTML element
<code>document.appendChild()</code>	Add an HTML element
<code>document.replaceChild()</code>	replace an HTML element
<code>document.write(<i>text</i>)</code>	Writing into the HTML output stream



Creating New HTML Elements (Nodes)

```
<div id="div1">  
< p id="p1">This is a paragraph.</p>  
< p id="p2">This is another paragraph.</p>  
< /div>  
<script>  
var para=document.createElement("p");  
var node=document.createTextNode("This is new.");  
para.appendChild(node);  
  
var element=document.getElementById("div1");  
element.appendChild(para);  
< /script>
```



Removing Existing HTML Elements

```
<div id="div1">  
< p id="p1">This is a paragraph.</p>  
< p id="p2">This is another paragraph.</p>  
< /div>
```

```
<script>  
var parent=document.getElementById("div1");  
var child=document.getElementById("p1");  
parent.removeChild(child);  
< /script>
```

Προσθήκη Event Handlers



Method	Description
<code>document.getElementById(id).onclick = function(){code}</code>	Adding event handler code to an onclick event

```
<!DOCTYPE html>
<html>
<body>
  <h1 id="id1">My Heading 1</h1>
  <button type="button"
    onclick="document.getElementById('id1').style.color='red'">
Click Me!</button>
</body>
</html>
```

Άλλα αντικείμενα

Navigator & Event objects



- Δύο άλλα αντικείμενα που υποστηρίζει το DOM:
 - “navigator” το οποίο περιλαμβάνει πληροφορίες για το φυλλομετρητή
 - “event” περιλαμβάνει πληροφορίες για τα γεγονότα που συμβαίνουν στη σελίδα HTML.

Αντικείμενο Navigator



```
var x = navigator  
document.write("CodeName=" + x.appCodeName)  
document.write("MinorVersion=" + x.appMinorVersion)  
document.write("Name=" + x.appName)  
document.write("Version=" + x.appVersion)  
document.write("CookieEnabled=" + x.cookieEnabled)  
document.write("CPUClass=" + x.cpuClass)  
document.write("OnLine=" + x.onLine)  
document.write("Platform=" + x.platform)  
document.write("UA=" + x.userAgent)  
document.write("BrowserLanguage=" + x.browserLanguage)  
document.write("SystemLanguage=" + x.systemLanguage)  
document.write("UserLanguage=" + x.userLanguage)
```

Αντικείμενο Navigator



The screenshot shows a Microsoft Internet Explorer browser window. The address bar displays the URL `http--logic.pdmi.ras.ru-cs...`. The browser's status bar at the bottom indicates a zoom level of 100%. The main content area displays the following properties of the Navigator object:

```
CodeName=Mozilla
MinorVersion=0
Name=Microsoft Internet Explorer
Version=5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0; Touch; .NET4.0E; .NET4.0C; .NET CLR 3.5.30729; .NET CLR 2.0.50727; .NET CLR 3.0.30729; Tablet PC 2.0; CMNTDFJS; InfoPath.3)
CookieEnabled=true
CPUClass=x86
OnLine=true
Platform=Win32
UA=Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0; Touch; .NET4.0E; .NET4.0C; .NET CLR 3.5.30729; .NET CLR 2.0.50727; .NET CLR 3.0.30729; Tablet PC 2.0; CMNTDFJS; InfoPath.3)
BrowserLanguage=en-US
SystemLanguage=el-GR
UserLanguage=en-US
```


Αντικείμενο Event



```
<html><head>
```

```
<script type="text/javascript">
```

```
function whichType(event)
```

```
{
```

```
  alert(event.type)
```

```
}
```

```
</script></head>
```

```
<body onload="whichType(event)" onmousedown="whichType(event)"  
onkeypress="whichType(event)" onresize="whichType(event)"  
onunload="whichType(event)">
```

```
<p>Click on the document. An alert box will alert which type of event occurred.</p>
```

```
</body></html>
```

Dynamic HTML - DHTML

Μανώλης Μαραγκουδάκης

Πανεπιστήμιο Αιγαίου
Τμήμα Μηχανικών Πληροφοριακών και
Επικοινωνιακών Συστημάτων



- Σκοπός της οι σελίδες να είναι δυναμικές
- Συνδυασμός από HTML, CSS και JavaScript
 - Χρησιμοποιεί τα objects της JavaScript
 - Κυρίως στηρίζεται στο Document Object Model
- Οριοθέτηση (positioning)
- Μεταβολή style
- Χειρισμός γεγονότων (event handling)

Χρήση του DOM



```
<div id="mydiv" style="position:absolute;
  visibility:visible; left:400px; top:100px;">
  <a href="www.icsd.aegean.gr">ICSD</a>
</div>
```

- Άμεσα προσπελάσιμα ως objects από την JavaScript μέσω του DOM
- `document.getElementById("mydiv").style.visibility="hidden";`
- Μέσω JavaScript μπορούμε να αλλάξουμε τις ιδιότητες ενός style και να δούμε τις αλλαγές αυτόματα

Δυναμική οριοθέτηση



```
<p id = "pText" style = "position: absolute; left: 0; color: blue"> Welcome! </p>
```

```
_pt = document.getElementById("mydiv");
```

```
_pt.style.fontSize = myvar/3;
```

```
_pt.style.left = myvar;
```

```
_pt.innerHTML = "<BR> Font size: " + myvar + "px";
```

Κρύψιμο Αντικειμένων



- `Style="visibility: {visible; hidden; inherit}";`
- `Style="display: {inline; block; none;}";`

Ποια η διαφορά ανάμεσα στις δύο ιδιότητες;

Μεταβολή style



- Επιλογή φόντου

```
<BODY bgcolor="red">
```

```
inpCol = prompt( «Select background color", "" );
```

```
document.body.style.backgroundColor = inpCol;
```

- Επιλογή μεγέθους κειμένου

```
pText.className = inputClass;
```

```
<style>
```

```
.bigText { font-size: 3em; font-weight: bold }
```

```
.smallText { font-size: .5em }
```

```
</style>
```