

Προγραμματισμός Ιστού

Javascript

# JavaScript

- Είναι μία scripting language
    - «Ελαφριά» γλώσσα προγραμματισμού
  - Μπορεί να προστεθεί σε ένα HTML αρχείο
    - «Εκτελείται» από τον Internet browser
  - Έχει ομοιότητες με τη Java
  - Δημιουργήθηκε από τη Netscape
    - Την υποστηρίζει και ο MS Internet Explorer (3.0 ...)
- ! Πρόκληση: κώδικας που να υποστηρίζεται από τις περισσότερες εκδόσεις

# Χρήση της JavaScript

- Μικρά κομμάτια κώδικα σε ένα HTML αρχείο
  - π.χ. εμφάνιση σειράς αριθμών (1, 2, ..., 100)
- Εμφάνιση «δυναμικού» περιεχομένου
- Αντιλαμβάνεται και αντιδρά σε γεγονότα – αλληλεπίδραση με τον χρήστη
  - π.χ. ο χρήστης επιλέγει ένα link
- Αλλάζει τα περιεχόμενα σε html elements
- Ελέγχει δεδομένα που δίνονται σε μία φόρμα πριν αυτή γίνει submit

# Εισαγωγή scripting language

- JavaScript

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

[View page](#)

# Τοποθέτηση scripting language

## • Head section

- Η ετικέτα head φορτώνεται πρώτα
- Για scripts που εκτελούνται μετά την κλήση τους
- Κατάλληλο για συναρτήσεις

## • Body Section

- Εκτελούνται όταν η σελίδα φορτώνεται

## • External script

```
<script language="JavaScript" src="global.js"></script>
```

! Επιτρέπονται πολλαπλά «scripts» σε ένα αρχείο

# Μεταβλητές

## JavaScript

- Ξεκινούν με γράμμα ή “\_”
- Case sensitive
- Ορισμός  
`var name` (*var optional*)
- Ανάθεση τιμής  
`name = “Alex”`  
`mynumber = 15`
- Τοποθέτηση σε κείμενο  
`"My name is " + name`

[View page](#)

# Μεταβλητές - τύποι δεδομένων

- Αριθμοί : ακέραιοι ή κινητής υποδιαστολής
  - Boolean : true / false (=0)
    - true :  $\neq 0$
  - Strings : ανάμεσα σε μονά ή διπλά εισαγωγικά
  - Αντικείμενα (objects)
  - Null : διαφορετικός από τη μηδενική τιμή
    - Σημαίνει ότι ΔΕΝ έχει τιμή
  - Undefined / Empty : αμέσως μετά τη δημιουργία
- ! Δεν είναι απαραίτητο να δηλώνεται ο τύπος πριν την ανάθεση τιμής

# Τελεστές

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



# If ... else

- JavaScript

```
if (condition)
```

```
{ statements1;      }
```

```
else
```

```
{ statements2;      }
```

! else if

```
varname = (condition)?trueVal:falseVal
```

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

[View page](#)

# Switch --- Select

- JavaScript

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

[View page](#)

# For

- JavaScript

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

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

[View page](#)

# Do ... while

- JavaScript

```
while (condition)  
{ statements }
```

```
do  
{ statements; }  
while (condition)
```

```
do  
{ document.write(i + "<br>")  
  i=i+1 }  
while (i < 10)
```

[View page](#)

# Έξοδος από βρόχο ... σχόλια

- Break :
  - εγκαταλείπει την εκτέλεση του βρόχου
- Continue :
  - εγκαταλείπει την απλή επανάληψη του βρόχου
- Σχόλια
  - //
  - /\* ... \*/

[View page](#)

# 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

- JavaScript

noarg()

print("Hello")

mySum = sum(23, 7)

alert("This is alert")

[View page](#)

# JavaScript Objects

- Υπάρχουν δύο τύποι αντικειμένων
  - Εγγενή της γλώσσας
    - πχ. Math, Date, String, Array
  - Συγκεκριμένα για μία εφαρμογή (πχ. Browser)
    - πχ. Window, Document, Form
- Όλα τα αντικείμενα έχουν
  - Ιδιότητες : `objectName.propertyName`
  - Μεθόδους : `objectName.methodName()`
  - ! Case sensitive



# Objects or Classes

- Objects : ξεκινούν με μικρό γράμμα
  - Δημιουργούνται με τη χρήση του new
  - πχ. `var myDate = new Date`
- Classes : ξεκινούν με κεφαλαίο γράμμα
  - Δεν χρειάζεται να οριστούν
  - exist by default
  - πχ. `document`

# 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<sup>th</sup> μέχρι και το b-1<sup>th</sup> (η αρίθμηση από το 0)

[View page](#)

# Math object

- Ιδιότητες
  - E, LN2, LN10, PI, SQRT2, ...
- Μέθοδοι
  - `abs(x)`, `cos(x)`, `sin(x)`, `ceil(x)`, `floor(x)`, `max(x,y)`, `min(x,y)`, `random()`
    - `random()` : ένα τυχαίο αριθμό μεταξύ του 0 και 1

# Date object

- Μέθοδοι
  - Date() : επιστρέφει ένα αντικείμενο Date
  - getDate() : επιστρέφει τη μέρα (1-31)
    - getMonth(), getDay(), getYear()
  - getHours(), getTime()
  - setDate(), setMonth(), setHours()
  - toString() : επιστρέφει την ημερομηνία ως string

[View page](#)

# Window object

- `alert("Hello Students")`
- `prompt("What is your name?", "")`
  - `prompt("question", "given_answer")`
- `confirm("Do you agree?")`
  - Select "Ok" or "Cancel"
- `location="string.html"`
- `window.status = "Your message"`
- `window.open("window2.html")`
  - A new window is opened
- `onclick()`

[View page](#)

# Form object

```
<form name="myForm">
```

```
  <input type="text" name="myAge">
```

- document.forms[0].myAge.focus()
- document.myForm.myAge.focus()
- document.myForm.myAge.value
- onsubmit()

[View page](#)

# DHTML

- Σκοπός της οι σελίδες να είναι δυναμικές
- Συνδυασμός από HTML, CSS και JavaScript
  - Χρησιμοποιεί τα objects της JavaScript
  - Κυρίως στηρίζεται στο Document Object Model
    - Διαφορετικό DOM υποστηρίζουν η Netscape και η Microsoft
- 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.ceid.upatras.gr">CEID</a>
```

```
</div>
```

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



# Δυναμική οριοθέτηση

```
<p id = "_pt" 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";
```

[View page](#)

# Κρύψιμο Αντικειμένων

- Style=“visibility: {visible; hidden; inherit}”;

[View page](#)

- Style=“display: {inline; block; none;}”;

[View page](#)

Ποια η διαφορά ανάμεσα στις δύο ιδιότητες;

# Μεταβολή style

- Επιλογή φόντου

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

```
inpCol = prompt( «Select background color", "" );  
document.body.style.backgroundColor = inpCol;
```

[View page](#)

- Επιλογή μεγέθους κειμένου

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

```
<style>
```

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

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

```
</style>
```

[View page](#)

# Χειρισμός γεγονότων

- Γεγονότα
  - πάτημα κουμπιού : **click**
  - ποντίκι πάνω σε κάποιο στοιχείο : **mouseover**
  - αποστολή δεδομένων φόρμας : **submit**
  - ολοκλήρωση του φορτώματος όλων των αντικειμένων (όχι παρέμβαση του χρήστη) : **load**
- Handler : eventName
  - π.χ. onClick, onMouseOver, onSubmit, onLoad

# Παραδείγματα

- [onClick](#)

```
<input type= "button" value= "Click Me!"
```

```
onClick= "alert( 'Hi again' )">
```

```
onClick="myFunction()"
```

- [onMouseOver / onMouseOut](#)

```
function mOver() {
```

```
if ( event.srcElement.id )
```

```
    event.srcElement.style.color = event.srcElement.id; }
```

```
function mOut() {
```

```
if ( event.srcElement.id )
```

```
    event.srcElement.innerText = event.srcElement.id; }
```

- [onLoad](#)

```
<body onLoad= "startTimer()">
```

# Τέλος

Ευχαριστώ πολύ