

Προγραμματισμός Παγκόσμιου Ιστού

Προγραμματισμός με JAVASCRIPT και Document Object Model

Ορέστης Τελέλης

telelis@unipi.gr

Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς

Σε αυτό το μάθημα

- JAVASCRIPT στο πλαίσιο χρήσης της.
- Document Object Model.
- Το αντικείμενο document.
- JAVASCRIPT Events.
- Το αντικείμενο window.

Σύνοψη Προηγούμενου

- Εισαγωγικά για τη JAVASCRIPT
- Συστατικά της Γλώσσας – Χρήση σε Έγγραφα HTML
- Συναρτήσεις
- Αντικείμενα
- Προεπισκόπηση Αλληλεπίδρασης με HTML/Css
(παρουσίαση του παραδείγματος RollDice.html)

Αλληλεπίδραση με HTML

Βασικές Λειτουργίες

- Ανάκτηση κάθε στοιχείου (element) HTML:
 - ▶ ως αντικείμενου JAVASCRIPT.
 - ▶ ανάγνωση/τροποποίηση των χαρακτηριστικών του (attributes).
 - ▶ ανάγνωση/τροποποίηση του Css του.
- Ανάκτηση/Τροποποίηση χαρακτηριστικών που αφορούν σε όλο το έγγραφο.
- Απόκριση σε συμβάντα (events).
- Ορισμός και ανάγνωση cookies.

Τροποποίηση με write/writeln

- Μέθοδοι του αντικειμένου document.
 - ▶ Ορίζεται από τον browser για τη μοντελοποίηση του εγγράφου.
- Μπορούν να χρησιμοποιηθούν εντός του tag `<script>`.
`<script>document.writeln("<h1>Hello World!</h1>");</script>`
- **Προσοχή:** Δεν δουλεύουν όταν όλο το έγγραφο έχει φορτωθεί!
- Αν π.χ. είναι τοποθετημένες σε εξωτερικό script που φορτώνεται με:
`<script src="script.js"defer></script>`
- Τροποποίηση του εγγράφου με write/writeln απαιτεί:
 - ▶ είτε «Ρητή» (inline) συμπερίληψή τους εντός `<script>`,
 - ▶ είτε blocking `<script>` εφόσον πρόκειται για εξωτερικό αρχείο.

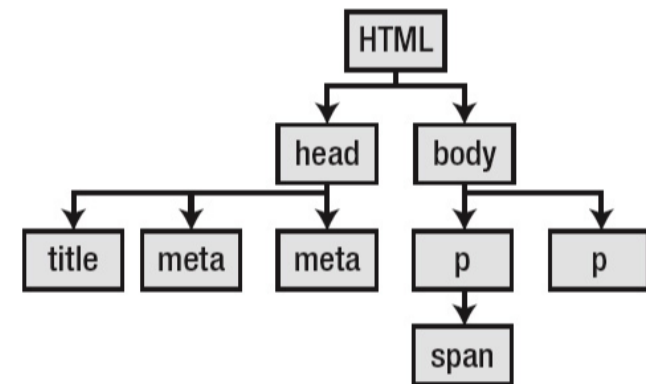
Document Object Model

- Επιτρέπει τη χρήση JAVASCRIPT για την:
 - ▶ «εξερεύνηση» των περιεχομένων ενός εγγράφου HTML.
 - ▶ διαχείριση περιεχομένων εγγράφου HTML (ανάγνωση, τροποποίηση των χαρακτηριστικών των στοιχείων HTML).
 - ▶ **δυναμική απόκριση** του εγγράφου σε ενέργειες του χρήστη.
- Είναι ένα **αντικειμενοστρεφές μοντέλο** του εγγράφου HTML.
- Ο browser το χτίζει κατόπιν ανάγνωσης του εγγράφου.

Παράδειγμα 1

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Simple Example</title>
    <meta charset="UTF-8">
    <meta name="description" content="fruit-stuff">
  </head>
  <body>
    <p id="fruittext">
      There are lots of different kinds of fruit - there are
      over 500 varieties of <span id="banana">banana</span>
      alone. By the time we add the countless types of apples,
      oranges, and other well-known fruit, we are faced with
      thousands of choices.</p>
    <p id="apples">
      One of the most interesting aspects of fruit is the
      variety available in each country. I live near London,
      in an area which is known for its apples.</p>
  </body>
</html>
```

Παράδειγμα DOM



Αλληλεπίδραση με το document

Μερικές ιδιότητες πρόσβασης στα στοιχεία HTML του εγγράφου:

document.documentElement	όλο το στοιχείο <code><html></code>
document.body	όλο το στοιχείο <code><body></code>
document.myElm	στοιχείο με name attribute <code>"myElm"</code>
document.getElementById("myElm")	στοιχείο με id attribute <code>"myElm"</code>
document.getElementsByTagName("tag")	array HTML στοιχείων <code><tag>...</tag></code>

Επιστρέφουν όλες αντικείμενα (ή απαγ αντικειμένων):

- Κάθε τέτοιο αντικείμενο μοντελοποιεί το (τα) στοιχείο(-α) HTML.
- Η μοντελοποίηση παρέχει πρόσβαση σε όλα τα attributes των στοιχείων.

Αντικείμενο document

- Το πιο βασικό αντικείμενο του DOM.
- Μοντελοποιεί όλο το έγγραφο HTML.
- **Τύπου** Document.
- Προσπελαύνεται από την **αναφορά** document.

```
<script>
  document.writeln("<pre>URL: " + document.URL);
  var elems = document.getElementsByTagName("p");
  for (var i = 0; i < elems.length; i++) {
    document.writeln("Element ID: " + elems[i].id);
    elems[i].style.border = "medium double black";
    elems[i].style.padding = "4px";
  }
  document.write("</pre>");
</script>
```

Παράδειγμα 1

```
<body>
  <p id="fruittext"><!-- ... -->
    <span id="banana">banana</span>
  <!-- ... --></p>

  <p id="apples"><!-- ... --></p>

  <script>
    document.writeln("<pre>URL: " + document.URL);
    var elems = document.getElementsByTagName("p");
    for (var i = 0; i < elems.length; i++) {
      document.writeln("Element ID: " + elems[i].id);
      elems[i].style.border = "medium double black";
      elems[i].style.padding = "4px";
    }
    document.write("</pre>");
  </script>
</body>
```

Επεξεργασία των Attributes Στοιχείων HTML

Κάθε attribute είναι ιδιότητα του αντικειμένου που αναπαριστά το στοιχείο HTML

Παράδειγμα:

```
<a id = "myLink" href = "http://www.unipi.gr" title="UNIPI">
  Press here
</a>
```

Ο ακόλουθος κώδικας JAVASCRIPT μεταβάλλει τα attributes του παραπάνω στοιχείου <a>:

```
<script>

  var aLink = document.getElementById("myLink");

  aLink.href = "http://www.ds.unipi.gr";

  aLink.title = "DS.UNIPI";

  aLink.innerHTML = "Please press here!";

</script>
```

- Με την ιδιότητα **style** των αντικειμένων που αναπαριστούν στοιχεία HTML.
- Η ιδιότητα **style** είναι με τη σειρά της αναφορά σε αντικείμενο:
 - ▶ με ιδιότητες όλες τις ιδιότητες Css που μπορούν να εφαρμοστούν
 - ▶ καθεμία έχει όνομα ίδιο με αυτό της ιδιότητας Css,
 - ▶ εκτός από αυτές με παύλα «-» στο όνομά τους:
 - ▶ σε αυτές παραλείπεται η παύλα και το 2ο τμήμα ξεκινά με κεφαλαίο.

• Παραδείγματα:

```
document.body.style.color = "red";  
document.body.style.backgroundColor = "white";
```

Προσφέρεται η δυνατότητα ορισμού «διαχειριστών συμβάντων» (event handlers)

- Κώδικας JAVASCRIPT που θα εκτελεστεί όταν συμβεί το event.
- Μπορεί να είναι μερικά JAVASCRIPT statements,
- ή συνάρτηση JAVASCRIPT που εκτελεί συγκεκριμένη λειτουργία.
 - ▶ Π.χ. έλεγχος ορθής εισόδου σε φόρμα (form validation).
- Απλούστερος τρόπος ορισμού:
 - ▶ Με ανάθεση κώδικα JAVASCRIPT σε attribute στοιχείου HTML.
 - ▶ Attribute: **on<eventName>** όπου **<eventName>** το όνομα του event.
- **Παράδειγμα:** `<div onmouseover="alert('Hello')"></div>`

Μοντελοποίηση Αλληλεπίδρασης του Χρήστη με το Έγγραφο

- Μετακίνηση του mouse pointer πάνω από στοιχείο HTML.
- Επιλογή στοιχείου HTML με το mouse (click/double click).
- Επιλογή ενός συνδέσμου με το mouse.
- Πίεση ενός πλήκτρου / ελευθέρωση πλήκτρου.
- και πολλά άλλα ακόμα

Ειδικά Events για Φόρμες

- Αλλαγή περιεχομένων φόρμας.
- Αποστολή στοιχείων φόρμας.
- Επαναφορά κατάστασης φόρμας.
- Επιλογή στοιχείου φόρμας.

Mouse Events

- onclick
- ondblclick
- onmousedown
- onmousemove
- onmouseout
- onmouseover
- onmouseup

Keyboard Events

- onkeydown
- onkeypress
- onkeyup

Window Events

- onload
- onunload

Form Events

- onchange
- onsubmit
- onreset
- onselect
- onblur
- onfocus

Διαχείριση «Συμβάντων» (Events)

- Υπάρχουν διαφορετικοί τρόποι διαχείρισης.
- Πιο απλά με δημιουργία ενός «διαχειριστή συμβάντος» (event handler):
 - ▶ με χρήση ενός event attribute εντός ενός HTML tag.
- Τα tags ορίζουν ένα event attribute για κάθε event που υποστηρίζουν.
- Π.χ. το **onmouseover** είναι ένα event attribute για το mouseover event.
 - ▶ ενεργοποιείται όταν «περάσει» ο mouse pointer πάνω από το στοιχείο HTML.
- Για event με όνομα **<name>**, το HTML attribute έχει όνομα **on<name>**.
- Ορισμός διαχειριστή με «ανάθεση» κώδικα JS (σαν συμβολοσειρά) στο event attribute.

Παράδειγμα 13: «Inline» Διαχείριση Event

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Example</title>
    <style type="text/css">
      p {
        background: gray;
        color:white;
        padding: 10px;
        margin: 5px;
        border: thin solid black
      }
    </style>
  </head>
  <body>
    <!-- document body follows -->
```

Παράδειγμα 13: «Inline» Διαχείριση Event

```
<body>
  <p onmouseover="this.style.background='white';\
    this.style.color='black'"
    onmouseout="this.style.background='gray';\
    this.style.color='white';">
    There are lots of different kinds of fruit - there are over
    500 varieties of banana alone. By the time we add the
    countless types of apples, oranges, and other well-known
    fruit, we are faced with thousands of choices.
  </p>
</body>
</html>
```

Παράδειγμα 14: Διαχείριση Event με Συνάρτηση

```
<p onmouseover="handleMouseOver(this)"
  onmouseout="handleMouseOut(this)">
  <!-- ... .. -->
</p>

<script type="text/javascript">

  function handleMouseOver(elem) {
    elem.style.background='white';
    elem.style.color='black';
  }

  function handleMouseOut(elem) {
    elem.style.background='gray';
    elem.style.color='white';
  }

</script>
```

Παράδειγμα 15: Χρήση του Αντικειμένου Event 1/2

```
<script type="text/javascript">
// get all <p> elements
var pElems = document.getElementsByTagName("p");

for (var i = 0; i < pElems.length; i++) {
  // Assign name of handling function not a call to it!
  pElems[i].onmouseover = handleMouseOver;
  pElems[i].onmouseout = handleMouseOut;
}
// function argument "e" is of type Event
function handleMouseOver(e) {
  // e.target is the element receiving the event
  e.target.style.background='white';
  e.target.style.color='black';
}
function handleMouseOut(e) {
  e.target.style.background='gray';
  e.target.style.color='white';
}
</script>
```

Παράδειγμα 16: Χρήση του Αντικειμένου Event 2/2

```
<script type="text/javascript">
var pElems = document.getElementsByTagName("p");
for (var i = 0; i < pElems.length; i++) {
  pElems[i].addEventListener("mouseover", handleMouseOver);
  pElems[i].addEventListener("mouseout", handleMouseOut);
}
document.getElementById("pressme").onclick = function() {
  document.getElementById("block2").removeEventListener(
    "mouseout", handleMouseOut);
}
function handleMouseOver(e) {
  e.target.style.background='white';
  e.target.style.color='black';
}
function handleMouseOut(e) {
  e.target.style.background='gray';
  e.target.style.color='white';
}
</script>
```

Ιδιότητες Αντικειμένου Event

Όνομα	Περιγραφή	Τύπος
type	όνομα του event (π.χ. mouseover)	String
target	στοιχείο-στόχος του event	HTMLElement
currentTarget	στοιχείο του οποίου οι «listeners» εκτελούνται	HTMLElement
bubbles	true αν το event κάνει «φουσαλίδα» στο έγγραφο	Boolean
cancelable	true αν το event έχει προκαθορισμένη δράση που μπορεί να ακυρωθεί	Boolean
timeStamp	χρόνος δημιουργίας του event	

Παράδειγμα 17: Χειρισμός Event ανάλογα με τον Τύπο του

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

var pElems = document.getElementsByTagName("p");
for (var i = 0; i < pElems.length; i++) {
  pElems[i].onmouseover = handleMouseEvent;
  pElems[i].onmouseout = handleMouseEvent;
}

function handleMouseEvent(e) {
  if (e.type == "mouseover") {
    e.target.style.background='white';
    e.target.style.color='black';
  } else {
    e.target.style.background='gray';
    e.target.style.color = 'white';
  }
}
</script>
```

Ανάκτηση HTML Στοιχείων του Εγγράφου (1/2)

- Η ανάκτηση στοιχείων HTML μπορεί να γίνει με διάφορους τρόπους.
- Με χρήση κάποιων από τις μεθόδους του Document Object:

Όνομα Ιδιότητας (Μεθόδου) Επιστρέφει

getElementById(<id>)	Αντικείμενο τύπου HTMLElement
getElementsByClassName(<class>)	Array αντικειμένων, HTMLElement []
getElementsByName(<name>)	Array αντικειμένων, HTMLElement []
getElementsByTagName(<tag>)	Array αντικειμένων, HTMLElement []
querySelector(<selector>)	Το 1ο στοιχείο με τον δεδομένο CSS selector.
querySelectorAll(<selector>)	Array αντικειμένων, HTMLElement []

Ανάκτηση HTML Στοιχείων του Εγγράφου (2/2)

Μπορούμε επίσης να ανακτήσουμε στοιχεία HTML χρησιμοποιώντας ιδιότητες-πεδία του αντικειμένου document.

Όνομα Ιδιότητας Τύπος αντικειμένου

activeElement	HTMLElement που έχει το «focus»
body	HTMLElement που αναπαριστά το «σώμα»
forms	HTMLCollection (όλες οι φόρμες)
head	HTMLHeadElement (και τύπου HTMLElement)
images	HTMLCollection (όλες οι εικόνες)
links	HTMLCollection (όλοι οι σύνδεσμοι <a>)
scripts	HTMLCollection (όλα τα στοιχεία <script>)

Παράδειγμα 6 (<body>)

```
<pre id="results"></pre>

<p><!-- ... .. --></p>

<p><!-- ... .. --></p>

<script>
  var resultsElement = document.getElementById("results");
  var pElems = document.getElementsByTagName("p");
  resultsElement.innerHTML += "There are " + pElems.length
    + " p elements\n";
  var fruitsElems = document.getElementsByClassName("fruits");
  resultsElement.innerHTML += "There are " + fruitsElems.length
    + " elements in the fruits class\n";
  var nameElems = document.getElementsByName("apple");
  resultsElement.innerHTML += "There are " + nameElems.length
    + " elements with the name 'apple'";
</script>
```

Παράδειγμα 7

```
<!-- body -->
<pre id="results"></pre>

<p><!-- ... .. --></p>

<p><!-- ... .. --></p>

<script>
  var resultsElement = document.getElementById("results");

  var elems = document.images;

  for (var i = 0; i < elems.length; i++)
    resultsElement.innerHTML += "Image Element: "
      + elems[i].id + "\n";

  var srcValue = elems.namedItem("apple").src;
  resultsElement.innerHTML += "Src for apple element is: "
    + srcValue + "\n";
</script>
```

Περιήγηση στο DOM μέσω της Δεντρικής Δομής

(Παράδειγμα 8)

Όνομα Ιδιότητας Επιστρέφει

<code>childNodes</code>	<code>HTMLCollection</code>	παιδιά του στοιχείου HTML.
<code>firstChild</code>	<code>HTMLElement</code>	1ο παιδί του στοιχείου HTML.
<code>hasChildNodes()</code>	<code>true</code> / <code>false</code>	αν το στοιχείο HTML έχει παιδιά ή όχι
<code>lastChild</code>	<code>HTMLElement</code>	τελευταίο παιδί του στοιχείου HTML
<code>nextSibling</code>	<code>HTMLElement</code>	επόμενο «αδελφό» στοιχείο HTML
<code>parentNode</code>	<code>HTMLElement</code>	«πατρικό» στοιχείο HTML του τρέχοντος
<code>previousSibling</code>	<code>HTMLElement</code>	προηγούμενο «αδελφό» στοιχείο HTML

Ιδιότητες Μεταδεδομένων Εγγράφου

Όνομα	Περιγραφή	Τύπος
<code>compatMode</code>	ανάγνωση mode συμβατότητας	<code>string</code>
<code>cookie</code>	ανάγνωση/ορισμός cookies	<code>string</code>
<code>defaultCharset</code>	κωδικοποίηση κειμένου του browser	<code>string</code>
<code>defaultView</code>	Window object τρέχοντος εγγράφου	Window
<code>dir</code>	ανάγνωση/ορισμός κατεύθυνσης κειμένου	<code>string</code>
<code>domain</code>	ανάγνωση/ορισμός domain εγγράφου	<code>string</code>
<code>lastModified</code>	χρόνος τροποποίησης εγγράφου	<code>string</code>
<code>location</code>	πληροφορία για το URL του εγγράφου	Location

Παράδειγμα 2

```
<body>
<script>
  document.writeln("<pre>");
  document.writeln("characterSet: " + document.characterSet);
  document.writeln("charset: " + document.charset);
  document.writeln("compatMode: " + document.compatMode);
  document.writeln("defaultCharset: " + document.
    defaultCharset);
  document.writeln("dir: " + document.dir);
  document.writeln("domain: " + document.domain);
  document.writeln("lastModified: " + document.lastModified);
  document.writeln("referrer: " + document.referrer);
  document.writeln("title: " + document.title);
  document.write("</pre>");
</script>
</body>
```

Το αντικείμενο Location

Προσπελάινεται με `document.location`

Ιδιότητα	Περιγραφή	Τύπος
<code>protocol</code>	πρωτόκολλο ανάκτησης εγγράφου	<code>string</code>
<code>href</code>	τοποθεσία (διεύθυνση) του εγγράφου	<code>string</code>
<code>hostname</code>	όνομα του host του εγγράφου	<code>string</code>
<code>port</code>	«πόρτα» ανάκτησης του εγγράφου	<code>string</code>
<code>pathname</code>	path εγγράφου στον host	<code>string</code>
<code>assign(<URL>)</code>	μεταβαίνει στο <URL>	<code>void</code>
<code>reload()</code>	φορτώνει ξανά το έγγραφο	<code>void</code>

Παράδειγμα 3

```
<body>
  <script>
    document.writeln("<pre>");
    document.writeln("protocol: "+document.location.protocol);

    document.writeln("host: "+document.location.host);

    document.writeln("hostname: "+document.location.hostname);

    document.writeln("port: "+document.location.port);

    document.writeln("pathname: "+document.location.pathname);

    document.writeln("search: "+document.location.search);

    document.writeln("hash: "+document.location.hash);
    document.write("</pre>");
  </script>
</body>
```

Εγγραφή και Ανάγνωση Cookies

- Πληροφορία που αποθηκεύεται συσχετισμένη με το τρέχον έγγραφο.
- Απλή δομή: συμβολοσειρά με μορφή « field = value ».
- Γενικά ασφαλής μέθοδος αποθήκευσης «κατάστασης»:
 - ▶ της αλληλεπίδρασης του χρήστη με το έγγραφο.
- Όταν το έγγραφο φορτωθεί εκ νέου:
 - ▶ ο browser στέλνει στον server τα cookies που έχει αποθηκευμένα γι' αυτό.
- Έτσι π.χ. ένα σιτε ηλεκτρονικού εμπορίου γνωρίζει πόσα αντικείμενα υπάρχουν στο shopping cart του χρήστη.

Παράδειγμα 4: Περιήγηση με το αντικείμενο Location

```
<body>
  <p><!-- ... .. --></p>
  <button id="pressme" onclick="document.location.hash = 'apple'">
    Press Me
  </button>
  <p><!-- ... .. --></p>
  
</body>
```

Εγγραφή και Ανάγνωση Cookies

- Τα cookies δε μπορούν να εγκαταστήσουν κακόβουλο λογισμικό.
- Όμως μπορούν να χρησιμοποιηθούν για profiling χρηστών:
 - ▶ αγοραστικές προτιμήσεις, φύλο, ηλικία, κ.λ.π.
- Το profiling μπορεί να είναι:
 - ▶ ενοχλητικό / απωθητικό για τον χρήστη,
 - ▶ ή παραβιαστικό για προσωπικά δεδομένα.
- Τα cookies μπορεί να περιέχουν πραγματικά ευαίσθητα δεδομένα, π.χ.:
 - ▶ αριθμούς πιστωτικών καρτών,
 - ▶ passwords, κ.λ.π.
- Η νομοθεσία προδιαγράφει προειδοποίηση των χρηστών για cookies.

Η ιδιότητα `cookie` του αντικειμένου `document` επιτρέπει:

- να διαβάσουμε όλα τα cookies σχετικά με το έγγραφο HTML.
- να προσθέσουμε ένα cookie στο έγγραφο.
- να ενημερώσουμε cookies σχετικά με το έγγραφο HTML.

Η λειτουργία της ιδιότητας είναι σχετικά ασυνήθιστη.

- Επιτρέπει την εγγραφή / ενημέρωση ενός μόνο cookie τη φορά.
- Η ανάγνωσή της δίνει όλα τα cookies που είναι αποθηκευμένα και συσχετισμένα με το τρέχον έγγραφο HTML.

Παράδειγμα 5α

```
<body>
  <p id="cookiedata"></p>
  <button id="write">Add Cookie</button>
  <button id="update">Update Cookie</button>
  <script>
    var cookieCount = 0;

    document.getElementById("update").onclick = updateCookie;

    document.getElementById("write").onclick = createCookie;

    readCookies();

    /** Definition of functions: **/
    // function readCookies() { ... }
    // function createCookie() { ... }
    // function updateCookie() { ... }
```

- Ανάγνωση της `document.cookie` δίνει όλα τα cookies του εγγράφου.
 - ▶ Σε συμβολοσειρά: `"name1=value1;name2=value2;..."`

- Για να εγγράψουμε νέο cookie, δίνουμε απλώς:

```
document.cookie = "name=value"
```

- Ενημέρωση τιμής cookie με όνομα `<name>`, γίνεται επίσης με:

```
document.cookie = "name=value";
```

- Μπορούμε να δημιουργήσουμε / ενημερώσουμε ένα cookie τη φορά.

Παράδειγμα 5β

```
function readCookies() {
  document.getElementById("cookiedata").innerHTML =
    document.cookie;
}

function createCookie() {
  cookieCount++;
  document.cookie = "Cookie_" + cookieCount + "=Value_" +
    cookieCount;
  readCookies();
}

function updateCookie() {
  document.cookie = "Cookie_" + cookieCount + "=Updated_" +
    cookieCount;
  readCookies();
}
</script>
</body>
```

Αντικείμενο Window

- Επισήμως εισηγμένο στην HTML5 – ήταν ανεπίσημο «standard».
- Οι browsers υλοποιούν τη λειτουργικότητά του παρομοίως.
- Η HTML5 προδιαγράφει τη λειτουργικότητα αυτή και επιπλέον χαρακτηριστικά.
- Η υποστήριξη των επιπλέον χαρακτηριστικών διαφέρει από browser σε browser.
- Η πρόσβαση στο αντικείμενο Window γίνεται μέσω:
 - ▶ της ιδιότητας `document.defaultView` – επισήμως, στην HTML5.
 - ▶ της καθολικής αναφοράς `window` – πιο παραδοσιακά.
- Το αντικείμενο Window είναι το «πλαίσιο» εκτέλεσης των scripts.

Παράδειγμα 9

```
<!DOCTYPE HTML>
<html>
  <head><title>Example</title></head>
  <body id="bod">
    <table>
      <tr>
        <th>outerWidth:</th>
        <td id="owidth"></td></tr>
      <tr>
        <th>outerHeight:</th>
        <td id="oheight"></td></tr>
    </table>
    <script type="text/javascript">
      document.getElementById("owidth").innerHTML =
        window.outerWidth;
      document.getElementById("oheight").innerHTML =
        document.defaultView.outerHeight;
    </script>
  </body>
</html>
```

Ιδιότητες του Αντικειμένου Window

Όνομα	Περιγραφή	Τύπος
<code>innerHeight</code>	ύψος παραθύρου περιεχομένου	Number
<code>innerWidth</code>	πλάτος παραθύρου περιεχομένου	Number
<code>outerHeight</code>	εξωτερικό ύψος παραθύρου	Number
<code>outerWidth</code>	εξωτερικό πλάτος παραθύρου	Number
<code>pageXOffset</code>	#pixels οριζόντιου scroll	Number
<code>pageYOffset</code>	#pixels κάθετου scroll	Number
<code>screen</code>	αντικείμενο που περιγράφει την οθόνη	Screen
<code>screenLeft</code> <code>screenX</code>	απόσταση (#pixels) παραθύρου από αριστερό άκρο οθόνης	Number
<code>screenTop</code> <code>screenY</code>	απόσταση (#pixels) παραθύρου από κορυφή οθόνης	Number

Μέθοδοι Αλληλεπίδρασης με το Αντικείμενο Window

Όνομα	Περιγραφή	Επιστρέφει
<code>blur()</code>	απομακρύνει το «focus» από το παράθυρο	void
<code>close()</code>	κλείνει το παράθυρο <i>δεν το επιτρέπουν όλοι οι browsers</i>	void
<code>focus()</code>	φέρνει το «focus» στο παράθυρο	void
<code>print()</code>	προτρέπει εκτύπωση σελίδας	void
<code>scrollBy(x,y)</code>	κάνει scroll κατά x,y από την τρέχουσα θέση	void
<code>scrollTo(x,y)</code>	κάνει scroll στη θέση x, y	void
<code>stop()</code>	σταματά τη φόρτωση του εγγράφου	void

Παράδειγμα 10

```
<body>
  <p>
    <button id="scroll">Scroll</button>
    <button id="print">Print</button>
    <button id="close">Close</button></p>
  <p><!-- ... .. -->
    
    
  </p>
  <script>
    var buttons = document.getElementsByTagName("button");
    for (var i = 0; i < buttons.length; i++)
      buttons[i].onclick = handleButtonPress;

    function handleButtonPress(e) {
      if (e.target.id == "print") window.print();
      else if (e.target.id == "close") window.close();
      else window.scrollTo(0, 400);
    }
  </script>
</body>
```

Σύνοψη Pop-Ups στη JAVASCRIPT

alert(msg) προβάλλει το μήνυμα **msg**

prompt(msg, default) προβάλλει το μήνυμα **msg** και ζητά είσοδο, επιστρέφει σε συμβολοσειρά την είσοδο, ή **default** αν δόθηκε «κενή» είσοδος

confirm(msg) προβάλλει το **msg** και κουμπιά «OK», «Cancel», επιστρέφει **true** αν ο χρήστης πατήσει «OK», διαφορετικά επιστρέφει **false**.

Είναι μέθοδοι του ήδη ορισμένου αντικείμενου **window** (από τον browser).

Χρονομετρητές (Timers)

- Το Window Object παρέχει τη δυνατότητα να θέσουμε timers.
- Εκτελούν μια συνάρτηση μετά από συγκεκριμένο χρόνο.
- Η δυνατότητα δίνεται από συγκεκριμένες μεθόδους του window.

Μέθοδος	Περιγραφή	Επιστρέφει
clearInterval(<id>)	ακυρώνει έναν timer	void
clearTimeout(<id>)	ακυρώνει έναν timer	void
setInterval(<func>,<time>)	δημιουργεί timer	int
setTimeout(<func>,<time>)	δημιουργεί timer	int

Παράδειγμα 12α

```
<p id="msg"></p>
<p>
  <button id="settime">Set Time</button>
  <button id="cleartime">Clear Time</button>
  <button id="setinterval">Set Interval</button>
  <button id="clearinterval">Clear Interval</button>
</p>
<script>
  var buttons = document.getElementsByTagName("button");
  for (var i = 0; i < buttons.length; i++)
    buttons[i].onclick = handleButtonPress;

  var timeID;
  var intervalID;
  var count = 0;
```

Παράδειγμα 12β

```
function handleButtonPress(e) {
  if (e.target.id == "settime") {
    timeID = window.setTimeout(
      function() {displayMsg("Timeout Expired"); }, 5000);
    displayMsg("Timeout Set");
  } else if (e.target.id == "cleartime") {
    window.clearTimeout(timeID);
    displayMsg("Timeout Cleared");
  } else if (e.target.id == "setinterval") {
    intervalID = window.setInterval(
      function() {
        displayMsg("Interval expired. Counter: " + count++);},
      2000);
    displayMsg("Interval Set");
  } else if (e.target.id == "clearinterval") {
    window.clearInterval(intervalID);
    displayMsg("Interval Cleared");
  }
}
```

Παράδειγμα 12γ

```
function displayMsg(msg) {
  document.getElementById("msg").innerHTML = msg;
}
</script>
```