

Τεχνολογίες Διαδικτύου

Εισαγωγή στη javascript
Γεγονότα και Αντικείμενα

Καθηγητής Χρήστος Δουληγέρης
Δρ. Ρόζα Μαυροπόδη

Γεγονότα – τί είναι

- ▶ Τα γεγονότα είναι ενέργειες ή καταστάσεις τα οποία συμβαίνουν στο περιβάλλον/σύστημα που προγραμματίζετε.
- ▶ Το σύστημα ενημερώνει πότε συμβαίνουν αυτά τα γεγονότα, ώστε να μπορείτε να απαντήσετε σε κατά κάποιο τρόπο, αν το επιθυμείτε.
- ▶ Οι απαντήσεις περιλαμβάνουν, συνηθέστερα, την εκτέλεση κάποιας συνάρτησης, ή τμήματος κώδικα.
- ▶ Τα γεγονότα αναγνωρίζονται από πολλές γλώσσες προγραμματισμού, π.χ. C, όχι μόνο την Javascript

Τύποι γεγονότων

Στην περίπτωση του προγραμματισμού WEB εφαρμογών, όπως με την Javascript, τα γεγονότα συμβαίνουν μέσα στο παράθυρο του φυλλομετρητή (browser Window).

Σε αυτή την περίπτωση συνδέονται με

- ▶ κάποιο συγκεκριμένο στοιχείο, π.χ. ένα κουμπί,
- ▶ ή με ομάδα στοιχείων π.χ. τα στοιχεία φόρμας
- ▶ ή με το ίδιο το έγγραφο HTML, π.χ. όταν αυτό φορτώνεται
- ▶ ή με το παράθυρο του φυλλομετρητή, π.χ. όταν αλλάζει μέγεθος, resize

Γεγονότα – παραδείγματα 1/?

Παραδείγματα γεγονότων αποτελούν

- ▶ το πέρασμα του ποντικιού πάνω από ένα στοιχείο της HTML, π.χ. μια εικόνα, ένα κουμπί, μια επικεφαλίδα.
 - mouseover, click, dblclick, select, wheel
- ▶ το πάτημα ενός πλήκτρου από το πληκτρολόγιο του χρήστη,
 - cut, copy, paste, keydown, keypress, keyup
- ▶ η αλλαγή του μεγέθους ή το κλείσιμο του παραθύρου του φυλλομετρητή
 - resize, scroll, fullscreenchange, close

Γεγονότα – παραδείγματα 2/?

- ▶ η φόρτωση μιας ιστοσελίδας
 - load, unload, abort
- ▶ η αποστολή μιας φόρμας συλλογής στοιχείων
 - reset, submit
- ▶ η αναπαραγωγή ενός αρχείου ήχου
 - play, pause, seeking
- ▶ ακόμα και η εμφάνιση ενός λάθους
 - error,και πολλά άλλα γενικότερα όπως: chargingchange, online, offline

Βασικοί όροι γεγονότων

- ▶ **event handler:** αποτελεί το τμήμα του κώδικα, οποίος θα εκτελεστεί όταν συμβεί το συγκεκριμένο γεγονός. Συνήθως έχει τη μορφή συνάρτησης.
- ▶ **registering an event handler:** όταν συνδέουμε το γεγονός με κώδικα.
- ▶ **event listeners:** εναλλακτική ονομασία για τα event handlers. Ουσιαστικά τα listeners παρακολουθούν ώστε να εντοπίσουν ένα γεγονός και στη συνέχεια να εκτελεστεί ο κώδικας που ορίζεται ως handler.

Σύνδεση γεγονότων και κώδικα

Τρεις τρόποι ώστε να συνδεθούν τα γεγονότα με κάποιο κώδικα, ο οποίος πρέπει να εκτελεστεί.

- ▶ αντιμετώπιση των γεγονότων ως **χαρακτηριστικά** των ετικετών HTML.
 - `<element onγεγονός=" myScript ">`
 - onclick, onload, onsubmit κλπ.
- ▶ Αντιμετώπιση των γεγονότων ως **παραμέτρους** αντικειμένων.
 - `object.onγεγονός = function() {myScript};`
 - onclick, onload, onsubmit κλπ.
- ▶ Αντιμετώπιση των γεγονότων ως **ορίσματα** της συνάρτησης `addEventListener`
 - `object.addEventListener("γεγονός", myScript);`
 - click, load, submit κλπ.

Γεγονότα ως Χαρακτηριστικά HTML παραδείγματα

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>onClick</title>
</head>
<body>
<form>
Όνομα: <input id="firstname" type="text" ><br ><br >
Επίθετο: <input id="surname" type="text" ><br ><br >
Ηλικία: <input id="textage" type="text" ><br ><br >
<bbutton onClick="ShowPrice()" >Order</bbutton>
</form>
<script >
    function ShowPrice() {
        age=document.getElementById("textage").value;
        if (age<24){
            alert("Τιμή υπηρεσίας με έκπτωση 20%: 80");
        }
        else {
            alert("Τιμή υπηρεσίας: 100 ευρώ")
        }
    }
</script>
</body></html>
```


Γεγονότα ως Χαρακτηριστικά HTML παραδείγματα

```
onClick="ShowPrice()"
```

```
onmouseover="myFunction()"
```

Γεγονότα μέσα στην HTML – Μειονεκτήματα

- ▶ Ανακατεμένος κώδικας HTML και JavaScript. Δυσκολία στη διατήρηση και επιδιόρθωση.
- ▶ Κατάργηση του event handler σημαίνει επέμβαση στον κώδικα HTML. Έστω ότι η JavaScript είναι σε εξωτερικό αρχείο το οποίο κληροδοτείται σε πολλά αρχεία HTML.
- ▶ Χρήση των γεγονότων μόνο σε στοιχεία HTML τα οποία εμφανίζονται ήδη στο τμήμα της HTML. Δεν μπορεί να εφαρμοστεί σε στοιχεία τα οποία δημιουργούνται δυναμικά.

Γεγονότα μέσα σε κώδικα Javascript, ως παράμετροι

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>onClick</title>
</head>
<body>
<form>
Όνομα: <input id="firstname" type="text" ><br ><br >
Επίθετο: <input id="surname" type="text" ><br ><br >
Ηλικία: <input id="textage" type="text"><br ><br >
<button id="btn ">Order</button>
</form>
<script >
```

```
function ShowPrice() {
    age=document.getElementById("textage").value;
    if (age<24){
        alert("Τιμή υπηρεσίας με έκπτωση 20%: 80");
    }
    else {
        alert("Τιμή υπηρεσίας: 100 ευρώ")
    }
}
```

```
document.getElementById("btn ").onclick=function(){ShowPrice()};
```

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

Γεγονότα μέσα σε κώδικα Javascript, ως παράμετροι

```
document.getElementById("btn").onclick=function() {ShowPrice()};
```

```
document.getElementById("demo").onmouseover=myFunction;
```

Παραδείγματα ανάθεσης γεγονότων

Ανάθεση του περιεχομένου του ορισμού της συνάρτησης χωρίς όνομα, η οποία καλεί τη συνάρτηση ShowPrice, στην ιδιότητα onclick του αντικειμένου με id="btn"

```
document.getElementById("btn").onclick=function(){ShowPrice()};
```

Ανάθεση του περιεχομένου του ορισμού της συνάρτησης με όνομα ShowPrice στην ιδιότητα onclick του αντικειμένου με id="btn".

```
document.getElementById("btn").onclick=ShowPrice;
```

Η συνάρτηση δεν έχει ορίσματα.

Παραδείγματα ανάθεσης γεγονότων

Εκτέλεση της συνάρτησης με όνομα ShowPrice και ανάθεση του αποτελέσματός της στην ιδιότητα onclick του αντικειμένου με id="btn"

```
document.getElementById("btn").onclick=ShowPrice();
```

Λάθος δε θα δώσει το επιθυμητό αποτέλεσμα

element.addEventListener(event, function, useCapture)

- ▶ Αποτελεί τον ενδεδειγμένο τρόπο ανάθεσης
- ▶ Δεύτερο τρόπο χειρισμού των γεγονότων μέσα από κώδικα Javascript
- ▶ Πλεονέκτημα: μπορεί να οριστούν περισσότερα handlers (κώδικας) για ένα γεγονός ενός στοιχείου.
- ▶ Σε περίπτωση πολλαπλών handlers εκτελούνται με τη σειρά που έχουν οριστεί.
- ▶ πλήρης ορισμός:

`element.addEventListener(event, function, useCapture)`

element.addEventListener(event, function, useCapture)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>onClick</title>
</head>
<body>
<form>
Όνομα: <input id="firstname" type="text" ><br ><br >
Επίθετο: <input id="surname" type="text" ><br ><br >
Ηλικία: <input id="textage" type="text"><br ><br >
<button id="btn">Order</button>
</form>
<script >
```

```
function ShowPrice() {
    age=document.getElementById("textage").value;
    if (age<24){
        alert("Τιμή υπηρεσίας με έκπτωση 20%: 80");
    }
    else {
        alert("Τιμή υπηρεσίας: 100 ευρώ")
    }
}
```

```
document.getElementById("btn").addEventListener("click",
function(){ShowPrice()});
```

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


element.addEventListener(event, function, useCapture)

```
document.getElementById("btn").addEventListener("click", function() {ShowPrice()});
```

```
document.getElementById("demo").addEventListener("mouseover",myFunction);
```

```
document.getElementById("demo").addEventListener("mouseover",myFunction1);  
document.getElementById("demo").addEventListener("mouseover",myFunction1);
```

Κατάργηση γεγονότων

- ▶ `document.getElementById("btn").onclick=null;`
- ▶ `document.getElementById("btn").removeEventListener("click", function() {ShowPrice()});`

capture ή bubbling

Αποτελούν δυο τρόπους διάδοσης γεγονότων. Έστω ότι ισχύει:

```
<div id="myDiv"> κείμενο <p id="myP">  
παράγραφος</p> και άλλο κείμενο</div>
```

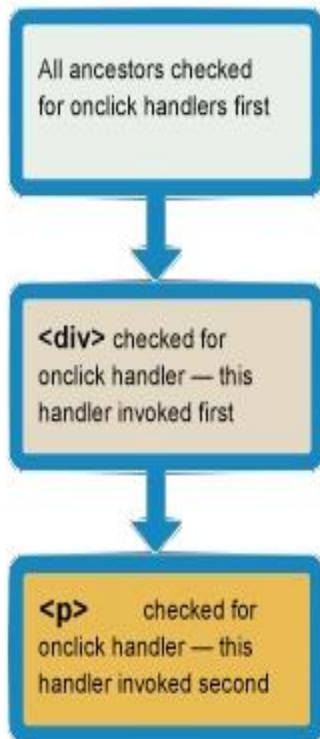
Εάν γίνει κλίκ πάνω στη λέξη **παράγραφος** ποιο γεγονός click ποιανού στοιχείου θα εκτελεστεί πρώτο;

bubbling: θα εκτελεστεί πρώτα το πιο εσωτερικό

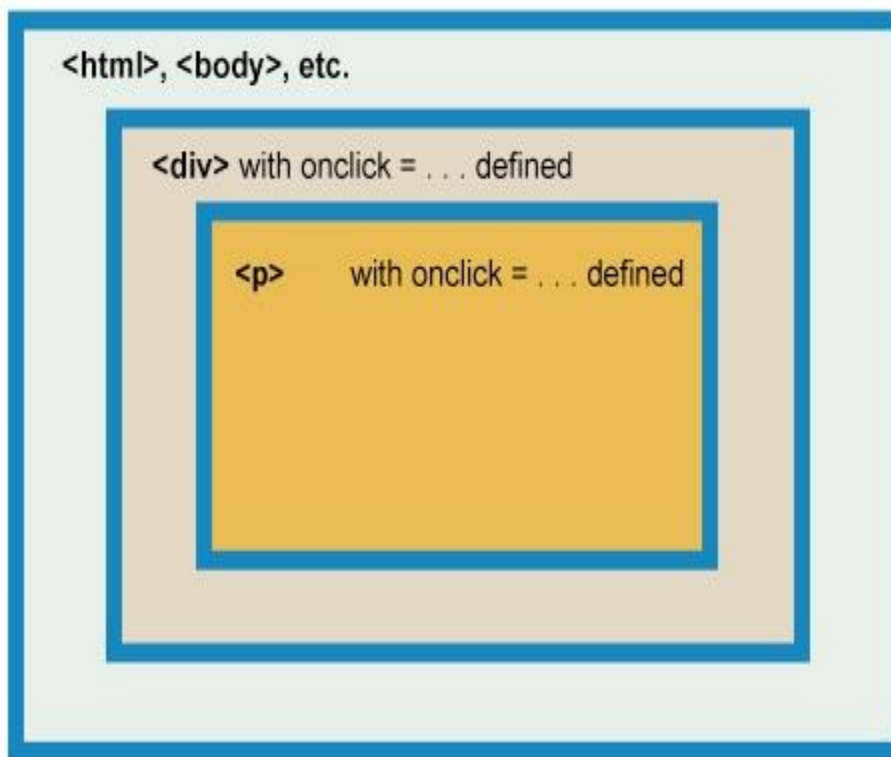
capture: θα εκτελεστεί πρώτα το πιο εξωτερικό.

capture ή bubbling

Event Capturing



click πάνω στην παράγραφο



Event Bubbling



capture ᅓ bubbling

Bubbling or Capture

Υποστηριζόμενα αντικείμενα στη JavaScript

Στη Javascript υπάρχουν διάφορα ενσωματωμένα αντικείμενα τα οποία διαθέτουν τις δικές τους ιδιότητες και παρουσιάζουν τις δικές τους, έτοιμες, μεθόδους. Τα κυριότερα από τα οποία είναι τα

Το αντικείμενο της συμβολοσειράς (string)

Το αντικείμενο του πίνακα (array)

Το αντικείμενο Math

Το αντικείμενο των αριθμών (Number)

και το αντικείμενο της ημερομηνίας (date)

Υποστηριζόμενα αντικείμενα στη JavaScript

Δημιουργία

- ▶ `var myObject = new Object;`
- ▶ `var myObject = new Object(parameter1, parameter2);`

Μη χρησιμοποιείτε το λεκτικό `new` για τη δημιουργία

`string`, `number`, `array` για λόγους απλότητας, ευκολίας στην ανάγνωση και ταχύτητας εκτέλεσης.

Αντικείμενο της συμβολοσειράς

```
function DisplayLength() {  
  
text1="A book on JavaScript"  
alert (text1.length )  
}
```

```
function DisplayChar() {  
  
text1="A book on JavaScript"  
alert (text1.substring(0,4) )  
}
```

```
function DisplayPosition() {  
  
text1="A book on JavaScript"  
y=text1.indexOf("r")  
return y; }
```

```
function TypeUpperCase() {  
  
text1="A book on JavaScript"  
alert (text1.toUpperCase() )  
}
```

```
function ChangeColor () {  
  
text1="A book on JavaScript"  
document.write  
    (text1.fontcolor('blue')) }
```

```
function Concatenate_Strings() {  
  
text1="JavaScript"  
text2="course"  
txt = text1 + text2  
    (ή txt=text1.concat(text2) )  
alert (txt) }
```


Αντικείμενο της συμβολοσειράς

Ειδικοί χαρακτήρες προκαλούν την εκτύπωση των χαρακτήρων που τους ακολουθούν:

\' single quote

\" double quote π.χ var y = "We are the so-called \"Vikings\" from the north."

\\ backslash

Ιδιαίτερες ενέργειες

\b Backspace

\r Carriage Return

\f Form Feed

\t Horizontal Tabulator

\v Vertical Tabulator

Αντικείμενο της συμβολοσειράς

Επέκταση σε πολλαπλές γραμμές

```
document.getElementById("demo").innerHTML =  
"Γειά σου Κόσμε!";
```

ή

```
document.getElementById("demo").innerHTML = " Γειά σου \  
Κόσμε!";
```

Το αντικείμενο του πίνακα

Εάν ισχύει

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.toString(); //επιστρέφει τον πίνακα ως string,  
διαχωριζόμενες τιμές με κόμμα, άρα : Banana,Orange,Apple,Mango  
fruits.join(" * "); // Banana * Orange * Apple * Mango  
fruits.pop(); //αποβάλλει το τελευταίο στοιχείο του πίνακα,  
άρα το Mango  
fruits.shift(); //αποβάλλει το πρώτο στοιχείο του πίνακα, άρα  
το Banana  
fruits.push("Kiwi"); //προσθέτει ένα νέο στοιχείο στο τέλος  
του  
fruits.unshift("Lemon"); //προσθέτει ένα νέο στοιχείο στην  
αρχή του
```

Το αντικείμενο του πίνακα

Εάν ισχύει:

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.splice(2, 0, "Lemon", "Kiwi"); // Προσθέτει δύο  
νέα στοιχεία στη θέση 2 αφαιρώντας 0 στοιχεία  
έστω ότι ισχύει :
```

```
var moreFruits = ["pear", "peach", "raspberry"];  
fruits.concat(moreFruits); // ενώνει τα στοιχεία των  
πινάκων, άρα: "Banana", "Orange", "Apple", "Mango",  
"pear", "peach", "raspberry"
```

Το αντικείμενο του πίνακα

Εάν ισχύει:

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
fruits.slice(3); //δημιουργεί ένα νέο πίνακα με  
κομμάτια του παλιού, άρα: "Mango"
```

```
fruits.slice(1, 3); //από την πρώτη θέση του πίνακα  
έως την 3, χωρίς να την περιλάβει, άρα: "Apple"
```

```
fruits.sort(); //ταξινομεί τον πίνακα:
```

```
"Apple", "Banana", "Mango", "Orange"
```

```
fruits.reverse(); //φθίνουσα ταξινόμηση:
```

```
"Orange", "Mango", "Banana", "Apple"
```

Το αντικείμενο των αριθμών

Μετατροπή σε string:

```
var x = 123;
```

```
x.toString(); // returns 123 from variable x
```

```
(123).toString(); // returns 123 from literal 123
```

```
(100 + 23).toString(); // returns 123 from expression 100 + 23
```

Το αντικείμενο των αριθμών

Μετατροπή μεταβλητών σε number:

```
x = true;
```

```
Number(x);    // returns 1
```

```
x = false;
```

```
Number(x);    // returns 0
```

```
x = "10"
```

```
Number(x);    // returns 10
```

```
x = "10 20"
```

```
Number(x);    // returns NaN
```

Το αντικείμενο των αριθμών

Μετατροπή μεταβλητών σε number:

```
parseInt("10"); // returns 10
```

```
parseInt("10.33"); // returns 10
```

```
parseInt("10 20 30"); // returns 10
```

```
parseInt("10 years"); // returns 10
```

```
parseInt("years 10"); // returns NaN
```


Το αντικείμενο των αριθμών

Μετατροπή μεταβλητών σε number:

```
parseFloat("10"); // returns 10
```

```
parseFloat("10.33"); // returns 10.33
```

```
parseFloat("10 20 30"); // returns 10
```

```
parseFloat("10 years"); // returns 10
```

```
parseFloat("years 10"); // returns NaN
```

Το αντικείμενο Math

Χρησιμοποιείται για την εκτέλεση μαθηματικών πράξεων σε αριθμούς.

Δημιουργία: δεν χρειάζεται να το δημιουργήσουμε.
Χρησιμοποιείται απευθείας.

```
Math.PI; // returns 3.141592653589793
```

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

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

```
Math.POW(8, 2); // returns 64
```

```
Math.SQRT(64); // returns 8
```

```
Math.MIN(0, 150, 30, 20, -8, -200); // returns -200
```

```
Math.RANDOM();
```

```
Math.FLOOR(4.7); // returns 4
```

Το αντικείμενο Math

- ▶ `Math.floor(Math.random() * 10);` // returns a number between 0 and 9

random

Το αντικείμενο Date

Επιτρέπει εργασίες με ημερομηνίες, όπως (years, months, days, hours, minutes, seconds, and milliseconds)

Η ημερομηνίες μπορούν να παρασταθούν έτσι:

Sun Nov 19 2017 23:56:45 GMT+0200 (GTB Standard Time)

ή έτσι:

1511128605730 το οποίο και αντιπροσωπεύει τα milliseconds από το January 1, 1970, 00:00:00 .

Το αντικείμενο Date

Για να χρησιμοποιηθεί πρέπει πρώτα να δημιουργηθεί ως: `new Date()`

`new Date(milliseconds)`

`new Date(dateString)`

`new Date(year, month, day, hours, minutes, seconds, milliseconds)`

Το αντικείμενο Date

var d = new Date(); // παρέχει την τρέχουσα ημερομηνία

Mon Nov 20 2017 00:14:03 GMT+0200 (GTB Standard Time)

var d = new Date("October 13, 2014 11:13:00");

//την παρεχόμενη ημερομηνία

Mon Oct 13 2014 11:13:00 GMT+0300 (GTB Daylight Time)

var d = new Date(86400000);

// παρέχει την ημερομηνία από το 01 January 1970 00:00:00 + τον αριθμό των millisecond

Fri Jan 02 1970 02:00:00 GMT+0200 (GTB Standard Time)

var d = new Date(99, 5, 24, 11, 33, 30, 0);

//Παρέχει τα year, month, day, hour, minute, second, and millisecond

Thu Jun 24 1999 11:33:30 GMT+0300 (GTB Daylight Time)

Το αντικείμενο Date

If now = Thu Dec 5 23:55:16 2011

now.**getDay()**= 4 Day of week (0=Sunday)

now.**getMonth()**= 11 Month (0 to 11)

now.**getDate()**=5 Day of month

now.**getFullYear()**= 2011 Year

now.**getHours()**= 23 Hours

now.**getMinutes()**= 55 Minutes

now.**getSeconds()**= 16 Seconds

Το αντικείμενο Date

If now = Thu Dec 5 23:55:16 2011

now.**setDate**(15)= Wed Dec 15 23:55:16 2011

now.**setFullYear**(2020)=

Tue Jan 14 2020 00:30:56 GMT+0200 (GTB
Standard Time)

Το αντικείμενο παραθύρου

Ιδιότητα	Περιγραφή
name	Το όνομα του παραθύρου
closed	Δείχνει εάν ένα παράθυρο είναι κλειστό
location	Πληροφορία σχετική με το url του παραθύρου
opener	Το όνομα του παραθύρου που άνοιξε αυτό το παράθυρο
document	Το έγγραφο του παραθύρου
status	Κείμενο που θα εμφανιστεί στη μπάρα κατάστασης
innerHeight ή outerHeight	Εσωτερικό ή εξωτερικό ύψος παραθύρου
innerWidth ή outerWidth	Εσωτερικό ή εξωτερικό πλάτος παραθύρου

Ιδιότητες παραθύρου

Μηνύματα που φαίνονται στην μπάρα / γραμμή κατάστασης
Μπάρα στο κάτω μέρος που δείχνει το URL ή γράφει
“Document:Done”.

Μπορούμε να χρησιμοποιήσουμε αυτή την μπάρα για να
εμφανίσουμε ένα μήνυμα όταν ο χρήστης βάλει το δείκτη του
ποντικιού του πάνω σε κάποιο σύνδεσμο

Για να επιτευχθεί αυτό, απλά προσθέτουμε ένα μικρό
JavaScript σε μια κανονική άγκυρα.

```
<a href= "unipi.html"  
onMouseOver="window.status='University of Piraeus'  
>Unipi</a>
```

Μέθοδοι παραθύρου

Οι πιο σημαντικές μέθοδοι του παραθύρου είναι η `open(arguments)`, η οποία ανοίγει ένα παράθυρο βασισμένη στις τιμές που δίνονται στα ορίσματα της και η `close()` η οποία κλείνει ένα παράθυρο.

```
function openWin(URL) {  
  aWindow = window.open(URL, "thewindow", "toolbar=no,width=350,  
  height=400,status=no,scrollbars=yes,resize=no,menubar=no");  
}
```

...

```
<a href="javascript:openWin('Intro.html');" >Introduction</a>
```

Υπάρχουν, όμως, και οι μέθοδοι `alert(string)`, `confirm(string)` και `prompt(string, [inputDefault])` οι οποίες εμφανίζουν στην οθόνη παράθυρα μηνυμάτων με την τιμή της συμβολοσειράς, που δίνεται ως όρισμά τους.

Μέθοδοι παραθύρου

window

Το αντικείμενο εγγράφου

Το αντικείμενο του εγγράφου (document) προσδιορίζει το περιεχόμενο ενός παραθύρου, δηλαδή τη σελίδα HTML που εμφανίζεται στο παράθυρο.

☒ Κάποιες χρήσιμες ιδιότητες του εγγράφου είναι οι ακόλουθες: forms, images, links, anchors, title, url

☒ Το αντικείμενο του εγγράφου έχει μεθόδους όπως:

-open(), που ανοίγει ένα έγγραφο για να γραφτεί το περιεχόμενό του

-close(), που κλείνει το έγγραφο όταν ολοκληρωθούν οι αλλαγές του περιεχομένου του

-write() και writeln() που γράφουν το περιεχόμενό του. Η διαφορά των write() και writeln() είναι ότι η δεύτερη συμπληρώνει το χαρακτήρα αλλαγής γραμμής στο τέλος του κειμένου

Το αντικείμενο φόρμας

Το αντικείμενο της φόρμας έχει **ιδιότητες** όπως:

name (όνομα φόρμας),

elements (ένας πίνακας με τα στοιχεία της φόρμας),

length (ο αριθμός των στοιχείων της φόρμας)

method (η μέθοδος (GET ή POST) με την οποία υποβάλλονται τα στοιχεία της) και

action (το σενάριο στο οποίο στοχεύει η υποβολή των στοιχείων της φόρμας).

Οι βασικές **μέθοδοι** του στοιχείου της φόρμας είναι οι **reset()** και **submit()**. Η πρώτη αρχικοποιεί τις τιμές των στοιχείων μιας φόρμας ενώ η δεύτερη υποβάλει τις τιμές τους.

Αντίστοιχα τα πιο σημαντικά γεγονότα σε μια φόρμα είναι το **onReset** και το **onSubmit** τα οποία συμβαίνουν όταν πατηθούν τα πλήκτρα τύπου **reset** και **submit**, δηλαδή όταν κληθούν οι μέθοδοι **reset()** και **submit()**.

Το αντικείμενο φόρμας

- ▶ Όλα τα στοιχεία μιας φόρμας, (πεδία κειμένου (**text**), πλήκτρα (**button**), εικόνες (**images**) και πεδία επιλογής (**radio** και **checkbox**) έχουν **ιδιότητες** όπως **name** (το όνομα του πεδίου), **value** (η τιμή του πεδίου) και **defaultValue** (μια προκαθορισμένη τιμή για το πεδίο)
- ▶ και **μεθόδους** όπως **focus()** και **select()**.
- ▶ Στα αντικείμενα μιας φόρμας μπορούν να συμβούν **γεγονότα**:
- ▶ **onFocus** - όταν ο κέρσορας εισέρχεται στη συγκεκριμένη περιοχή κειμένου ή πεδίο κειμένου.
- ▶ **onBlur** - όταν ένα αντικείμενο παύει να είναι ενεργό. Όταν, δηλαδή ο κέρσορας μεταφερθεί σε άλλο αντικείμενο.
- ▶ **onChange** - όταν αλλάξει την τιμή των αντικειμένων.
- ▶ **onClick** - όταν ο χρήστης επιλέξει με το ποντίκι ένα αντικείμενο.
- ▶ **onMouseOver** - όταν ο δείκτης του ποντικιού βρεθεί πάνω από ένα αντικείμενο.
- ▶ **onMouseOut** - όταν ο δείκτης του ποντικιού φύγει πάνω από κάποιο αντικείμενο.

Συνήθη λάθη

Undefined Variables

```
abc = 23;
```

```
var abc = 23;
```

```
alert(abc);
```

```
function foo(parametrOne) {  
    alert(parameterOne);  
}
```


Συνήθη λάθη

Case Sensitivity

```
var myName = "Jeremy";  
If (myName == "jeremy") {  
    alert(myName.toUpperCase());  
}
```

Συνήθη λάθη

Incorrect Number of Closing Braces

```
function myFunction()  
{  
  var x = 1;  
  var y = 2;  
  if (x <= y)  
  {  
    if (x == y)  
    {  
      alert("x equals y");  
    }  
  }  
}  
myFunction();
```

Συνήθη λάθη

Incorrect Number of Closing Parentheses

```
if (myVariable + 12) / myOtherVariable < myString.length)
```

Συνήθη λάθη

Using Equals (=) Rather than Equality (==)

```
var myNumber = 99;  
if (myNumber = 101) {  
    alert("myNumber is 101");  
} else {  
    alert("myNumber is " + myNumber);  
}
```

Συνήθη λάθη

Missing Plus Signs During Concatenation

```
var myName = "Jeremy";
```

```
var myString = "Hello";
```

```
Error Handling | 555
```

```
var myOtherString = "World";
```

```
myString = myName + " said " + myString + " "
```

```
myOtherString;
```

```
alert(myString);
```