

Ανάπτυξη Διαδικτυακών Εφαρμογών

Εισαγωγή στη Javascript

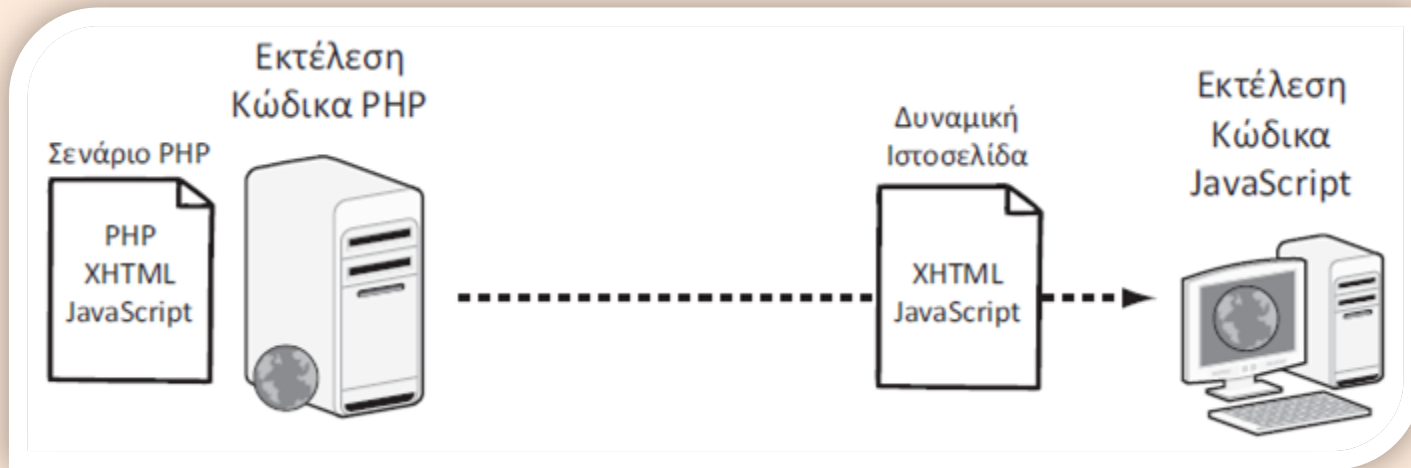
Σαπρίκης Βαγγέλης

Υλικό από το βιβλίο του μαθήματος: Κεντερλής Π. (2009).
Ανάπτυξη Διαδικτυακών Εφαρμογών – Θεωρία και Πράξη,
Αθήνα.

Τι είναι η JavaScript;

- Η JavaScript είναι μία **γλώσσα σεναριακού προγραμματισμού** σχεδιασμένη για χρήση σε διαδικτυακές εφαρμογές και δεν έχει τις δυνατότητες που παρέχει μια τυπική γλώσσα προγραμματισμού.
- **Εκτελείται από τον Web Browser** και όχι από τον Web Server (Client-Side Programming).
- **Δεν έχει σχέση με τη γλώσσα Java** καθώς αναπτύχθηκε από τον Brendan Eich της Netscape και όχι από τη Sun.
- Έχει συντακτικό που βασίζεται στη γλώσσα Java (κατά συνέπεια και στη γλώσσα C).
- Ενσωματώνεται σε έγγραφα HTML/ XHTML και επιτρέπει τον έλεγχο στοιχείων των εγγράφων και του ίδιου του Web Browser.

Javascript vs PhP



Ισχυρά Σημεία της JavaScript

- Προσθέτει δυνατότητες που λείπουν από την HTML:
 - **Επικύρωση δεδομένων σε φόρμες** χωρίς την ανάγκη κατάθεσης της φόρμας στον Web Server.
 - **Διαλογική επικοινωνία της εφαρμογής με το χρήστη** χωρίς την ανάγκη χρήσης ενδιάμεσων σελίδων HTML.
 - **Δυναμική ανανέωση, έλεγχος και επεξεργασία συγκεκριμένων σημείων** ενός εγγράφου HTML.
 - **Δυναμική ασύγχρονη επικοινωνία του εμφανιζόμενου εγγράφου με τον Web Server** μέσω της τεχνολογίας Ajax.

Αδύναμα Σημεία της JavaScript

- Η επιχειρηματική λογική και ευαίσθητα σημεία της εφαρμογής δεν πρέπει να υλοποιείται με JavaScript καθώς ο κώδικας μεταφέρεται στον Web Browser, είναι διαθέσιμος και ευάλωτος σε επιθέσεις κακόβουλων χρηστών.
 - Ο σχεδιαστής της εφαρμογής πρέπει να **αντιμετωπίζει τα δεδομένα και τις λειτουργίες του κώδικα JavaScript με καχυποψία.**
- Είναι γλώσσα περιορισμένων δυνατοτήτων:
 - Ο κώδικας για λόγους ασφάλειας εκτελείται από μια ιδεατή μηχανή που παρέχεται από τον Web Browser, παρέχοντας ένα ιδιαίτερα περιορισμένο περιβάλλον εκτέλεσης.
 - Δεν υπάρχει δυνατότητα χειρισμού αρχείων του Client για λόγους ασφάλειας.
 - Δεν υπάρχουν δυνατότητες άμεσης εκτέλεσης δυαδικών προγραμμάτων στον Client.
 - Δεν υπάρχει δυνατότητα διασύνδεσης με βάσεις δεδομένων.

Ενσωμάτωση Κώδικα JavaScript σε Έγγραφα HTML (1)

- Ως κώδικας τυπικά εντός της περιοχής head ή body ενός εγγράφου, προσβάσιμος μόνο από αυτό.
- Ως εξωτερικό αρχείο με κατάληξη .js το οποίο συνδέεται με τον ακόλουθο κώδικα HTML σε κάθε έγγραφο από το οποίο θέλουμε να είναι προσβάσιμος και εκτελέσιμος.

```
<script language="javascript"
  type="text/javascript">
<!--
// JavaScript code
-->
</script>
```

```
<script language="javascript"
  type="text/javascript"
  src="myscript.js">
</script>
```

Ενσωμάτωση Κώδικα JavaScript σε Έγγραφο HTML (2)

- Καλό είναι ο κώδικας να είναι συγκεντρωμένος σε ένα μέρος.
- Είναι εφικτό να υπάρχουν πολλά scripts της Javascript σε ένα html έγγραφο.
- Εάν τοποθετείται ο κώδικας στο <body>, είναι προτιμότερο το script να βρίσκεται στο τέλος του <body> (για βελτίωση της φόρτωσης της ιστοσελίδας).
- Η Javascript δεν έχει ενσωματωμένη 'print' ή καμία σχετική συνάρτηση εμφάνισης, για την προβολή των ενεργειών της.

Javascript:

Εναλλακτικές εμφάνισης αποτελεσμάτων

1. `Window.alert()`
2. `Document.write()`
3. `innerHTML`

Παράδειγμα - window.alert()

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

<h1>My First Web Page</h1>

<script>
window.alert(1 + 2);
</script>

</body>
</html>
```

Παράδειγμα - document.write()

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

<h1>My First Web Page</h1>

<script>
document.write(1 + 2);
</script>

</body>
</html>
```

Παράδειγμα - innerHTML

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
<h1>My First Web Page</h1>
```

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

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

```
<script>
```

```
document.getElementById("demo").innerHTML = 1 + 2;
```

```
</script>
```

```
</body>
```

```
</html>
```

Σχόλια

- // : σχόλια μίας γραμμής
- /*
..... */ : σχόλια πολλών γραμμών

Μεταβλητές στη JavaScript (1)

- Όμοια με την PHP, οι μεταβλητές της JavaScript δεν έχουν αυστηρά καθοριζόμενο τύπο δεδομένων, ο οποίος μπορεί να αλλάζει δυναμικά κατά τη διάρκεια εκτέλεσης του κώδικα.
- Η δήλωση του ονόματός της μπορεί να ξεκινάει από λατινικό γράμμα, κάτω παύλα (_) ή \$. Στη συνέχεια, μπορεί να έπονται και αριθμοί.
- Πρέπει, όπως και σε κάθε γλώσσα προγραμματισμού, να είναι μία λέξη.
- Όμοια με την PHP, οι μεταβλητές είναι ευαίσθητες στη σύνταξη με κεφαλαίους-πεζούς χαρακτήρες (case sensitive).

Μεταβλητές στη JavaScript (2)

- Όμοια με την PHP, οι μεταβλητές διακρίνονται σε τοπικές και καθολικές
- Μια τοπική μεταβλητή ορίζεται με τη χρήση του προθέματος **var**, π.χ.:
 - `var year_of_millennium = 2000;`
 - `var name = 'Mr. Smith';`
 - `var IsAllowed = true;`
 - `var PricePerKg = 2.36;`
- Οι τοπικές μεταβλητές ενός επιπέδου είναι προσβάσιμες και από όλα τα σημεία κώδικα του ίδιου επιπέδου ή τις οριζόμενες σε αυτό συναρτήσεις.
- Οι τοπικές μεταβλητές διαγράφονται με την ολοκλήρωση της συνάρτησης στην οποία ανήκουν.
- Οι καθολικές μεταβλητές διαγράφονται μόλις κλείσει η σελίδα.

**Παραδείγματα
Αριθμητικών
Μεταβλητών και
Μεταβλητών
Συμβολοσειρών**

```
var aNumericVariable1 = 1979;
var aNumericVariable2 = -2009;
var anotherNumVariable1 = 3.14;
var anotherNumVariable2 = 2.7E3; // 2700
var anotherNumVariable3 = -678.9E-2; // -
    6.789
var anotherNumVariable4 = 0127; //
    οκταδική αναπαράσταση της τιμής 87
var anotherNumVariable5 = 0x1fa32; //
    δεκαεξαδική αναπαράσταση της τιμής
    129586
var aStringVariable = 'The classic Hello
    World string';
var aStringVariable1 = 'This is one line
    of text\nThis is another line.';
var aStringVariable2 = "It's a pity";
var aStringVariable3 = 'And then she said
    "Good night"';
var aTrueStatement = true;
var aFalseStatement = false;
```

Μεταβλητές στη JavaScript (3)

- Πολλές μεταβλητές μπορούν να δηλωθούν μαζί, με την παρεμβολή ενός κόμματος μεταξύ τους.

π.χ. `var x = 2, y = "John", price=100;`

- Εάν δηλωθεί εκ νέου μία μεταβλητή, δε χάνει την προϋπάρχουσα τιμή της.
- Όταν προσθέτουμε κείμενο με αριθμό σε μία μεταβλητή, η Javascript «συμπεριφέρεται» στον αριθμό σαν κείμενο (string).

π.χ. `var x = 16 + 4 + "George";`

π.χ. `var x = "George" + 16 + 4;`

- Εάν δηλωθεί αριθμός σε εισαγωγικά, η συγκεκριμένη μεταβλητή είναι τύπου κειμένου (string).
- Μπορούμε να ενώσουμε κείμενα και αριθμούς μεταξύ τους με τον τελεστή της πρόσθεσης.

π.χ. `var d = "John" + " " + "Doe";`

Μεταβλητές στη JavaScript (4)

- Στη Javascript μία μεταβλητή μπορεί να χρησιμοποιηθεί ως διαφορετικός τύπων, πολλές φορές.

π.χ. `var x;`

`var x = 9;`

`var x = "John";`

Αριθμητικοί Τελεστές

Παράδειγμα	Περιγραφή
$a + b$	Πρόσθεση
$a - b$	Αφαίρεση
$a * b$	Πολλαπλασιασμός
a / b	Διαίρεση
$a \% b$	Υπόλοιπο διαίρεσης

Τελεστές Σύγκρισης

Παράδειγμα	Περιγραφή
<code>a == b</code>	Έλεγχος ισότητας τιμής
<code>a === b</code>	Έλεγχος ισότητας τιμής και τύπου δεδομένων
<code>a != b</code>	Έλεγχος ανισότητας τιμής
<code>a <> b</code>	Έλεγχος ανισότητας τιμής
<code>a !== b</code>	Έλεγχος ανισότητας τιμής και τύπου δεδομένων
<code>a < b</code>	Έλεγχος μικρότερης τιμής
<code>a > b</code>	Έλεγχος μεγαλύτερης τιμής
<code>a <= b</code>	Έλεγχος μικρότερης-ίσης τιμής
<code>a >= b</code>	Έλεγχος μεγαλύτερης-ίσης τιμής

Λογικοί Τελεστές

Παράδειγμα	Περιγραφή
<code>!a</code>	Λογική αντιστροφή της τιμής της μεταβλητής
<code>a && b</code>	Λογικό AND
<code>a b</code>	Λογικό OR

1. If/ if-else

```
if (condition) {  
    code to be executed if this condition is true;  
} else if (condition) {  
    code to be executed if this condition is true;  
} else {  
    code to be executed if all conditions are false;  
}
```

Παράδειγμα:

```
if (x>5) {  
    y=10; }  
else if (x<5) {  
    y=2;  
} else {  
    y=90;  
}
```

Συναρτήσεις της Javascript

- Οι συναρτήσεις χρησιμοποιούνται για την επαναχρησιμοποίηση κώδικα, όσες φορές το επιθυμούμε. Μάλιστα, η χρήση συναρτήσεων με διαφορετικές τιμές παραμέτρων είναι σε θέση να μας επιστρέψει παραμετροποιημένα αποτελέσματα.
- Το όνομα μιας συνάρτησης θα πρέπει απαραίτητα να ξεκινάει από ένα αλφαβητικό χαρακτήρα (κεφαλαία ή πεζά) ή το χαρακτήρα της κάτω παύλας (underscore).
- Η JavaScript κάνει ισχυρή διάκριση στη χρήση κεφαλαίων ή πεζών χαρακτήρων στα ονόματα των συναρτήσεων.
- Εξ ορισμού, τα δεδομένα των ορισμάτων εισόδου αποδίδονται στον κώδικα της συνάρτησης με αντιγραφή της τιμής του σε τοπικές μεταβλητές αυτή (κλήση μέσω τιμής – By Value).
- Η προσπάθεια κλήσης συνάρτησης χωρίς τα εισαγωγικά της () επιστρέφει τη μορφή δήλωσής της.

Γενική Μορφή Σύνταξης Συναρτήσεων

```
function όνομα_συνάρτησης( όρισμα_1, όρισμα_2, ... , όρισμα_N ) {  
    δήλωση_1;  
    δήλωση_2;  
    . . .  
    δήλωση_N;  
    return επιστρεφόμενα_δεδομένα;  
}
```

Προαιρετικά Ορίσματα Εισόδου Συναρτήσεων

Η JavaScript δεν υποστηρίζει κάποια εύκολη και άμεση σύνταξη της λίστας ορισμάτων εισόδου με εξ ορισμού τιμές όπως η PHP.

Αντίθετα, απαιτείται εσωτερικός έλεγχος με επιπλέον κώδικα που θα εντοπίζει τη μη χρήση του ορίσματος και θα του αναθέτει μια τιμή.

```
function myOwnFunction( optional ) {  
    if ( typeof optional == 'undefined' )  
        optional = 1;  
    return optional + 1;  
}
```

```
myOwnFunction(3); // Επιστρέφεται η τιμή 4  
myOwnFunction(); // Επιστρέφεται η τιμή 2
```

Γραμματική Σημειογραφία Συνάρτησης

Η γραμματική σημειογραφία συνάρτησης (Function Literal Notation) αποτελεί εναλλακτικό τρόπο σύνταξης των συναρτήσεων και οφείλει την ύπαρξη της στο γεγονός ότι η JavaScript αντιμετωπίζει και τις συναρτήσεις ως αντικείμενα.

Μετά την εκτέλεση αυτής της δήλωσης της συνάρτησης, αυτή μπορεί να κληθεί βάσει του ονόματος της.

```
var όνομα_συνάρτησης = function( όρισμα_1,  
    όρισμα_2, ... , όρισμα_N ) {  
    δήλωση_1;  
    δήλωση_2;  
    . . .  
    δήλωση_N;  
    return επιστρεφόμενα_δεδομένα;  
}
```

Κλήση συνάρτησης

- Ο κώδικας της συνάρτησης εκτελείται όταν:
 1. ένα συμβάν λαμβάνει χώρα (π.χ. ο χρήστης κλικάρει ένα κουμπί)
 2. όταν καλείται από κώδικα Javascript
 3. αυτόματα

Ασκήσεις σε Javascript – 1^η (1)

Δημιουργήστε μία εφαρμογή σε javascript, η οποία θα δέχεται σε 2 texts το οποιοδήποτε όνομα και επίθετο. Θα το εμφανίζει, μόλις πατηθεί το κουμπί «Εμφάνισε», με τη μορφή «Το όνομά σου είναι ... και το επίθετό σου είναι ...». Επίσης, το κουμπί «Καθαρισμός» θα επαναφέρει το αρχείο html στην αρχική του κατάσταση.

Ασκήσεις σε Javascript – 1^η (2)

Απάντηση

```
<html>
<head></head>
<body>
<form name="form1">
Όνομα: <input type="text" name="text1"><br><br>
Επίθετο: <input type="text" name="text2"><br><br>
<input type="button" name="button1" value="Εμφάνισε" onclick="er()">
<input type="button" name="button2" value="Καθαρισμός" onclick="xe()">
</form>
<p id="x1"></p>
<script>
function er(){
var x=form1.text1.value;
var y=form1.text2.value;
document.getElementById("x1").innerHTML= "Το όνομα σου είναι " + x + " και το επίθετό σου είναι " + y;
}
function xe(){
document.getElementById("x1").innerHTML=" ";
form1.text1.value=" ";
form1.text2.value=" ";
} </script>
</body>
</html>
```

Ασκήσεις σε Javascript – 2^η (1)

Δημιουργήστε μία εφαρμογή σε javascript η οποία θα δέχεται σε 2 texts τη βάση και το ύψος ενός οποιουδήποτε τριγώνου, και θα εμφανίζει το εμβαδόν του. Ωστόσο, σε περίπτωση που αυτές οι 2 τιμές δεν είναι θετικές, να εμφανίζεται μήνυμα λάθους. Επίσης, το κουμπί «Καθαρισμός» θα επαναφέρει το αρχείο html στην αρχική του κατάσταση.

Ασκήσεις σε Javascript – 2^η (2)

Απάντηση

```
<html>
<head></head>
<body>
<form name="form1">
Δώσε βάση τριγώνου:<input type="text" name="text1"><br><br>
Δώσε ύψος τριγώνου:<input type="text" name="text2"><br><br>
<input type="button" name="button1" value="Εμφάνισε" onclick="test1()">
<input type="button" name="button2" value="Καθαρισμός" onclick="test2()">
</form>
<p id="x1"></p>
<script>
function test1(){
var x=form1.text1.value;
var y=form1.text2.value;
if ((x>0) && (y>0))
document.getElementById("x1").innerHTML="Το εμβαδόν του τριγώνου είναι " + (x*y/2);
else
{document.getElementById("x1").innerHTML="error";}
}
function test2(){
form1.text1.value="";
form1.text2.value="";
document.getElementById("x1").innerHTML="";
}
</script>
</body>
</html>
```

Ασκήσεις σε Javascript – 3^η (1)

Δημιουργήστε μία εφαρμογή σε javascript η οποία θα δέχεται σε 1 text την ηλικία ενός ατόμου και θα εμφανίζει το μήνυμα: α) λάθος, για τιμή αρνητική ή μηδενική, β) ανήλικος, για τιμή 1 έως 17, γ) ενήλικος, για τιμή 18 έως 67 και δ) ηλικιωμένος, για τιμή μεγαλύτερη από 67.

Ασκήσεις σε Javascript – 3^η (2)

Απάντηση

```
<html>
<head></head>
<body>
<form name="form1">
Δώσε ηλικία: <input type="text" name="text1"><br><br>
<input type="button" name="button1" value="Εμφάνισε" onclick="t1()">
</form>

<p id="x1"></p>

<script>
function t1() {
var age=form1.text1.value;
if (age<=0)
{document.getElementById("x1").innerHTML="error";}
else if ((age>=1) && (age<=17)) // (age<18)
{document.getElementById("x1").innerHTML="ανήλικος";}
else if ((age>=18) && (age<=67)) // (age<=67)
{document.getElementById("x1").innerHTML="ενήλικος";}
else // else if (age>67)
{document.getElementById("x1").innerHTML="ηλικιωμένος";}
}
</script>
</body>
</html>
```