JavaScript™

Τεχνολογίες προγραμματισμού πελάτη (client side)

Εισαγωγή στην Javascript

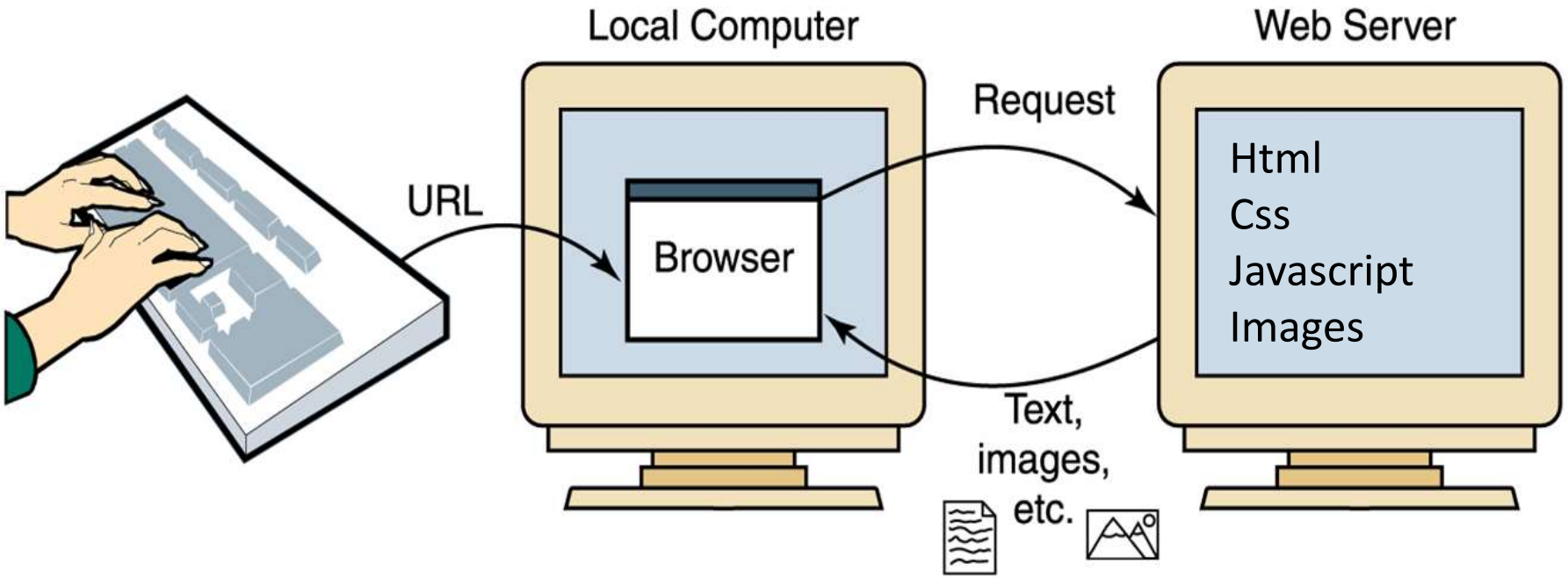
Τεχνολογίες προγραμματισμού πελάτη (client side)

Μέχρι τώρα μάθαμε για την HTML και την CSS. Τεχνολογία με την οποία μπορούμε να απεικονίσουμε την πληροφορία στατικά. Δεν μπορούμε όμως να απεικονίσουμε την δυναμική της πληροφορίας. Για αυτό το λόγο χρησιμοποιούνται οι τεχνολογίες προγραμματισμού πελάτη (client side) με τις οποίες μπορούμε να εισάγουμε κώδικα που εκτελείται δυναμικά και αλληλεπιδρά με τα στοιχεία της HTML. Ο κώδικας αυτός εκτελείται στον φυλλομετρητή του χρήστη (client). Διάφορες τεχνολογίες έχουν δημιουργηθεί για αυτό το σκοπό όπως είναι η vbscript και η Javascript . Στις επόμενες ενότητες θα μάθουμε τις βασικές έννοιες της Javascript έτσι ώστε να μπορούμε να την χρησιμοποιούμε αλλά και να αποκτήσουμε τις βάσεις για περεταίρω εκμάθηση αυτής της τεχνολογίας.

Εισαγωγή στην Javascript

Η Javascript αποτελεί μια από τις πιο διαδεδομένες γλώσσες συγγραφής σεναρίων (scripting language) για προγραμματισμό στην πλευρά του πελάτη (client side). Έχει πολλές δυνατότητες και χρησιμοποιείται για τον εμπλουτισμό ενός online πληροφοριακού συστήματος με αυξημένη λειτουργικότητα και αλληλεπίδραση. Το σημαντικότερο πλεονέκτημά της είναι ότι εκτελείται στο πελάτη (στον φυλλομετρητή) καθιστώντας την εφαρμογή ολοένα και πιο γρήγορη. Επίσης η Javascript χρησιμοποιείται από πολλούς προγραμματιστές για αυτό και υπάρχουν άφθονοι πόροι στο διαδίκτυο τόσο για την εκμάθησή της όσο και για την γρήγορη επίλυση προβλημάτων (έτοιμες και πολλές βιβλιοθήκες είναι ελεύθερες προς χρήση).

Client side programming



HTML και Javascript

Η Javascript ενσωματώνεται σε ένα αρχείο HTML με την ετικέτα <SCRIPT> της HTML. Ενδιάμεσα σε αυτή την ετικέτα μπορεί να γραφεί κώδικας σε Javascript. Επίσης κώδικας σε αυτή τη γλώσσα μπορεί να γραφεί σε εξωτερικό αρχείο με κατάληξη .js (π.χ. library.js). Η σύνδεση με το αρχείο αυτό γίνεται μέσω της ιδιότητας src της ετικέτας <SCRIPT>. Η ετικέτα <SCRIPT> μπορεί να ενσωματωθεί σε οποιοδήποτε μέρος μιας σελίδας HTML (BODY ή HEAD).

```
<!DOCTYPE html>
<!--Αρχείο HTML με ενσωματωμένο το κώδικα Javascript
-->
<html>
  <head>
   <title>Τίτλος ιστοσελίδας</title>
   </head>
   <body>
      <script type="text/javascript">
            alert('Ο Κόσμος μας!');
      </script>
   </body>
</html>
```

```
<!DOCTYPE html>
<!--Αρχείο HTML με σύνδεσμο σε εξωτερικό αρχείο κώδικα Javascript
<html>
  <head>
   <title>Τίτλος ιστοσελίδας</title>
      <script type="text/javascript“ src= “message.js”>
      </script>
   </head>
   <body>
</body>
</html>
```

```
// Αρχείο message.js
alert('Ο Κόσμος μας!');
```

Το πρώτο μας παράδειγμα σε Javascript

```
<!DOCTYPE html>
<html>
    <head>
        <title>Τίτλος ιστοσελίδας</title>
        <meta charset="utf-8">
    </head>
    <body>
        <script type="text/javascript">  // ετικέτα αρχής
            alert('Ο Κόσμος μας!');      // Συνάρτηση alert
            document.write('Ενας Ψεύτικος Κόσμος');
        </script>                        // Ετικέτα τέλους
    </body>
</html>
```

Αυτό το παράδειγμα εμφανίζει ένα πλαίσιο μηνύματος με το κείμενο "**Ο Κόσμος μας!**" και το μήνυμα "**Ενας Ψεύτικος Κόσμος**" στην οθόνη μας.
Πρέπει να περιβάλουμε τον κώδικα javascript με τις ετικέτες αρχής και τέλους ώστε το πρόγραμμα περιήγησης να το εκτελέσει, ως JavaScript.
Η JavaScript καλεί την συνάρτηση **alert**, και περνάει σε αυτή το κείμενο που βρίσκεται μέσα στα μονά εισαγωγικά. Μετά εκτελεί την εντολή **document.write** Για να τυπώσει το κείμενο στην οθόνη. Το ερωτηματικό, δηλώνει το τέλος της εντολής και δεν μπορεί να παραλειφθεί.

Μεταβλητές στην Javascript

Όπως όλες οι γλώσσες προγραμματισμού έτσι και η Javascript υποστηρίζει τον μηχανισμό προσωρινής αποθήκευσης δεδομένων στην μνήμη RAM, δηλαδή τον μηχανισμό των μεταβλητών. Οι τύποι δεδομένων που υποστηρίζονται από αυτήν είναι:

Οι τύποι δεδομένων που υποστηρίζονται:

- Αριθμοί (ακέραιοι – δεκαδικοί)
- Αλφαριθμητικά (strings)
- Λογικές (true–false)
- null (κενή μεταβλητή)
- undefined (μη προσδιορισμένη)

Παραδείγματα δημιουργίας μεταβλητών:

- var number = 8;
- var pi = 3.14;
- var book_title = “Internet programming ”;
- var book_year = “2009”;
- var booked = FALSE;
- var temp = null;
- var feature = undefined;

Παράδειγμα δήλωσης μεταβλητής

```
<!DOCTYPE html>
<html>
    <head>
        <title>Τίτλος ιστοσελίδας</title>
        <meta charset="utf-8">
    </head>
    <body>
        <script type="text/javascript">   // ετικέτα αρχής
            var onoma = 'Κώστας';         // Δήλωση μεταβλητής
        </script>                        // Ετικέτα τέλους
    </body>
</html>
```

Οι μεταβλητές στην Javascript δημιουργούνται με την *δεσμευμένη λέξη var, το όνομα της μεταβλητής, το σύμβολο εκχώρησης = και την τιμή που θέλουμε να εκχωρήσουμε.* Ο τύπος δεδομένων της μεταβλητής ορίζεται αυτόματα από τον τύπο δεδομένων της εκχωρούμενης τιμής. Η χρήση της var, δεν είναι πάντα απαραίτητη. Θα δούμε αργότερα, πότε θα πρέπει να τη χρησιμοποιήσετε.

Δοκιμάστε το μόνοι σας. Αποθηκεύστε το όνομα σας σε μια μεταβλητή με το όνομα onoma. Αφού έχετε δημιουργήσει τη μεταβλητή, εμφανίστε την χρησιμοποιώντας τη συνάρτηση alert.

Άσκηση 1

```
<html>
    <head>
        <title>Τίτλος ιστοσελίδας</title>
        <meta charset="utf-8">
    </head>
    <body>
        <script type="text/javascript">
// Αυτό είναι ένα σχόλιο μιας γραμμής.
// Τα σχόλια δεν εκτελούνται από την javascript
            var onoma= 'Άριστος';
            alert(onoma);
 /*
Αν θέλουμε να εισάγουμε ένα σχόλιο πολλών γραμμών,
θα πρέπει να το περιβάλουμε από /* και */
  */

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

Εμφάνιση πολλών μεταβλητών μαζί

```
<html>
    <head>
        <title>Τίτλος ιστοσελίδας</title>
        <meta charset="utf-8">
    </head>
    <body>
        <script type="text/javascript">
            onoma = 'Άριστος';
            epitheto = 'Πασιάς';
            alert(onoma + ' ' + epitheto);
        </script>
    </body>
</html>
```

Με τη χρήση του χαρακτήρα συν (+) μπορούμε να εμφανίσουμε όσες μεταβλητές και τιμές θέλουμε. Εδώ εμφανίζουμε την μεταβλητή onoma, μετά ένα κενό , και τέλος την μεταβλητή epitheto.

Αντικατάσταση κειμένου

```
<script type="text/javascript">
  minima = 'Τα αγαπημένα μου χρώματα είναι πορτοκαλί, μαύρο και
            πράσινο.';
  alert(minima);
</script>
```

```
<script type="text/javascript">
  minima = 'Τα αγαπημένα μου χρώματα είναι πορτοκαλί, μαύρο και
            πράσινο.';

  // Θα αντικαταστήσουμε το πορτοκαλί με το κίτρινο
  minima = minima.replace('πορτοκαλί', 'κίτρινο');
  alert(minima);
</script>
```

Παρατηρήστε τον τρόπο με τον όποιο καλέσαμε την εντολή replace.

Η replace παίρνει δυο (2) παραμέτρους εισόδου.

- Το κείμενο που θα αντικατασταθεί. (πορτοκαλί).
- Το κείμενο που θα αντικαταστήσει το κείμενο που βρέθηκε. (κίτρινο).

Θα πρέπει να έχουμε κατά νου ότι το κείμενο που περνάει ως παράμετρος, έχει διάκριση πεζών-κεφαλαίων.

Εμφάνιση μέρους συμβολοσειράς

```
<script type="text/javascript">
  onoma_epitheto = 'Άριστος Πασιάς';
  onoma = onoma_epitheto.substr(0, 7);
  alert(onoma);
</script>
```

Η εντολή substr δέχεται δυο (2) παραμέτρους:

- Η αρχική θέση για την εξαγωγή μέσα στη συμβολοσειρά. Πρέπει να σημειωθεί ότι ο πρώτος χαρακτήρας βρίσκεται στη θέση 0, ο δεύτερος στη θέση 1 κτλ.
- Το πλήθος των χαρακτήρων για εξαγωγή. Στην περίπτωση μας 7, επειδή η λέξη Άριστος έχει 7 γράμματα.

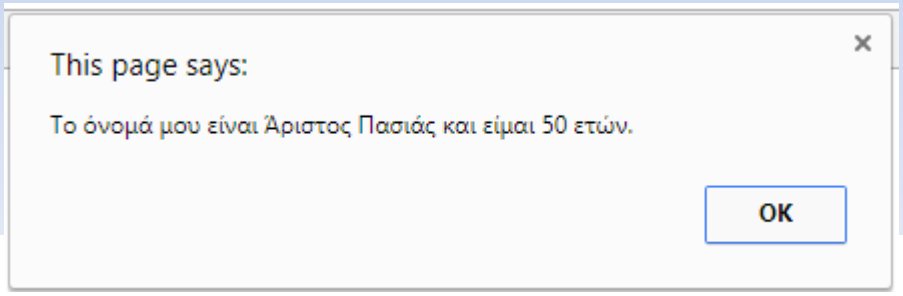
Έλεγχος μεγέθους συμβολοσειράς

```
<script type="text/javascript">
  onoma = 'Άριστος';
  alert(onoma.length);  // Το αποτέλεσμα θα είναι 7
</script>
```

```
<script type="text/javascript">
  password = '';
  if (password.length > 0 && password.length < 8)
  {
    alert('Ο κωδικός είναι πολύ μικρός');
  }
    else
  {
    alert('εισάγετε κωδικό')
  }
</script>
```

Δήλωση Αριθμητικής μεταβλητής

```
<!DOCTYPE html>
<html>
    <head>
        <title>Τίτλος ιστοσελίδας</title>
        <meta charset="utf-8">
    </head>
    <body>
        <script type="text/javascript">
            ilikia = 50;
            alert(ilikia);
        </script>
    </body>
</html>
```



Δοκιμάστε το μόνοι σας, δηλώστε τρείς μεταβλητές με το όνομα, το επώνυμο και την ηλικία σας και εμφανίστε το μήνυμα ακριβώς όπως φαίνεται στην πιο πάνω εικόνα. Κάντε το ίδιο με την εντολή document.write.

Άσκηση 2

```
<!DOCTYPE html>
<html>
  <head>
    <title>Τίτλος ιστοσελίδας</title>
    <meta charset="utf-8">
  </head>
  <body>
     <script type="text/javascript">
        onoma = 'Άριστος';
        epitheto = 'Πασιάς';
        ilikia = 50;

        alert("Το όνομά μου είναι " + onoma + ' ' +   epitheto +
        " και είμαι " + ilikia + " ετών.");

        document.write("Το όνομά μου είναι " + onoma + ' ' +
        epitheto + " και είμαι " + ilikia + " ετών.");

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

Υπολογισμοί με αριθμούς

```
<script type="text/javascript">
  // Πρόσθεση
  alert(1 + 1);

  // Αφαίρεση
  alert(20 - 10);

  // Πολλαπλασιασμός
  alert(5 * 5);

  // Διαίρεση
  alert(100 / 5);

  // Σύνθετος υπολογισμός
 alert(((1 + 10) * 3) / 11);
</script>
```

2
10
25
20
3

Δοκιμάστε το μόνοι σας να εμφανίσετε το αποτέλεσμα διαφόρων αριθμητικών πράξεων. Τέλος δοκιμάστε την ακόλουθη αριθμητική πράξη: $(8*(2+3)-5(\frac{8-3}{24-20})$

Υπολογισμοί με μεταβλητές

```
<script type="text/javascript">
  a = 10;
  b = 11;
  c = 2;
  alert((a + b) * c); // το οποίο στην πραγματικότητα είναι:
                          alert((10 + 11) * 2);
</script>
```

Δοκιμάστε το μόνοι σας να εμφανίσετε το αποτέλεσμα διαφόρων αριθμητικών πράξεων στις οποίες θα συνδυάσετε αριθμούς και μεταβλητές

Αριθμητικοί Τελεστές

Οι βασικοί τελεστές που χρησιμοποιούνται για τη επεξεργασία μεταβλητών είναι οι: +, -. *, / και η χρήση τους φαίνεται στα παρακάτω παραδείγματα (τα σύμβολα // προσθέτουν σχόλια στον κώδικα).

- number = number + 8; // αυξάνεται η τιμή της number κατά 8
- number += 2; // αυξάνεται η τιμή της number κατά 2
- book_title += internet; //Η μεταβλητή book_title θα αποκτήσει την τιμή "Internet"
- number = number – 2; // Η τιμή του number μειώνεται κατά 2
- number = number * 3; // Η τιμή του number πολλαπλασιάζεται με το 3
- number = number / 4; // Η τιμή του number διαιρείται με το 4
- number ++ // Αυξάνει την τιμή του number κατά 1
- number -- // Μειώνει την τιμή του number κατά 1

Οι τελεστές της Javascript

Όπως σε όλες τις γλώσσες προγραμματισμού έτσι και στην Javascript υπάρχει η δομή επιλογής. Πριν προχωρήσουμε στην περιγραφή αυτής της δομής θα περιγράψουμε τους τελεστές σύγκρισης και τους λογικούς τελεστές της γλώσσας.

Τελεστές σύγκρισης

- < , μικρότερο
- > , μεγαλύτερο
- <= , μικρότερο ή ίσο
- >= , μεγαλύτερο ή ίσο
- = = , ίσο (προσοχή να μην συγχέεται με τον τελεστή εκχώρησης =)
- != , όχι ίσο
- = = = , αυστηρά ίσο (ίσο σε τιμή και σε τύπο δεδομένων)
- != = , αυστηρά μη ίσο (μη ίσο σε τιμή και σε τύπο δεδομένων)

Λογικοί τελεστές

- && , λογικός τελεστής ΚΑΙ
- | | , λογικός τελεστής Η'
- ! , λογικός τελεστής ΟΧΙ (αντιστρέφει μια λογική πρόταση)
- ^, λογικός τελεστής XOR

Η εντολή prompt

Παράδειγμα:

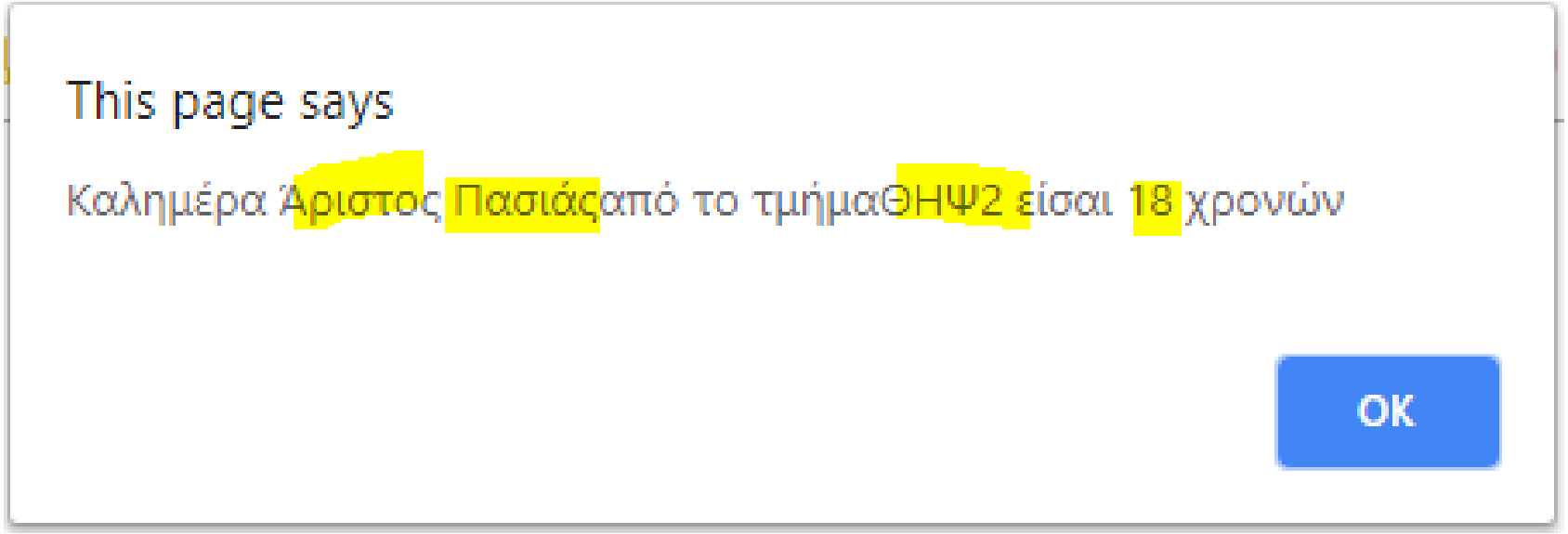
```
var name= prompt("Γράψε το όνομα σου", "Άριστος");
var sname= prompt("Γράψε το επίθετο σου", "Πασιάς");

var txt = ("Καλημέρα " + name+ " " +sname");
```

Άσκηση

Να γράψετε τον απαραίτητο κώδικα σε JavaScript ώστε όταν φορτώνεται μια ιστοσελίδα, να ζητά από τον επισκέπτη να καταχωρήσει :

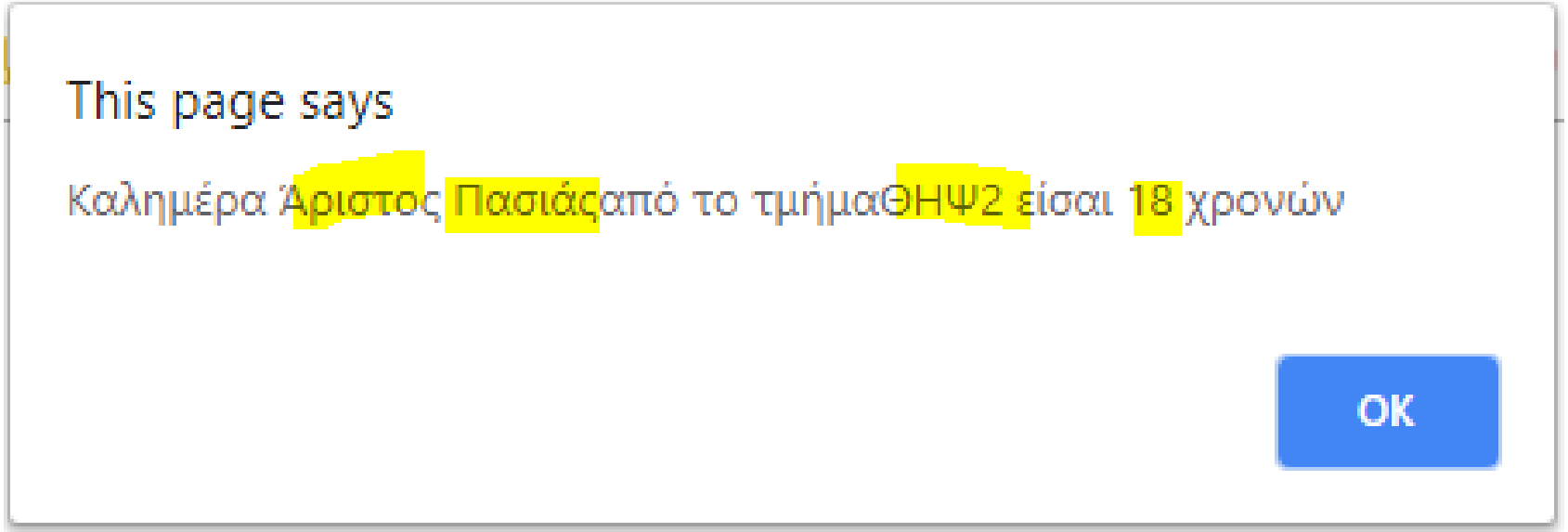
- Όνομα
- Επίθετο
- Τμήμα
- Έτος γέννησης
- Τρέχον έτος

Μόλις δώσει όλα τα στοιχεία να του εμφανίζει το πιο κάτω μήνυμα:



Άσκηση

```
<script type="text/javascript">
        var myclass= prompt("Γράψε το τμήμα σου", "ΘΗΨ2");
        var name= prompt("Γράψε το όνομα σου", "Άριστος");
        var sname= prompt("Γράψε το επίθετο σου", "Πασιάς");
        var gen1= prompt("Έτος γέννησης;", "2000");
        var gen =Number(gen1);
        var thisyear=2018;
        var ilikia = (thisyear-gen);
        var txt =("Καλημέρα " + name+ " " +sname +"από το τμήμα" +myclass+ "
                είσαι "+ ilikia+ " χρονών");
        alert (txt)
</script>
```



Δομή επιλογής (Συνθήκη if)

```
<script type="text/javascript">
  bathmos = 65;
    if (bathmos < 70)  // Εάν ο βαθμός είναι μικρότερος από 70
  {
    alert('Απέτυχες!'); // Ο φοιτητής έχει αποτύχει
  }
</script>
```

Σε αυτή την εκδοχή η δομή if ελέγχει εάν ισχύει η συνθήκη μέσα στις παρενθέσεις ().

- Σε περίπτωση που ισχύει **εκτελείται** ο κώδικας ανάμεσα στις αγκύλες {}.
- Στην αντίθετη περίπτωση **δεν εκτελείται** ο κώδικας ανάμεσα στις {} και η ροή του προγράμματος συνεχίζει μετά την δομή if.

Δομή επιλογής (Συνθήκη if - else)

```
//Κάποιες φορές είναι χρήσιμος ο συνδυασμός της εντολής If με
το else
<script type="text/javascript">
  bathmos = 65;
    if (bathmos < 70)  // Εάν ο βαθμός είναι μικρότερος από 70
  {
    alert('Απέτυχες!'); // Ο φοιτητής έχει αποτύχει
  }
    else  // Στις υπόλοιπες περιπτώσεις,
  {
    alert('Πέρασες');   // περνάει την τάξη.
  }
</script>
```

Σε αυτή την εκδοχή η δομή if ελέγχει εάν ισχύει η συνθήκη μέσα στις παρενθέσεις.

- Σε περίπτωση που ισχύει εκτελείται μόνο ο κώδικας ανάμεσα στις πρώτες αγκύλες {}.
- Στην αντίθετη περίπτωση εκτελείται μόνο ο κώδικας ανάμεσα στις δεύτερες αγκύλες else {}.
- Η ροή του προγράμματος συνεχίζει μετά την δομή if.

Πολλαπλές συνθήκες

```
<script type="text/javascript">
        bathmos = 80;
        if (bathmos >= 90)
        {
        alert('Άριστα!');
        }
        else if (bathmos >= 70 && bathmos < 90);
        {
        alert('Πολύ καλά!');
        }
        else
        {
        alert('Πρέπει να διαβάσεις.');
        }
</script>
```

Το && σημαίνει ΛΟΓΙΚΟ ΚΑΙ. Με αυτό εννοούμε ότι και οι δυο προϋποθέσεις πρέπει να πληρούνται. Μπορούμε να προσθέσουμε όσες συνθήκες χρειαζόμαστε.

Συνθήκες με λογικές μεταβλητές

```
<script type="text/javascript">
  xroma = 'Μαύρο';
  if (xroma == 'Άσπρο')
  {
    einaiAspro = true;
  }
  else
  {
    einaiAspro = false;
  }
  if (einaiAspro == true)
  {
    alert('Το χρώμα είναι άσπρο!');
  }
  else
  {
    alert('Το χρώμα δεν είναι άσπρο');
  }
</script>
```

JavaScript switch statement

```
switch(expression) {
        case value1:
                code to be executed;
        break;
        case value2:
                code to be executed;
         break;
        ……

        default:
                code to be executed if above values are not matched;
}
```

JavaScript switch statement Παράδειγμα

```
<script>
var grade='B';
var result;
.....
switch(grade){
        case 'A':
        result="A Grade";
break;
        case 'B':
        result="B Grade";
break;
        case 'C':
        result="C Grade";
break;
        default:
        result="No Grade";
}
document.write(result);
</script>
```

Ασκήσεις Επανάληψης

Δημιουργία / επεξεργασία πινάκων

Η Javascript επιτρέπει την δημιουργία πινάκων, δηλαδή την προσωρινή αποθήκευση στην μνήμη RAM πολλαπλών δεδομένων με το ίδιο όνομα. Μπορούμε να ξεχωρίσουμε τα δεδομένα με την χρήση δείκτη. Στην Javascript δεν είναι απαραίτητο όλα τα στοιχεία ενός πίνακα να είναι του ίδιου τύπου. Συγκεκριμένα ένας πίνακας στην Javascript δημιουργείται με τους τρεις εναλλακτικούς τρόπους όπως φαίνεται παρακάτω:

Α) var Books = new Array();

//δημιουργείται ο κενός πίνακας Books

Β) var Books = new Array("Learning JAVA", "Data structures in C++", "Arduino projects");

//δημιουργείται ο πίνακας Books τριών θέσεων όπου σε κάθε θέση έχει τον τίτλο του κατά σειρά βιβλίου. Η αρίθμηση των θέσεων του πίνακα ξεκινάει από το 0. Έτσι το πρώτο στοιχείο είναι το Books[0], το δεύτερο το Books[1] και το τρίτο το Books[2].

Γ) var Books = ["Learning JAVA", "Data structures in C++", "Arduino projects"];

//Ισοδύναμος χειρισμός με το παράδειγμα Β.

Άλλες λειτουργίες πινάκων

Α) document.write(Books[1]); //θα τυπωθούν στην ιστοσελίδα τα περιεχόμενα της δεύτερης θέσης του πίνακα Books.

Β) Books[2] = “Raspberry PI in action”; //τα περιεχόμενα της τρίτης θέσης του πίνακα Books θα αλλάξουν σε “Raspberry PI in action” (εντολή εκχώρησης).

Γ) Books.length = 3; //Διαγραφή των στοιχείων από την θέση 3 μέχρι το τέλος.

Δ) Books.reverse(); //αντιστρέφει την σειρά των στοιχείων.

Ε) Books.push(“Internet of Things”); //εισάγει στο τέλος του πίνακα ένα νέο στοιχείο.

Δουλεύοντας με πίνακες

```
<script type="text/javascript">
  a1 = 1;
  a2 = 5;
  a3 = 10;
  a4 = 15;
  a5 = 20;
  a6 = 25;
  a7 = 30;
  a8 = 35;
  a9 = 40;
  a10 = 45;
  alert(a1);
</script>
```

```
<script type="text/javascript">
  a = new Array(1, 5, 10, 15, 20, 25, 30, 35, 40, 45);
  alert(a[0]);
</script>
```

Προσθήκη / επεξεργασία στοιχείων πίνακα

```
<script type="text/javascript">
  a = new Array(1, 5, 10, 15, 20, 25, 30, 35, 40, 45);
  alert(a[0]);

  a.push(60);

  a[10] = 50;

  megethos = a.length;
  alert(megethos);    // Τι ακριβώς θα εμφανίσει η εντολή alert;
</script>
```

Δομή επανάληψης for στην Javascript

Η Javascript διαθέτει δύο δομές επανάληψης την while και την for. Η while χρησιμοποιείται όταν, κατά την σχεδίαση-συγγραφή του προγράμματος, δεν είναι γνωστός ο αριθμός των επαναλήψεων ενώ η for χρησιμοποιείται όταν αυτός ο αριθμός είναι γνωστός.

```
for (i = 0; i <= 10; i = i + 1)
{
  alert(i);
}
```

Αρχική τιμή; Συνθήκη; Βήμα

Δομή επανάληψης for

Στην δομή επανάληψης for χρησιμοποιείται μια μεταβλητή που έχει τον ρόλο του μετρητή των επαναλήψεων. Το τμήμα ελέγχου της δομής επανάληψης, που βρίσκεται ανάμεσα στις παρενθέσεις, αποτελείται από τρία τμήματα

α) Αρχικοποίηση της μεταβλητής-μετρητή (δίνεται η αρχική τιμή της μεταβλητής) var i = 0;

β) Συνθήκη ελέγχου μεταβλητής-μετρητή (λογική συνθήκη, i < = 10, που περιέχει οπωσδήποτε την μεταβλητή-μετρητής, όσο ισχύει η συνθήκη επαναλαμβάνεται η εκτέλεση των εντολών που βρίσκονται ανάμεσα στις αγκύλες {})

γ) ρυθμός μεταβολής μεταβλητής-μετρητή σε κάθε επανάληψη i++ ή i=i+1 (στο τμήμα αυτό δηλώνεται το πόσο θα αυξάνεται η μεταβλητή-μετρητής σε κάθε επανάληψη).

Δομή επανάληψης for στην Javascript
Παράδειγμα

```
<script>
for (i=1; i<=5; i++)
{
document.write(i + "<br/>")
}
</script>
```

Δομή επανάληψης for

Δομή επανάληψης στην while στη Javascript

Ο βρόχος while παίρνει μόνο μία παράμετρο. Μια έκφραση που όταν αξιολογείται ως false, ο βρόχος τερματίζεται. Προσέξτε ότι ο βρόχος while δεν προσαυξάνει αυτόματα κάποιο μετρητή. Στην πραγματικότητα, δεν υπάρχει καθόλου μετρητής. Εμείς πρέπει να φροντίσουμε, προκειμένου να αποφευχθεί ένας ατέρμονας βρόχος. Σε αυτό το παράδειγμα ελέγχεται αν η μεταβλητή i είναι μικρότερη από 15, εάν είναι, τυπώνεται το στοιχείο i. Σε κάθε επανάληψη η i αυξάνεται κατά 1(i++;). Με αυτόν τον τρόπο τυπώνονται όλοι οι αριθμοί από 11 μέχρι 15.

```
<script>
var i=11;
while (i<=15)
{
document.write(i + "<br/>");
i++;
}
</script>
```

Δομή επανάληψης while

Δομές Επανάληψης: For Vs While

```
<script type="text/javascript">
  for (i = 1; i <= 10; i = i + 1) //Τρείς παράμετροι
  {
    alert(i);
  }
</script>
```

Ο βρόχος **for**. Χρησιμοποιείται κυρίως όταν γνωρίζουμε τον αριθμό των επαναλήψεων που χρειαζόμαστε.

```
<script type="text/javascript">
  var books = new array ("Turbo Pascal", "Learn Java", ..)
  i = 0;
  while (i <= books.lenght) //Μία μόνο παράμετρος
  {
    alert(books[i]);
    i++;
  }
</script>
```

Ο βρόχος **while**. Χρησιμοποιείται κυρίως όταν **δεν** γνωρίζουμε τον αριθμό των επαναλήψεων που χρειαζόμαστε.

Δομή επανάληψης do while στην Javascript

```
do {
   code to be executed
}while (condition);
```

```
<script>
var i=21;
do{
document.write(i + "<br/>");
i++;
}while (i<=25);
</script>
```

Δομή επανάληψης do while

Πίνακες και επαναλήψεις

```
<script type="text/javascript">
  a = new Array(10, 15, 32, 99, 21, 9, 92);
</script>
```

Σκεφτείτε ότι έχουμε τον πιο πάνω πίνακα. Θέλουμε να υπολογίσουμε το άθροισμα όλων των τιμών του. Λογικά θα πρέπει να εφαρμόσουμε ένα βρόχο επανάληψης (*for ή else*).

- Να θυμάστε ότι ο αριθμός των στοιχείων σε ένα πίνακα, επιστρέφεται χρησιμοποιώντας τη ιδιότητα ***length***.
- Επίσης, να θυμάστε, ότι οι δείκτες του πίνακα αρχίσουν από το μηδέν (0), και τελειώνουν σε ***pinakas.length - 1***.

Στο notepad++ να γράψετε τον απαραίτητο κώδικα για να εμφανίστε το άθροισμα των τιμών του πίνακα a.

```
<script type="text/javascript">
  a = new Array(10, 15, 32, 99, 21, 9, 92);
  athroisma = 0;
  for (i = 0; i < a.length; i = i + 1)
  {
    athroisma = athroisma + a[i];
  }
  alert(athroisma);
</script>
```

Συναρτήσεις

Όταν χρησιμοποιούμε το ίδιο τμήμα κώδικα συχνά, αντί να τον αντιγράφουμε συνεχώς, μπορούμε να τον μετατρέψουμε σε μια συνάρτηση.
Οι συναρτήσεις είναι τμήματα κώδικα που εκτελούν μια συγκεκριμένη εργασία

```
<script type="text/javascript">
  function Hello()
  {
    alert('Καλοσορίσατε στο ηλεκτρονικό μας κατάστημα!');
  }
  Hello();
//...Σε οποιοδήποτε σημείο του κώδικα απλά εκτελώ τη συνάρτηση
  Hello();
</script>
```

```
//Συνάρτηση με παράμετρο εισόδου
<script type="text/javascript">
  function sqroot(n)
  {
    echo (n * n);
  }
    sqroot(5);
</script>
```

Πολλαπλές παράμετροι συναρτήσεων

```
<script type="text/javascript">
 function bigerNo() {
    if (a > b)
    {
      alert('Ο πρώτος αριθμός είναι μεγαλύτερος από τον
δεύτερο.');
    }
    else if (a < b)
    {
      alert('Ο δεύτερος αριθμός είναι μεγαλύτερος από τον
πρώτο.');
    }
    else
    {
      alert('Οι δύο αριθμοί είναι ίσοι.');
    }
  }

  bigerNo(1, 10);
</script>
```

Ιδιότητες Αντικειμένων στη Javascript

```
// Άμεσος ορισμός αντικειμένου
<script>
        emp={id:102,name:“Γιάννης Άνυφτος",salary:40000}
        document.write(emp.id+" "+emp.name+" "+emp.salary);
</script>
```

```
<script>
        var emp=new Object();
        emp.id=101;
        emp.name=«Κώστας Αχαίρευτος";
        emp.salary=50000;
        document.write(emp.id+" "+emp.name+" "+emp.salary);
</script>
```

Αλληλεπίδραση Javascript με στοιχεία HTML (DOM)

Η Javascript αλληλεπιδρά με ετικέτες της HTML μέσω του μοντέλου DOM (Document Object Model). Το DOM αποτελεί μια ιδεατή δομή της ιστοσελίδας έτσι ώστε η Javascript να έχει πρόσβαση στα στοιχεία της. Η βασική εντολή της DOM (με τις παραλλαγές της) είναι η:

```
document.getElementById("someId").value;

document.getElementById("someId").name;

document.getElementById("someId").innerHTML;
```

Η εντολή document.getElementById("someId") επιστρέφει το στοιχείο της HTML που έχει την ιδιότητα id ίση με "someId". Γι αυτό το λόγο στα στοιχεία της HTML που θέλουμε να αλληλεπιδρούν με την Javascript ορίζουμε την ιδιότητα id. Προσθέτοντας την ιδιότητα .value η εντολή επιστρέφει την τιμή της ιδιότητας value του στοιχείου HTML. Ομοίως συμβαίνει προσθέτοντας την ιδιότητα .name. Ενώ προσθέτοντας την ιδιότητα innerHTML η εντολή επιστρέφει το περιεχόμενο του στοιχείου. Ακολουθεί παράδειγμα.

Αλληλεπίδραση Javascript με στοιχεία HTML (DOM)

```
<html>
 <head>
  <title> Βιβλιοθήκη </title>
 </head>
<body>
    <H1 id="et1"> Το WebSite της βιβλιοθήκης μας </H1>
    <H2 id="et2"> Εδώ θα βρήτε τα καλύτερα βιβλία</H1>
    <p id="part1">Αναζήτηση Βιβλίων</p>
<script type="text/javascript">
    var s1 = document.getElementByid("et1").innerHTML;
    document.write(s1+ "<br />");
    var s2 = document.getElementByid("et2").innerHTML;
    document.write(s2+ "<br />");
    var s3 = document.getElementByid("part1").innerHTML;
    document.write(s3+ "<br />");
    document.getElementByid("part1").innerHTML = ("Μας
    τελείωσαν τα βιβλία");
    var s4 = document.getElementByid("part1").innerHTML;
</script>
</body>
</html>
```

Αλληλεπίδραση Javascript με στοιχεία HTML (DOM)

Μια άλλη εντολή του DOM που χρησιμοποιείται για διασύνδεση στοιχείων φόρμας <FORM> είναι η:

document.form[0].element[0].value/name/innerHTML;

Το form[0] αναφέρεται στην πρώτη φόρμα της ιστοσελίδας, το form[1] στην δεύτερη και γενικότερα το form[n] στην n+1 φόρμα για την περίπτωση που έχουμε παραπάνω από μια φόρμες σε μια ιστοσελίδα. Συνήθως έχουμε μια φόρμα οπότε το form[0] αρκεί. Το element[0] αναφέρεται στο πρώτο στοιχείο της φόρμας (π.χ. εάν αυτό είναι πλαίσιο κειμένου, τότε θα αναφέρεται σε αυτό), το element[1] στο δεύτερο στοιχείο της φόρμας και γενικότερα το element[n] στο n+1 στοιχείο της φόρμας. Οι ιδιότητες value/ name/ innerHTML έχουν τις ίδιες λειτουργίες όπως περιγράφηκαν στο παράδειγμα της getElementById.

Διαχείριση γεγονότων στην Javascript

Στις προηγούμενες ενότητες μάθαμε να δημιουργούμε προγράμματα σε Javascript και να τα εκτελούμε φορτώνοντας την ιστοσελίδα στον φυλλομετρητή. Η εκτέλεση αυτή γινόταν με την φόρτωση της ιστοσελίδας.
Πολλές φορές όμως είναι απαραίτητο ενέργειες-κώδικας να εκτελούνται όταν υπάρχει διάδραση με τον χρήστη. Με άλλα λόγια είναι απαραίτητο να εκτελείται κώδικας σε Javascript όταν συμβαίνουν κάποια γεγονότα στην ιστοσελίδα. Τέτοια γεγονότα μπορεί να είναι το πάτημα ενός κουμπιού, το πάτημα σε ένα πλαίσιο κειμένου , η απομάκρυνση από ένα πλαίσιο κειμένου κ.α.

Διαχείριση γεγονότων στην Javascript

Μερικά από τα ποιο συνηθισμένα γεγονότα είναι:

- onclick, συμβαίνει όταν πατηθεί κουμπί
- onfocus, συμβαίνει όταν πατηθεί πλαίσιο κειμένου
- onblur, συμβαίνει όταν γίνει απομάκρυνση από ένα στοιχείο ελέγχου φόρμας
- onkeyup, συμβαίνει όταν αφήνουμε ένα πλήκτρο

Πριν προχωρήσουμε ας θυμηθούμε πρώτα την λειτουργία της συνάρτησης (function)

```
function welcome(){
window.alert("Welcome to our library");
}

welcome();
```

Διαχείριση γεγονότων στην Javascript

Η φόρμα στο παράδειγμα αποτελείται από ένα πλαίσιο εισαγωγής κειμένου και ένα κουμπί. Στο πλαίσιο εισαγωγής κειμένου υπάρχει η ιδιότητα διαχείρισης γεγονότος **onfocus** η οποία παίρνει τιμή ίδια με το όνομα της συνάρτησης **welcome().** Ομοίως στο κουμπί έχουμε την ιδιότητα διαχείρισης γεγονότος **onclick** Αυτό σημαίνει ότι όταν συμβεί το γεγονός πατήματος μέσα στο πλαίσιο κειμένου, ή το γεγονός πατήματος του κουμπιού θα εκτελεστεί ο κώδικας της συνάρτησης welcome().

```
<HTML>
  <HEAD>
   <TITLE>Βιβλιοθήκη</TITLE>
  </HEAD>
  <BODY>
  <H1 id="et1"> Our library site</h1>

  <FORM>
  ENTER YOUR NAME: <INPUT type="text" onfocus="welcome();" value="">
  <INPUT type="button" onclick="welcome();" value="OK">
  </FORM>

  <SCRIPT TYPE="text/javascript">
  function welcome(){
  window.alert("Welcome to our library");
  }
  </SCRIPT>

  </BODY>
</HTML>
```

Έλεγχος ορθής συμπλήρωσης φόρμας με Javascript

Στο πρώτο βήμα σχεδιάζεται η φόρμα εισαγωγής στοιχείων στην HTML. Κάθε στοιχείο της φόρμας (πλαίσια κειμένου) έχει εισαχθεί σε ένα πίνακα. Επίσης έχει προστεθεί και μια τρίτη στήλη στον πίνακα (κενή προς το παρόν) για την προβολή μηνυμάτων-οδηγιών στον χρήστη ανά πεδίο εισαγωγής. Αξιοσημείωτο είναι ότι σε όλα τα πλαίσια κειμένου αλλά και στα κενά πεδία της τρίτης στήλης του πίνακα έχει προστεθεί η ιδιότητα id (για να μπορούν να είναι διαχειρίσιμα από το DOM). Όσον αφορά τα πλαίσια κειμένου έχει προστεθεί και η ιδιότητα-γεγονός onkeyup (όταν αφήνουμε ένα πλήκτρο) στην οποία εκχωρείται η συνάρτηση checking();

```
<HTML>
  <HEAD>
   <TITLE>Βιβλιοθήκη</TITLE>
  </HEAD>
  <BODY>
  <H1 id="et1"> Our library site</h1>

  <FORM>
    <TABLE border="1">
     <tr><td>Book Title</td><td><INPUT type="text" id="book_t" onkeyup="checking();" value=""></td><td id="comments_t"></td></tr>
     <tr><td>Book ISBN</td><td><INPUT type="text" id="book_i" onkeyup="checking();" value=""></td><td id="comments_i"></td></tr>
     <tr><td>Book Author</td><td><INPUT type="text" id="book_a" onkeyup="checking();" value=""></td><td id="comments_a"></td></tr>
    </TABLE>
     <INPUT type="button" id="button1" value="OK">
  </FORM>
  </BODY>
</HTML>
```

Έλεγχος ορθής συμπλήρωσης φόρμας με Javascript

Το επόμενο βήμα είναι να γραφεί κώδικας σε Javascript (για το γεγονός όταν αφήνουμε πλήκτρο) που θα ελέγχει κάθε πλαίσιο κειμένου. Αν το πλαίσιο κειμένου είναι κενό θα εμφανίζεται στο διπλανό κελί του πινάκα μήνυμα, που θα προτρέπει το χρήστη να εισάγει κατάλληλη τιμή. Διαφορετικά θα εμφανίζεται το μήνυμα “OK”.

```
<SCRIPT TYPE="text/javascript">
function checking(){
var elements = new Array();
var displays = new Array();
var messages = new Array();

elements[0] = document.getElementById("book_t").value;
elements[1] = document.getElementById("book_i").value;
elements[2] = document.getElementById("book_a").value;

displays[0] = "comments_t";
displays[1] = "comments_i";
displays[2] = "comments_a";

messages[0] = "Please insert book title";
messages[1] = "Please insert book ISBN";
messages[2] = "Please insert book Author";

  for (var i = 0; i < elements.length; i++){
    if (elements[i]==""){
       document.getElementById(displays[i]).innerHTML = messages[i];
      }else{
       document.getElementById(displays[i]).innerHTML = "OK";
       }
  }

}
</SCRIPT>
```

Με το DOM η ιδιότητα value του κάθε πλαισίου κειμένου εισάγεται σε ένα πίνακα (elements). Στον πίνακα displays εισάγονται τα id των κελιών του πίνακα όπου θα εμφανίζονται τα μηνύματα ενώ στον πίνακα messages αποθηκεύονται τα μηνύματα. Κατόπιν για κάθε πλαίσιο κειμένου (for) γίνεται έλεγχος αν ο χρήστης δεν έχει εισάγει δεδομένα (elements[i] == “”). Αν δεν έχει εισάγει, μέσω DOM documenet.getElementById(display[i]) .innerHTML=messages[i]; εκχωρείται στο κατάλληλο κελί το κατάλληλο μήνυμα. Αν έχει εισάγει, με παρόμοιο τρόπο εκχωρείται η τιμή “OK”.