



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

Προγραμματισμός Διαδικτύου Ασκήσεις Εργαστηρίου

Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ Νο 12

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

[http:// arch.ict.e.uowm.gr/mdasyg](http://arch.ict.e.uowm.gr/mdasyg)

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Έκδοση Σεπτέμβριος 2011

Περιεχόμενα

1. Σκοπός της άσκησης 4
2. Ερωτήσεις/Ασκήσεις..... 4

1. Σκοπός της άσκησης

Εργαστήριο Javascript#2:

- Javascript/HTML DOM.

2. Ερωτήσεις/Ασκήσεις

Δημιουργήστε ένα φάκελο με το όνομα lab12 μέσα στο htdocs του xampp και τοποθετήστε μέσα σε αυτόν τα αρχεία που είχαν δημιουργηθεί στο προηγούμενο εργαστήριο.

Επισκεφτείτε τη σελίδα <http://localhost/lab12> και επιβεβαιώστε ότι εμφανίζεται σωστά ο ιστοχώρος.

Τροποποίηση της φόρμας εγγραφής νέου χρήστη

Ανοίξτε το αρχείο **signup.php**.

- Επισκεφτείτε τη σελίδα **signup.php** και επιβεβαιώστε την ορθή σύνδεση με το να δείτε το παρακάτω παράθυρο (πριν φορτώσει η σελίδα, Εικόνα 1):



Εικόνα 1 - Επιβεβαίωση ορθής σύνδεσης

- Αφού επιβεβαιώσετε την ορθή λειτουργία της σελίδας, τοποθετήστε το σύμβολο του σχόλιου μπροστά από τη γραμμή **alert(...)** για να μην εκτελείται.
- Θα δημιουργήσουμε τη συνάρτηση **validate_required()** η οποία θα ελέγχει κάθε πεδίο της φόρμας και αν είναι κενό, τότε θα εμφανίζει ένα μήνυμα στο χρήστη. Σημειώστε ότι είναι καλύτερο να γίνει ο έλεγχος στον πελάτη προκειμένου να αποκλείσουμε τα προβλήματα χωρίς να αυξήσουμε το φόρτο στο διακομιστή. Όμως, επειδή η javascript μπορεί να παρακαμφθεί θα πρέπει να υπάρχει και ο αντίστοιχος έλεγχος στο server για τους ελάχιστους κακόβολους χρήστες. Σημειώστε ότι ο client-side έλεγχος είναι πολύ γρήγορος αφού δε χρειάζεται να μεταδοθούν δεδομένα στο διακομιστή.

Θα κατασκευάσουμε ένα script το οποίο θα ελέγχει τις τιμές που έχει δώσει ο χρήστης μόλις πατηθεί το κουμπί submit. Η συνάρτηση που θα κατασκευάσουμε θα ελέγχει κάθε πεδίο της φόρμας ως προς κάποια κριτήρια που θα έχουμε θέσει για αυτό το πεδίο. Αν εμφανιστεί κάποιο πρόβλημα τότε δε θα αποσταλούν τα στοιχεία στο διακομιστή αλλά θα εμφανιστεί ένα μήνυμα στο χρήστη.

- Θα χρησιμοποιήσουμε την εξής τεχνική: Μόλις εμφανίζεται κάποιο πρόβλημα **θα γράφεται το πρόβλημα δίπλα ακριβώς στο κελί της φόρμας** που έχει εμφανιστεί το πρόβλημα. Για να γίνει αυτό θα πρέπει να δημιουργήσουμε ένα καινούργιο πεδίο με διαφορετικά id ώστε να μπορέσουμε να προσδιορίσουμε κάθε φορά σε ποιο αντικείμενο θα τοποθετούμε το μήνυμα. Το πεδίο μπορεί να είναι είτε τύπου **<td>**

είτε τύπου `<div>`, είτε τύπου ``. Επειδή εμείς έχουμε χρησιμοποιήσει πίνακα για τη δημιουργία της φόρμας, θα χρησιμοποιήσουμε το πεδίο τύπου `<td>`. Θα πρέπει λοιπόν δίπλα σε κάθε πεδίο της φόρμας (αλλά στην ίδια γραμμή του πίνακα==μέσα στο ίδιο `<tr> ...</tr>`) να τοποθετήσουμε ένα ακόμη κελί ως εξής:

- Ανοίξτε το αρχείο `signup.php` και πηγαίνατε στις γραμμές που βρίσκεται η φόρμα.
- Δίπλα στο `<input..` πεδίο για το e-mail τοποθετήστε ένα ακόμη κελί με `id="inf_email"` ως εξής : `<td id="inf_email"> </td>` **Προσοχή: Πρέπει να υπάρχει κενό μέσα στο td γιατί διαφορετικά δε θα μπορεί να εμφανιστεί ένα μήνυμα εκεί πέρα.**
- Με παρόμοιο τρόπο τοποθετήστε δίπλα σε κάθε άλλο πεδίο της φόρμας τα κελιά με `id= inf_password, inf_username, inf_fullname, inf_birthyear, inf_acceptterms`

Η συνάρτηση `validate e-mail`

- Η πρώτη συνάρτηση που καλείται είναι η `validateEmail()`. Θα χρησιμοποιήσουμε μια κανονική έκφραση για αυτό. Ορίστε τη συνάρτηση `validateEmail` να δέχεται τρεις παραμέτρους ως εξής:

```
function validateEmail (
    valfield,    // element to be validated
    infofield,  // id of element to receive info/error msg
    required    // true if required
)
```

- Η συνάρτηση αυτή θα έχει το παρακάτω κυρίως σώμα. Ορίζουμε την κανονική έκφραση στη μεταβλητή `email` μέσα σε `var` και στη συνέχεια χρησιμοποιούμε το όνομα αυτής της μεταβλητής με το `.test(tfld)` το οποίο επιστρέφει TRUE αν ταιριάζει το `tfld` (το `tfld` είναι το όρισμα που έχει δώσει ο χρήστης, αφού περάσει από τη συνάρτηση `trim()`). Αν ταιριάζει τότε η εκτέλεση συνεχίζει κανονικά, διαφορετικά εκτυπώνει με το `msg(...)` το μήνυμα στο κατάλληλο `id`, που έχει δοθεί ως 2η παράμετρος.

```
{
var stat = commonCheck (valfield, infofield, required);
if (stat != proceed) return stat;

var tfld = trim(valfield.value); //whitespace trimmed off
var email = /^[^@]+@[^@.]+\.[^@]*\w\w$/ ;
if (!email.test(tfld)) {
    msg (infofield, "error", "ERROR: not a valid e-mail address");
    setfocus(valfield);
    return false;
}

var email2 = /^[A-Za-z][\w.-]+@\w[\w.-]+\.[\w.-]*[A-Za-z][A-Za-z]$/ ;
if (!email2.test(tfld))
    msg (infofield, "warn", "Unusual e-mail address - check if correct");
else
    msg (infofield, "warn", "OK");
return true;
}
```

- Η συνάρτηση αυτή αρχικά καλεί μια άλλη συνάρτηση που κατασκευάζουμε με το όνομα **commoncheck()**, η οποία κάνει κάποιους αρχικούς ελέγχους στην τιμή του πεδίου. Η συνάρτηση αυτή:
 - Αρχικά ελέγχει αν ο φυλλομετρητής υποστηρίζει την ιδιότητα DOM **document.getElementById**. Αν την υποστηρίζει τότε μπορούμε να γράψουμε σε ένα συγκεκριμένο πεδίο του εγγράφου μέσω javascript, διαφορετικά δε θα μπορέσουμε. Επίσης, γίνονται κάποιοι έλεγχοι ακόμη για το αν υποστηρίζονται και τα υπόλοιπα στοιχεία του DOM.
 - Στη συνέχεια ελέγχεται η 3η παράμετρος της συνάρτησης, η οποία καθορίζει αν απαιτείται η συγκεκριμένη παράμετρος ή όχι. Αν απαιτείται και είναι κενή τότε εμφανίζει το αντίστοιχο μήνυμα.

```
// -----
//      commonCheck
// Common code for all validation routines to:
// (a) check for older / less-equipped browsers
// (b) check if empty fields are required
// Returns true (validation passed),
//      false (validation failed) or
//      proceed (don't know yet)
// -----

var proceed = 2;

function commonCheck      (
    valfield, // element to be validated
    infofield, // id of element to receive msg
    required // true if required
)
{
  if (!document.getElementById)
    return true; // not available on this browser
  var elem = document.getElementById(infofield);
  if (!elem.firstChild) return true; // not available on this browser
  if (elem.firstChild.nodeType != node_text) return true; // wrong type

  if (emptyString.test(valfield.value)) {
    if (required) {
      msg (infofield, "error", "ERROR: required");
      setfocus(valfield);
      return false;
    }
  }
  else {
    msg (infofield, "warn", ""); // OK
    return true;
  }
}

return proceed;
}
```

- Η συνάρτηση **validateEmail()** χρησιμοποιεί επίσης τη συνάρτηση trim που κατασκευάζουμε εμείς παρακάτω. Η συνάρτηση αυτή έχει την ίδια λειτουργία με την trim της PHP, δηλαδή αφαιρεί τα κενά πριν και μετά. Η συνάρτηση αυτή δίνεται παρακάτω:

```
// -----
//      trim
// Trim leading/trailing whitespace off string
// -----

function trim(str)
{
    return str.replace(/^\s+|\s+$/g, "");
}
```

- Η συνάρτηση **validateEmail()** χρησιμοποιεί επίσης τη συνάρτηση **msg()** η οποία δέχεται 3 ορίσματα. Το πρώτο όρισμα αναφέρει το πεδίο HTML στο οποίο θα γράψουμε το μήνυμα, το δεύτερο όρισμα το πόσο σοβαρό λάθος είναι, και το τρίτο όρισμα το μήνυμα που θα εμφανιστεί.
- Η συνάρτηση αυτή γράφει στο πεδίο του οποίου το id δίνεται ως πρώτο όρισμα το μήνυμα που δίνεται ως τρίτη παράμετρος.
- Η συνάρτηση αυτή είναι η παρακάτω:

```
// -----
//      msg
// Display warn/error message in HTML element.
// commonCheck routine must have previously been called
// -----
var nbsp = 160;           // non-breaking space char
var emptyString = /^\s*$/ ;

function msg(fld, // id of element to display message in
            msgtype, // class to give element ("warn" or "error")
            message) // string to display
{
    // setting an empty string can give problems if later set to a
    // non-empty string, so ensure a space present. (For Mozilla and Opera one could
    // simply use a space, but IE demands something more, like a non-breaking space.)
    var dispmessage;
    if (emptyString.test(message))
        dispmessage = String.fromCharCode(nbsp);
    else
        dispmessage = message;

    var elem = document.getElementById(fld);
    elem.firstChild.nodeValue = dispmessage;

    elem.className = msgtype; // set the CSS class to adjust appearance of message
}
```

- Επίσης χρησιμοποιούνται οι παρακάτω δύο συναρτήσεις. Σημειώστε ότι η συνάρτηση **focus()** τοποθετεί τον cursor στο πεδίο που έχει δοθεί ως παράμετρος στο **global_valfield()** προκειμένου να γράψουμε σε αυτό.

```
// -----
//      setfocus
// Delayed focus setting to get around IE bug
// -----
var global_valfield; // retain valfield for timer thread
var node_text = 3; // DOM text node-type

function setFocusDelayed()
{
    global_valfield.focus();
}

function setfocus(valfield)
{
    // save valfield in global variable so value retained when routine exits
    global_valfield = valfield;
    setTimeout( 'setFocusDelayed()', 100 );
}
```

- Προκειμένου να ενεργοποιήσουμε τη συνάρτηση **validateEmail()** θα πρέπει να χρησιμοποιήσουμε συμβάντα. Θα χρησιμοποιήσουμε το συμβάν **onchange** το οποίο θα το τοποθετήσουμε μέσα στο **<input...>** του αντίστοιχου πεδίου. Αυτό που θα γράψουμε θα έχει ως συνέπεια να ελέγχεται κάθε φορά που γράφουμε κάτι στο πεδίο e-mail και να μας ενημερώνει για το αν είναι σωστό ή όχι.
- Τοποθετήστε λοιπόν μέσα στο **<input....>** του e-mail τα εξής:
onchange="validateEmail(this, 'inf_email', true);"
- Αυτό σημαίνει ότι μόλις γίνει η αλλαγή του πεδίου, θα κληθεί η συνάρτηση **validateEmail()** με πρώτη παράμετρο την τιμή που έχει το πεδίο, δεύτερη παράμετρο το **inf_email** και τρίτη παράμετρο το **true**.
- Ανοίξτε το Firefox ή το IE και πηγαίνετε στη φόρμα εγγραφής χρήστη. Συμπληρώστε ένα άκυρο e-mail (π.χ. **sadf**) και ένα έγκυρο (**user@email.vlsi.gr**) και θα πρέπει να δείτε τα αντίστοιχα μηνύματα λάθους και OK, όπως δείχνουν οι παρακάτω εικόνες(Εικόνα 2, Εικόνα 3):



Εικόνα 2 - Αποτέλεσμα εισαγωγής άκυρου e-mail σε φόρμα εγγραφής



Εικόνα 3 - Αποτέλεσμα εισαγωγής έγκυρου e-mail σε φόρμα εγγραφής

Η συνάρτηση `validateYear()`

- Με παρόμοιο τρόπο θα κατασκευάσουμε τη συνάρτηση που ελέγχει ότι ο χρήστης έχει εισάγει 4 έγκυρα ψηφία (αριθμούς).
- Κατασκευάστε τη συνάρτηση `validateYear()` με παρόμοιο τρόπο με το `validateEmail`. Θα πρέπει να τροποποιήσετε την κανονική έκφραση ώστε να δεχόμαστε 4 αριθμητικά ψηφία από το 0 έως 9 (`/^[0-9][0-9][0-9][0-9]$/`)
- Μπορείτε να βελτιώσετε την παραπάνω κανονική έκφραση με το να θεωρήσετε ότι τα δύο πρώτα ψηφία είναι είτε 19 είτε 20
- Μη ξεχάσετε να τροποποιήσετε τη φόρμα σας, ώστε να ενεργοποιεί τον έλεγχο μόλις τροποποιηθεί το αντίστοιχο πεδίο, με το να προσθέσετε στο `<input...>` του έτους γέννησης, τα εξής:

```
onchange="validateYear(this, 'inf_birthyear', true);"
```

- Επιβεβαιώστε την ορθή λειτουργία με το να δώσετε διάφορες ημερομηνίες και να δείτε αν εκτυπώνεται το πρόβλημα, όπως παρακάτω (Εικόνα 4, Εικόνα 5):



A screenshot of a form with a label "Έτος γέννησης:" and an input field containing "1991". To the right of the input field is an "OK" button. The background is dark blue.

Εικόνα 4 - Αποτέλεσμα εισαγωγής έγκυρης ημερομηνίας σε φόρμα εγγραφής



A screenshot of a form with a label "Έτος γέννησης:" and an input field containing "199a". To the right of the input field is an error message: "ERROR: not a valid year". The background is dark blue.

Εικόνα 5 - Αποτέλεσμα εισαγωγής άκυρης ημερομηνίας σε φόρμα εγγραφής

Η συνάρτηση `validateUsername()`

- Με παρόμοιο τρόπο όπως παραπάνω κατασκευάστε τη συνάρτηση που επιβεβαιώνει το όνομα χρήστη. Το όνομα χρήστη θα αποτελείται από μικρούς χαρακτήρες a-z αριθμούς 0-9.
- Κατασκευάστε την κανονική έκφραση η οποία θα επιβεβαιώνει ότι το όνομα χρήστη ακολουθεί την ανωτέρω μορφή, διαφορετικά θα εμφανίζει ένα κατάλληλο μήνυμα.
- Τροποποιήστε κατάλληλα τη φόρμα ώστε να ενεργοποιηθεί η συνάρτηση `validateUsername()` μόλις αλλάξει το αντίστοιχο πεδίο της φόρμας.
- Επιβεβαιώστε την ορθή λειτουργία της συνάρτησης (Εικόνα 6, Εικόνα 7):



A screenshot of a form with a label "Username:" and an input field containing "da-". To the right of the input field is an error message: "ERROR: Επιτρέπονται μόνο χαρακτήρες a-z 0-9". The background is dark blue.

Εικόνα 6 - Αποτέλεσμα εισαγωγής άκυρου username σε φόρμα εγγραφής



A screenshot of a form with a label "Username:" and an input field containing "admin". To the right of the input field is an "OK" button. The background is dark blue.

Εικόνα 7 - Αποτέλεσμα εισαγωγής έγκυρου username σε φόρμα εγγραφής

Η συνάρτηση `complex_password()`

- Επιθυμητό είναι όταν ο χρήστης πληκτρολογεί κάποιον κωδικό να του εμφανίζεται ένα μήνυμα για το πόσο καλός είναι ο κωδικός του. Για να πετύχουμε αυτό μπορούμε να χρησιμοποιήσουμε κάποια δικιά μας συνάρτηση javascript. Στο Internet μπορούμε να βρούμε πλήθος τέτοιων συναρτήσεων. Σας δίνεται η παρακάτω συνάρτηση.
- Τοποθετήστε τη συνάρτηση μέσα στο αρχείο των συναρτήσεων σας και τροποποιήστε κατάλληλα τη φόρμα, ώστε μόλις πληκτρολογεί ο χρήστης έναν κωδικό να του εμφανίζεται ένα πληροφοριακό μήνυμα.
- Η συνάρτηση (προσέξτε που έχει αλλαγή γραμμής):

```
function complex_password (
    valfield, // element to be validated
    infofield, // id of element to receive info/error msg
    required // true if required
)
{
var stat = commonCheck (valfield, infofield, required);

if (stat != proceed) return stat;

var tfld = trim(valfield.value); // value of field with whitespace trimmed off

var nuclearNailsPassword = /^(?=.*[_$!a-zA-Z0-9]*?[$!])(?=.*[_$!a-zA-Z0-9]*?[a-z])(?=.*[_$!a-zA-Z0-9]*?[A-Z])(?=.*[_$!a-zA-Z0-9]*?[0-9])$/;

var hardPassword = /^(?=.*[_$!a-zA-Z0-9]*?[$!])(?=.*[_$!a-zA-Z0-9]*?[A-Z])(?=.*[_$!a-zA-Z0-9]*?[a-z])(?=.*[_$!a-zA-Z0-9]*?[0-9])$/;

var goodPassword = /^(?=.*[_$!a-zA-Z0-9]*?[A-Z])(?=.*[_$!a-zA-Z0-9]*?[a-z])(?=.*[_$!a-zA-Z0-9]*?[0-9])(?=.*[_$!a-zA-Z0-9]*?[$!])(?=.*[_$!a-zA-Z0-9]*?[a-z])(?=.*[_$!a-zA-Z0-9]*?[A-Z])(?=.*[_$!a-zA-Z0-9]*?[0-9])(?=.*[_$!a-zA-Z0-9]*?[$!])(?=.*[_$!a-zA-Z0-9]*?[a-z])(?=.*[_$!a-zA-Z0-9]*?[A-Z])$/;

var weakPassword = /^(((?=.*[_$!a-zA-Z0-9]*?[A-Z])(?=.*[_$!a-zA-Z0-9]*?[a-z]))|(?=.*[_$!a-zA-Z0-9]*?[a-z])(?=.*[_$!a-zA-Z0-9]*?[0-9]))|(?=.*[_$!a-zA-Z0-9]*?[A-Z])(?=.*[_$!a-zA-Z0-9]*?[0-9])$/;

var superWeakPassword = /^(?=.*[_$!a-zA-Z0-9]*?[A-Z])(?=.*[_$!a-zA-Z0-9]*?[a-z])(?=.*[_$!a-zA-Z0-9]*?[0-9])$/;

if(nuclearNailsPassword.test(tfld) == true)
{
msg (infofield, "warn", "Ισχυρός Κωδικός");

}
else if(hardPassword.test(tfld) == true)
{
msg (infofield, "warn", "Καλός Κωδικός");

}
else if(goodPassword.test(tfld) == true)
```

```

{
msg (infofield, "warn", "Μέτριος Κωδικός");
}
else if(weakPassword.test(tfld) == true)
{
msg (infofield, "warn", "Ασθενής Κωδικός");
}
else if(superWeakPassword.test(tfld) == true)
{
msg (infofield, "warn", "Ανασφαλής Κωδικός");
}
else
{
msg (infofield, "warn", "Τελείως ανασφαλής κωδικός");
}
}
}

```

- Επιβεβαιώστε την ορθή λειτουργία όπως παρακάτω (Εικόνα 8 , Εικόνα 9):



Εικόνα 8 - Αποτέλεσμα εισαγωγής ασφαλούς κωδικού πρόσβασης



Εικόνα 9 - Αποτέλεσμα εισαγωγής μη ασφαλούς κωδικού πρόσβασης

Η συνάρτηση `validateNotEmpty()`

- Με παρόμοιο τρόπο κατασκευάστε τη συνάρτηση `validateNotEmpty()` η οποία θα ελέγχει ότι έχουμε πληκτρολογήσει κάτι (δεν είναι κενό) στο πεδίο που θα δίνεται ως πρώτο όρισμα, και θα τυπώνει OK ή “Απαιτείται να συμπληρωθεί”.
- Η συνάρτηση θα έχει 3 ορίσματα, όπως προηγουμένως.
- Η κλήση της συνάρτησης στο αντίστοιχο πεδίο `<input...>` θα γίνει αντί για το `onchange` με το συμβάν `onblur` , το οποίο χρησιμοποιείται όταν ο χρήστης απομακρυνθεί από αυτό το πεδίο. Διαφορετικά, αν χρησιμοποιούσαμε το `onchange` και ο χρήστης πατούσε κλικ σε αυτό το πεδίο, δε συμπλήρωνε τίποτα και απομακρύνονταν σε άλλο πεδίο, δε θα εκτελούνταν η συνάρτηση. Με το `onblur` γίνεται η εκτέλεση της αντίστοιχης συνάρτησης, είτε αλλαχθεί είτε όχι το πεδίο.
- Η κανονική έκφραση που αντιστοιχεί στο αν κάτι είναι κενό είναι η `var emptyfield = /^$/ ;`
- Αμέσως μετά θα συγκρίνετε αν ταιριάζει η είσοδος που έχει δοθεί με την παραπάνω έκφραση. Αν ταιριάζει, τότε θα εμφανίζεται το μήνυμα προβλήματος:

```

if (emptyfield.test(tfld)) {

```

- Αφού κατασκευάσετε τη συνάρτηση **validateNotEmpty()** ενεργοποιήστε την με το να τοποθετήσετε στο αντίστοιχο πεδίο **<input...>** (του πλήρους ονόματος) της φόρμας το εξής:

```
onblur="validateNotEmpty(this, 'inf_fullname', true);"
```

- Επιβεβαιώστε την ορθή λειτουργία με το να κάνετε κλικ στο πεδίο και να απομακρυνθείτε χωρίς να γράψετε τίποτα. Θα πρέπει να εμφανιστεί το αντίστοιχο μήνυμα.

Η συνάρτηση validateOnSubmit()

- Εκτός από τους παραπάνω ελέγχους είναι καλό να κάνουμε και έναν έλεγχο στη φόρμα πριν γίνει η αποστολή. Αν υπάρχουν σφάλματα σε κάποια πεδία (π.χ. δεν έχουν συμπληρωθεί) τότε δε θα γίνεται η αποστολή.
- Για να το επιτύχουμε αυτό θα πρέπει στο σημείο που ορίζεται η φόρμα **<form ...>** να προσθέσουμε το

```
onsubmit="return validateOnSubmit()"
```

Το οποίο ορίζει ότι θα εκτελεστεί η συνάρτηση **validateOnSubmit()** μόλις ο χρήστης πατήσει το κουμπί της αποστολής (submit) και πριν σταλούν δεδομένα στο διακομιστή.

- Αν η συνάρτηση **validateOnSubmit()** επιστρέψει FAIL/FALSE τότε δε θα εκτελεστεί η αποστολή.
- Αν η συνάρτηση **validateOnSubmit()** επιστρέψει TRUE τότε θα εκτελεστεί η αποστολή.
- Αν ο επισκέπτης έχει απενεργοποιημένη τη javascript τότε θα αγνοηθεί η παραπάνω παράμετρος.
- Η συνάρτηση που θα χρησιμοποιήσετε είναι η παρακάτω:

```
function validateOnSubmit() {
    var elem;
    var errs=0;
    // execute all element validations in reverse order, so focus gets
    // set to the first one in error.

    if ( !validateYear(document.forms.theForm.birthyear, 'inf_birthyear', true)) errs += 1;
    if ( !validateEmail(document.forms.theForm.email, 'inf_email', true)) errs += 1;
    if (!validateUsername(document.forms.theForm.username, 'inf_username', true)) errs
+= 1;
    if (!validateNotEmpty(document.forms.theForm.fullname, 'inf_fullname', true)) errs
+= 1;
    if (!validateNotEmpty(document.forms.theForm.password, 'inf_password', true)) errs
+= 1;

    if (errs>1) alert("Υπάρχουν προβλήματα σε πεδία της φόρμας");
    if (errs==1) alert("Υπάρχει ένα πεδίο με πρόβλημα");
}
```

```
return (errs==0);  
};
```

Χρήση της τεχνολογίας AJAX

Μια χρήσιμη λειτουργία σε αρκετά site είναι κατά τη δημιουργία ενός νέου λογαριασμού μόλις πληκτρολογεί ο χρήστης ένα όνομά που υπάρχει στη βάση δεδομένων, τότε αμέσως να του εκτυπώνεται ένα μήνυμα σε κατάλληλο κελί. Θα υλοποιήσουμε αυτή τη λειτουργία στο site μας. Για να το πετύχουμε αυτό θα κάνουμε τα παρακάτω βήματα.

Βήμα1: Έλεγχος στη βάση για ύπαρξη του συγκεκριμένου ονόματος χρήστη

- Δημιουργήστε το αρχείο **checkexist.php**
- Θα δέχεται μια παράμετρο με GET μέθοδο με όνομα παραμέτρου q, δηλαδή θα καλείται ως **checkexist.php?q=όνομα-χρηστη**
- Να ελέγχεται η τιμή της παραμέτρου q ότι είναι μικροί λατινικοί χαρακτήρες ή κεφαλαίοι ή αριθμοί με κανονική έκφραση. Αυτό το κάνουμε για να μη μας κάνει κάποιος επίθεση με SQL injection.
- Αν υπάρχει το όνομα χρήστη στη βάση θα εκτυπώνει το μήνυμα “Δεν είναι διαθέσιμο το όνομα” (δε θα εκτυπώνει ούτε **DOCTYPE** ούτε **html** ούτε τίποτα άλλο)
- Αν δεν υπάρχει το όνομα του χρήστη δε θα εκτυπώνει τίποτα.
- Σημειώστε, ότι επειδή η έξοδος αυτού του αρχείου θα τοποθετηθεί μέσα σε μια σελίδα, αυτό το script δεν πρέπει να τυπώνει τίποτα άλλο εκτός από το string. Δηλαδή, μη βάλετε κανένα TAG.
- Το script αυτό θα κάνει σύνδεση στη βάση.
- Θα εκτελεί μια εντολή select για να δει αν υπάρχει στον κατάλληλο πίνακα το συγκεκριμένο όνομα.
- Αν το αποτέλεσμα της **mysql_query()** είναι 0 τότε δεν υπάρχει τέτοιο όνομα χρήστη και σταματάει η εκτέλεση.
- Αν το αποτέλεσμα της **mysql_query()** είναι διαφορετικό από 0 τότε υπάρχει τέτοιος χρήστης και εκτυπώνεται το μήνυμα.
- Τέλος, θα κλείνει τη σύνδεση με τη βάση.

Βήμα 2: Επιβεβαιώστε την ορθή λειτουργία ως εξής:

- Δώστε ένα όνομα χρήστη που υπάρχει στη βάση και ένα όνομα χρήστη που δεν υπάρχει στη βάση με

http://.../checkexist.php?q=onomaxrhsth
και δείτε αν εμφανίζεται ή όχι το μήνυμα.

Βήμα 3: Τροποποίηση της φόρμας μας:

- Στο input πεδίο username στο γεγονός onchange θα καλείται πρώτα η συνάρτηση check_username() και στη συνέχεια η συνάρτηση που είχαμε δώσει.
- Δίπλα στο inf_username, τοποθετήσετε ένα ακόμη πεδίο td το οποίο να έχει id="exist_username".
- Δημιουργήστε στο HEAD τον κατάλληλο κώδικα για τη συνάρτηση function check_username(str) ώστε μόλις καλείται με παράμετρο το όνομα χρήστη να γίνεται μια ασύγχρονη κλήση προς το checkexist.php και να εκτυπώνει το αποτέλεσμα στο td πεδίο exist_username.

Προχωρημένα θέματα σε AJAX

Βελτιώνοντας τη σελίδα αξιολόγησης

Χρησιμοποιώντας javascript μπορούμε να βελτιώσουμε την εμφάνιση της σελίδας αξιολόγησης, και συγκεκριμένα στο σημείο που επιλέγει ο χρήστης τη βαθμολογία. Υπάρχει πληθώρα scripts που εκτελούν αυτή τη λειτουργία. Μπορείτε να βρείτε αρκετά τέτοια script στη σελίδα <http://speckyboy.com/2010/04/21/15-javascript-and-plugins-and-tutorials-for-adding-ratings/>

Προκειμένου να τοποθετήσετε ένα τέτοιο script στη σελίδα σας συνήθως:

- Στο HEAD της σελίδας σας τοποθετείτε τη σύνδεση με το αρχείο js ως

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

- τοποθετείτε τις εικόνες του rating (π.χ. αριθμός αστεριών) που απαιτούνται στον φάκελο που ορίζεται στο script.
- Δημιουργείτε έναν κατάλληλο πίνακα στη βάση δεδομένων που χρησιμοποιείτε (δεν απαιτείται από όλες τις βιβλιοθήκες rating)
- Τοποθετείτε την κλήση σε συγκεκριμένη συνάρτηση στη βιβλιοθήκη που έχετε κατεβάσει, σύμφωνα με τις οδηγίες.

Για παράδειγμα αν επιλέξουμε να χρησιμοποιήσουμε τη βιβλιοθήκη starry από <http://www.duarte.com/starry/> τότε θα πρέπει να κάνουμε τα εξής βήματα:

- Μεταφορτώνετε το zip αρχείο και εξάγετε τα αρχεία στον κατάλογο του site
- Τοποθετείτε στο <head> της σελίδας με τη φόρμα τις παρακάτω γραμμές:

```
<script type="text/javascript" src="prototype.lite.js"></script>
```

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

```
<link rel="stylesheet" href="stars.css" type="text/css" />
```

- Τοποθετήστε στη φόρμα τον παρακάτω κώδικα

```
<script>
```

```
new Starry('id_of_container', {name:"name_of_hidden_form_field"})
```

```
</script>
```

όπου id_of_container είναι ένα id για το οποίο θέλουμε να δημιουργήσουμε ένα rating. Για κάθε νέο rating απαιτείται και διαφορετικό id. Το name_of_hidden_form_field, είναι ένα κρυφό πεδίο στη φόρμα που θα τοποθετήσουμε αμέσως μετά

- Στο σημείο που θέλουμε να εμφανίζονται τα αστέρια, δίνουμε

```
<input type="hidden" name="name_of_hidden_form_field" value="0" />
```

- Με αυτόν τον τρόπο, μαζί με τα στοιχεία της φόρμας αποστέλλεται και το rating στη μεταβλητή name_of_hidden_form_field.

- Στο php που γίνεται η επεξεργασία της φόρμας, πρέπει να προστεθεί κατάλληλος κώδικας ώστε να χρησιμοποιεί και τη μεταβλητή

`$_POST['name_of_hidden_form_field']` και να την τοποθετηθεί στη βάση δεδομένων.

ΑΣΚΗΣΗ

Να τοποθετήσετε στη σελίδα `add.php` κατάλληλο κώδικα ώστε να δίνεται η δυνατότητα να επιλέγεται το `rating` όχι όπως στα προηγούμενα εργαστήρια, αλλά με χρήση κώδικα `javascript`. Το `rating` να αποθηκεύεται στη βάση.

Όταν ο χρήστης κάνει `search` να εμφανίζεται δίπλα στο όνομα του στοιχείου και ο μέσος όρος των βαθμολογιών για τη συγκεκριμένη καταχώρηση.