



# Validation στην HTML

**Αθανάσιος Ανδρούτσος**



# Client-Side Validation (1)

- Το κουμπί (button) Submit, αν πατηθεί, αποστέλλει τα δεδομένα μίας φόρμας στο back-end (σε ολοκληρωμένες εφαρμογές). Θα μπορούσαμε, πριν αποσταλούν, να ελέγχουμε τα δεδομένα που δίνουν οι χρήστες και να **εμφανίζονται μηνύματα λάθους** αν κάτι δεν έχει δοθεί με έγκυρο τρόπο (υποχρεωτικά πεδία, μορφή e-mail, αριθμοί έξω από όρια, κείμενο μικρότερο ή μεγαλύτερο από το επιτρεπτό, κλπ.)
- Η διαδικασία αυτή λαμβάνει χώρα στον Browser και καλείται Client-side validation



# Client-Side Validation (2)

- Τον έλεγχο στον client (front-end) τον κάνουμε για λόγους **αύξησης του user experience** μιας και δεν επαφιάμαστε πραγματικά στο έλεγχο των δεδομένων στο client-side το οποίο ούτως ή άλλως δεν μπορούμε να ελέγξουμε πραγματικά (μπορεί ο χρήστης ή μία εφαρμογή να παρακάμψει τον έλεγχο)
- Τον πραγματικό έλεγχο τον κάνουμε στο back-end



# Client-Side Validation (3)

Προγραμματισμός στο Web

- Client-side validation, εν αντιθέσει με την server-side validation
- Για παράδειγμα ο browser ελέγχει ένα e-mail ότι είναι έγκυρο αλλά ο server μπορεί να το απορρίψει γιατί υπάρχει ήδη στη Βάση Δεδομένων



# Client-Side Validation (4)

- Αυτό που θα μπορούσαμε να ελέγχουμε στο client-side είναι: 1) το αν το πεδίο θεωρούμε ότι είναι υποχρεωτικό (required) να δοθεί από τον χρήστη και 2) το format του πεδίου αν είναι ορθό
- Το format περιλαμβάνει το μέγεθος (minlength/maxlength σε χαρακτήρες) για τα αλφαριθμητικά ή το range (min/max) για αριθμητικά δεδομένα ή ημερομηνίες ή τύπο range
- Επίσης περιλαμβάνει τη μορφή του τύπου email, αν δηλαδή περιέχει το σύμβολο @



# Validation HTML attributes

Προγραμματισμός στο Web

- **required:** Ορίζει αν το input πεδίο είναι υποχρεωτικό, δηλαδή πρέπει ο χρήστης να δώσει τιμή, πριν αποσταλεί η φόρμα
- **minlength/maxlength:** Ορίζει το ελάχιστο/μέγιστο πλήθος χαρακτήρων σε αλφαριθμητικά δεδομένα
- **min/max:** Ορίζει την ελάχιστη/μέγιστη τιμή σε αριθμητικά δεδομένα, ημερομηνίες και range
- **type:** Ορίζει τον τύπο του <input> , π.χ. number, email
- **pattern:** Ορίζει μία κανονική έκφραση (regular expression), που ορίζει την ακριβή μορφή των δεδομένων εισόδου



# Required / minlength / maxlength

Προγραμματισμός στο Web

```
<form>
  <div>
    <label for="firstname">Όνομα (ΕΛΛΗΝΙΚΑ)</label>
    <input type="text" name="firstname" id="firstname">
  </div>
  <div>
    <label for="lastname">Επώνυμο (ΕΛΛΗΝΙΚΟΙ ή ΛΑΤΙΝΙΚΟΙ ΧΑΡΑΚΤΗΡΕΣ )</label>
    <input type="text" name="lastname" id="firstnane">
  </div>
  <div>
    <label for="email">E-mail</label>
    <input type="email" name="email" id="email" required>
  </div>
  <div>
    <label for="password">Password</label>
    <input type="password" name="password" id="password" minlength="8" required>
  </div>
  <div>
    <label for="submission-date"></label>
    <input type="date" name="submission-date" id="submission-date" placeholder="dd-MM-yyyy" maxlength="10">
  </div>
  <div>
    <label for="submission-date-time"></label>
    <input type="datetime-local" name="submission-date-time" id="submission-date-time" placeholder="dd-MM-yyyy HH:mm" maxlength="10">
  </div>
  <div>
    <input type="submit" name="" id="">
  </div>
</form>
```

- Τα πεδία required είναι υποχρεωτικά, ενώ ορίζουμε πλήθος χαρακτήρων με minlength, maxlength



# Pattern (1)

- Τα παρακάτω παραδείγματα **παρατίθενται μόνο για λόγους πληρότητας της παρουσίασης και μπορούν να παραλειφθούν**
- Η ιδιότητα **pattern** της HTML χρησιμοποιείται για *pattern matching*, που υλοποιείται με Regular Expressions από την JavaScript
- Το pattern ορίζει (και αναγνωρίζει) τη μορφή του αλφαριθμητικού που θέλουμε ως είσοδο





## Pattern (2)

- Για παράδειγμα το `\d` σημαίνει ότι περιμένουμε ένα ψηφίο (digit). Το `\` λέγεται *escape* και χρησιμοποιείται για να αλλάξει την βασική ερμηνεία του `d`, που είναι το γράμμα `d`
- Το `\d` είναι ισοδύναμο με το `[0-9]` όπου τα *brackets* `[]` χρησιμοποιούνται για να ορίσουν μία τιμή από ένα *range* τιμών που περιλαμβάνουν



## Pattern (3)

- Αν θέλουμε να αναγνωρίσουμε ένα οποιοδήποτε ψηφίο, μπορούμε να δώσουμε ένα `pattern="\d"` ή `pattern="[0-9]"`
- Αν θέλουμε να αναγνωρίσουμε ένα οποιοδήποτε λατινικού χαρακτήρα, δίνουμε `pattern="[a-zA-Z]"`. Αν θέλουμε να αναγνωρίζουμε και το κενό: `[ a-zA-Z]` (υπάρχει ένα κενό πριν το a)



# Pattern (4)

- Αν θέλουμε να αναγνωρίσουμε τη μορφή μίας ημερομηνίας dd-MM-yyyy θα μπορούσαμε να έχουμε  $\backslash d\{2\}-\backslash d\{2\}-\backslash d\{4\}$
- Ή αν θέλαμε να περιορίσουμε και το εύρος των τιμών:  $(0?[1-9]| [12]\backslash d| 3[01])-(0?[1-9]| 1[0-2])-\backslash d\{4\}$
- Το | διαβάζεται ή (διαζευκτικό)
- Το ? είναι ποσοδείκτης και σημαίνει 0 ή 1 φορά το 0 που είναι ακριβώς πριν το ?. Οπότε η ημέρα (ή ο μήνας) μπορεί να δοθεί είτε ως 05 ή σκέτο 5 για παράδειγμα
- Επίσης, ποσοδείκτης είναι και το {4} που σημαίνει ένα ψηφίο 4 φορές και αναφέρεται στο έτος (year)



# Παράδειγμα validation (1)

Προγραμματισμός στο Web

```
<> validation.html X
chapter1 > <> validation.html > html > body > form
9   <body>
10   <form>
11     <div>
12       <label for="firstname">Όνομα (ΕΛΛΗΝΙΚΑ)</label>
13       <input type="text" name="firstname" id="firstname" pattern="[Α-Ωα-ωίϊϊόάέύϋϋήώ]{1,50}">
14     </div>
15     <div>
16       <label for="lastname">Επώνυμο (ΕΛΛΗΝΙΚΟΙ ή ΛΑΤΙΝΙΚΟΙ ΧΑΡΑΚΤΗΡΕΣ)</label>
17       <input type="text" name="lastname" id="firstname" pattern="[Α-Ωα-ωίϊϊόάέύϋϋήώ]{1,50}">
18     </div>
19     <div>
20       <label for="email">E-mail</label>
21       <input type="email" name="email" id="email" required>
22     </div>
23     <div>
24       <label for="password">Password</label>
25       <input type="password" name="password" id="password" minlength="8" required>
26     </div>
27     <div>
28       <label for="submission-date"></label>
29       <input type="date" name="submission-date" id="submission-date" placeholder="dd-mm-yyyy" maxlength="10"
30         pattern="^(0?[1-9]|[12]\d|3[01])-(0?[1-9]|1[0-2])-\d{4}$">
31     </div>
32     <div>
33       <input type="submit" name="" id="">
34     </div>
35   </form>
36 </body>
37 </html>
```



# Παράδειγμα validation (2)

Προγραμματισμός στο Web

```
<form>
  <div>
    <label for="firstname">Όνομα (ΕΛΛΗΝΙΚΑ)</label>
    <input type="text" name="firstname" id="firstname" pattern="[ Α-Ωα-ωίϊϊόάέύϋϋήώ]{1,50}">
  </div>
  <div>
    <label for="lastname">Επώνυμο (ΕΛΛΗΝΙΚΟΙ ή ΛΑΤΙΝΙΚΟΙ ΧΑΡΑΚΤΗΡΕΣ )</label>
    <input type="text" name="lastname" id="firstnane" pattern="[ A-Za-zΑ-Ωα-ωίϊϊόάέύϋϋήώ]{1,50}">
  </div>
  <div>
```

- Στο Όνομα αναγνωρίζουμε μόνο το κενό, ελληνικά κεφαλαία (Α-Ω) ή πεζά (α-ω) ή και πεζά ελληνικά με τόνους
- Στο Επώνυμο αναγνωρίζουμε τα ίδια συν Λατινικούς χαρακτήρες (A-Za-z) κεφαλαίους ή πεζά
- Μιας και τα [] αναφέρονται σε ένα χαρακτήρα, έχουμε στο τέλος ποσοδείκτες {1, 50} που σημαίνει 1 έως 50 χαρακτήρες της μορφής που περιγράφουμε



# Παράδειγμα validation (2)

```
<div>
  <label for="submission-date"></label>
  <input type="date" name="submission-date" id="submission-date" placeholder="dd-MM-yyyy" maxlength="10"
    pattern="^(0?[1-9]|[12]\d|3[01])-(0?[1-9]|1[0-2])-\d{4}$">
</div>
```

- **pattern** – Στην ημερομηνία μπορούν να δοθούν συγκεκριμένοι συνδυασμοί χαρακτήρων. Αυτό δεν ισχύει για τον τύπο date, αλλά για το text αν γίνει fallback το date σε text type σε παλαιότερους browsers. Στο παράδειγμα της ημερομηνίας θέλουμε αριθμούς ημέρας που να ξεκινούν με 0 (ή χωρίς 0, το ? μετά το 0 σημαίνει 0 ή 1 φορές αυτό που προηγείται) και να ακολουθούνται από 1-9 (μέσα σε [ ] δίνουμε μεμονωμένες τιμές ή range τιμών) ή να ξεκινούν με 1 ή 2 και να ακολουθούνται από 0-9 (το \d σημαίνει digit –escape το d- και είναι ισοδύναμο με [0-9]) ή να ξεκινούν με 3 και μετά να ακολουθούνται από 0 ή 1, μετά να ακολουθεί - και μετά θέλουμε πάλι ένα αριθμό, τον μήνα που ξεκινάει από 0 (ή χωρίς 0, λόγω του ? που όπως αναφέραμε σημαίνει 0 ή 1 φορές αυτό που προηγείται) και μετά ψηφίο 1-9 ή να αρχίζει από 1 και μετά ακολουθεί ψηφίο 0-2, και μετά ακολουθεί - και μετά πάλι έναν τρίτο αριθμό μεταξύ 0000-9999, το έτος, που είναι ψηφίο (\d) τέσσερις φορές. Το {4} είναι ποσοδείκτης και σημαίνει τέσσερις φορές, οτιδήποτε σύμβολο είναι αριστερά του. Το ^ (caret) συμβολίζει την αρχή του αλφαριθμητικού και το \$ το τέλος, δηλαδή δεν πρέπει να υπάρχει κάτι άλλο πριν ή μετά το αλφαριθμητικό εισόδου. Οι παρενθέσεις ομαδοποιούν τους επιμέρους δύο πρώτους αριθμούς της ημερομηνίας, ώστε να εφαρμόζεται σωστά το delimiter / σε όλη την έκφραση του αριθμού και όχι μόνο στο τελευταίο της μέρος



# Ημερομηνίες (1)

- Η εισαγωγή ημερομηνιών δημιουργεί προβλήματα πολλές φορές λόγω του διαφορετικού τρόπου αναπαράστασης μεταξύ διαφόρων ζωνών ώρας
- Η HTML παρέχει τον τύπο *date* (`type="date"`) που είναι ημερομηνία της μορφής `yyyy-MM-dd` (ISO8601, αναφερόμαστε στον τρόπο αναπαράστασης, ανεξάρτητα από τον τρόπο παρουσίασης στον browser. Μπορεί ο browser να έχει `Locale el` και να χρησιμοποιεί για εισαγωγή `dd-mm-yyyy` αλλά ο browser θα αποθηκεύσει ως `yyyy-mm-dd`)
- Παρέχεται και ο τύπος *datetime-local* (`type="datetime-local"`) που έχει τη μορφή `yyyy-MM-dd HH-mm` ή σε normalized form `yyyy-MM-ddTHH-mm` και δεν ορίζει ζώνη ώρας ως προς το UTC (παλαιότερα Greenwich)



# Ημερομηνίες (2)

- Για λόγους αύξησης της εμπειρίας του χρήστη θα ήταν καλό να χρησιμοποιούμε αυτούς τους δύο τύπους που παρέχουν και ένα calendar (date picker) για εισαγωγή και όχι απλό text
- Ωστόσο αυτοί οι δύο τύποι (ο date picker) δεν παρέχουν πραγματικό validation, αλλά απλά ένα περιορισμό στη μορφή των ημερομηνιών που δίνουμε. Μπορεί ωστόσο η μορφή να είναι σωστή, αλλά η ημερομηνία μη-έγκυρη. Επίσης, ο date picker που χρησιμοποιείται δεν χρησιμοποιεί το pattern
- Επιπλέον, μπορεί διάφοροι παλαιότεροι browsers να μην υλοποιούν αυτούς τους τύπους, οπότε τότε το πεδίο εισόδου γίνεται fallback σε text , οπότε τότε ισχύει το validation με το pattern





# Ημερομηνίες (3)

- Μία λύση σε αυτά τα προβλήματα θα ήταν να υλοποιούνται σε διαφορετικά text πεδία ξεχωριστά το έτος, ο μήνας, η ημέρα (η ώρα, τα λεπτά, κλπ.) και η ημερομηνία να συντίθεται από αυτά τα πεδία (και η ώρα αν το πεδίο είναι datetime-local)