

Πανεπιστήμιο Κρήτης

ΗΥ359 – Διαδικτυακός Προγραμματισμός (Web Programming)

Χειμερινό Εξάμηνο 2021/2022

Διδάσκων: **Μιχαήλ Μουνταντωνάκης**

Υπεύθυνος Βοηθός: **Γ. Κοκιάδης** (Συντονιστής) και οι υπόλοιποι βοηθοί

## **2η Σειρά Ασκήσεων (Ατομική)**

Διάρκεια: 30/10 – 15/11

**Αξία:** 10% του τελικού σας βαθμού

**(Συνολο ασκήσεων: 112 μονάδες)**

**Θεματική ενότητα:** Εξοικείωση με JavaScript – APIs - Ajax - JQuery

### **Άσκηση 1. Βελτίωση Φόρμας (20%)**

*Αποτελεί μέρος του συνόλου του project*

Εμπλουτισμός/βελτίωση της φόρμας εγγραφής που φτιάξατε στην άσκηση 1.

#### **a) (5%) Password Mismatch - DOM manipulation**

- Θα πρέπει να ελεγχθεί ότι τα 2 passwords που έχουν εισαχθεί από τον χρήστη είναι όντως τα ίδια.
- Σε περίπτωση λάθους θα πρέπει να τυπωθεί κατάλληλο μήνυμα δίπλα από τα πεδία (όχι με χρήση alert).
- Επίσης θα πρέπει ο χρήστης να μπορεί πατώντας ένα κουμπί να βλέπει το password του (και πατώντας το ξανά να το κρύβει).

#### **b) (5%) Ασφάλεια Password**

Θα πρέπει να ελεγχθεί με απλούς κανόνες ότι το password είναι ασφαλές. Συγκεκριμένα οι κανόνες πρέπει να είναι οι εξής:

- Αν το 50% του password και πάνω είναι αριθμοί να του εμφανίζει μήνυμα δίπλα από τα πεδία (όχι με χρήση alert) ότι είναι weak password.
  - πχ 12345ab\$ (5 αριθμοί στα 8 γράμματα)
- Αν ένας συγκεκριμένος χαρακτήρας καταλαμβάνει το 50% και πάνω του password πάλι να εμφανίζεται weak password
  - πχ 1apa4#aa (το α εδώ υπάρχει 4 στα 8 γράμματα)
- Αν το  $\geq 80\%$  των χαρακτήρων είναι διαφορετικοί (και δεν ισχύουν οι παραπάνω κανόνες) να του εμφανίζει strong password.
  - πχ. r4p0t\*wqw4 (8 μοναδικοί χαρακτήρες στους 10).
- Σε όλες τις άλλες περιπτώσεις να του εμφανίζει medium password

- πχ. 5art\*dtra3 (7 μοναδικοί χαρακτήρες στους 10, άρα δεν πιάνει τις παραπάνω περιπτώσεις).
- Μπορείτε να έχετε ένα κουμπί δίπλα στο password που να γράφει see password strength. Μόλις το πατάτε, μπορεί να γράφει σε ένα μήνυμα δίπλα στο password (όχι με χρήση alert) αν το password είναι weak, medium ή strong. Εναλλακτικά μπορεί να γίνει και με τη χρήση του event onchange.
- **Hint: Για να διαβάσετε όλους τους χαρακτήρες μιας λέξης:**

```
var text="abcdef";  
  
for(var i=0; i<text.length;i++){  
    console.log(text.charAt(i));  
}
```

**c) (5%) Έξτρα επιλογές για γιατρό:** Αν ο χρήστης είναι γιατρός,

- να εμφανίζεται μία επιλογή ειδικότητα, πχ ένα selection box (ή radio button) που να έχει επιλογές παθολόγος, γενικός ιατρός
- να εμφανίζεται ένα έξτρα textarea που να γράφει “Πληροφορίες για τον ιατρό”
- το label Διεύθυνση να αλλάζει όνομα σε Διεύθυνση Ιατρείου.

Αν ξαναεπιλεγεί ο απλός χρήστης, τότε να εξαφανίζεται η ειδικότητα και το label Διεύθυνση Ιατρείου να γίνεται πάλι Διεύθυνση.

**d) (4%) ΑΜΚΑ & Ημερομηνία γέννησης:** Θα πρέπει τα 6 πρώτα ψηφία του ΑΜΚΑ να συμφωνούν με την ημερομηνία γέννησης (χωρίς τα 2 πρώτα ψηφία του έτους), δηλαδή αν η ημερομηνία γέννησης είναι 25/05/1980, το ΑΜΚΑ να ναι της μορφής 250580XXXXX.

**e) (1%) Έλεγχος Checkbox:** Θα πρέπει να ελέγχεται αν το checkbox με τους όρους συμφωνίας έχει πατηθεί. Αν όχι να του εμφανίζει κατάλληλο μήνυμα.

## Άσκηση 2 Χρήση Χαρτών Open Street Maps [30%]

Αποτελεί μέρος του συνόλου του project

Εμπλουτισμός/βελτίωση της φόρμας εγγραφής που φτιάξατε στην άσκηση 1 με χάρτες.

**SOS Δείτε το σχετικό tutorial και τον κώδικα για το RAPID API και τα OSM maps.**

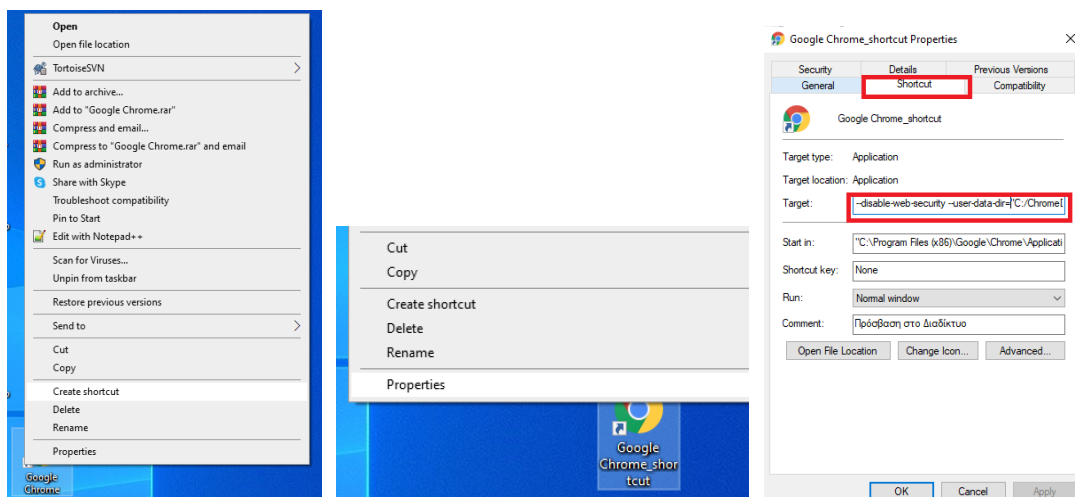
### ΠΡΟΣΟΧΗ!!!

Σε περίπτωση που ο browser παραπονιέται για προβλήματα με τον CORS (Cross-Origin Resource Sharing) μηχανισμό μπορείτε να κάνετε τα εξής:

- Να ανοίξετε την εφαρμογή του browser που χρησιμοποιείτε με απενεργοποιημένα το security μηχανισμό (π.χ. για τον chrome --disable-web-security)
- Θα πρέπει να βάλετε τη σελίδα σας σε κάποιο web server που θα τρέχει στο μηχανήμά σας (π.χ. Apache).
- Να εγκαταστήσετε κάποιο plugin το οποίο κάνει override τον CORS μηχανισμό (δεν συστήνεται)

### Οδηγίες για απενεργοποίηση CORS

- Δημιουργήστε ένα shortcut του chrome στην επιφάνεια εργασίας
- Δεξί κλικ στο shortcut και κλικ στα Properties.
- Πηγαίνετε στο tag Shortcut και επεξεργαστείτε το Target property.
- Βάλτε του την τιμή **--disable-web-security --user-data-dir="C:/ChromeDevSession"** μετά το τιμή που έχει πχ "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe"



- Ανοίξτε το chrome από αυτό το shortcut

#### a) (10%) Χρήση OpenStreetMaps (OSM) API - Επαλήθευση πόλης

Συνενώνοντας τις τιμές των πεδίων “Χώρα”, “Πόλη” και “Διεύθυνση” από τη φόρμα, θα πρέπει να επαληθεύσετε μέσω του **nominatim OSM geocode service** αν η συγκεκριμένη περιοχή υπάρχει ή όχι (πχ μέσω ενός κουμπιού).

<https://rapidapi.com/GeocodeSupport/api/forward-reverse-geocoding>

- Θα πρέπει να κάνετε το κατάλληλο request σε αυτό το service και να πάρετε την απάντησή του σε JSON μορφή (δείτε το παράδειγμα στο μάθημα). Θεωρήστε ότι η τοποθεσία υπάρχει εφόσον το API επιστρέφει μία μη κενή λίστα αποτελεσμάτων στην JSON απάντηση που επιστρέφει και αν δίνει πάνω από μία τοποθεσίες επιλέξτε την πρώτη σαν πιο σχετική.
- Σε περίπτωση που η απάντηση του service είναι κενή (άρα δηλαδή δεν γνωρίζει το OSM τη συγκεκριμένη τοποθεσία), θα πρέπει να τυπώνεται κατάλληλο μήνυμα λάθους στη σελίδα, σε κάποιο DOM element (όχι χρήση alert).
- Όλες οι απαντήσεις από τα παραπάνω services είναι σε JSON το οποίο μπορείτε να κάνετε parse ώστε να δημιουργήσετε ένα JavaScript αντικείμενο από την πλευρά της Javascript μέσω της εντολής `JSON.parse(json_str)`.
- Να κρατήσετε τα lat και lon, γιατί θα τα χρειαστείτε στο επόμενο βήμα.
- **Αν η περιοχή δε βρίσκεται στην Κρήτη, να εμφανίζεται μήνυμα ότι η υπηρεσία είναι διαθέσιμη μόνο στην Κρήτη αυτή τη στιγμή.**
  - **Hint:** Τσεκάρετε το `display_name` που επιστρέφεται από το json αν έχει μέσα την Κρήτη.

#### b) (10%) Χρήση OSM maps - Εμφάνιση χάρτη

Δεδομένης μιας σωστής τοποθεσίας (Χώρα, Πόλη και Διεύθυνση) θα πρέπει να δίνεται η επιλογή στο χρήστη μέσω κάποιου κουμπιού,

- να βλέπει σε χάρτη την τοποθεσία την οποία δήλωσε με ένα marker.
- Σε περίπτωση που ο χρήστης αλλάξει την πόλη, θα πρέπει να χαθούν από το DOM τα elements που σχετίζονται με τον container (π.χ. div) του χάρτη και να γίνει ξανά η επαλήθευση της τοποθεσίας για την εμφάνιση του κουμπιού.
- Σε περίπτωση που το geocode έχει επιστρέψει περισσότερες από μία σωστές τοποθεσίες, μπορείτε να χρησιμοποιήσετε την πρώτη. Κεντράρετε το χάρτη στη συγκεκριμένη θέση. Να λάβετε υπόψη ότι το OSM από default χρησιμοποιεί Spherical Mercator Coordinates αντί των κοινά χρησιμοποιούμενων WGS84 GPS συντεταγμένων. Οπότε θα πρέπει να γίνει μια μετατροπή από EPSG:4326 (WGS84) σε EPSG:900913 (Spherical Mercator) μέσω της transform που δίνεται από τη OpenLayers.LonLat (δείτε <http://dev.openlayers.org/docs/files/OpenLayers/BaseTypes/LonLat-js.html>).
- **Δείτε φυσικά και το έτοιμο παράδειγμα που είδαμε στο μάθημα.**

### c) (10%) **Αυτόματη Συμπλήρωση Πεδίων Διεύθυνσης, Πόλης και Χώρας με HTML5 geolocation και markers**

Στο συγκεκριμένο ερώτημα καλείστε να κάνετε αυτόματη συμπλήρωση των πεδίων “Χώρα”, “Πόλη” και “Διεύθυνση” βάσει της τωρινής θέσης του χρήστη και να δείξετε τη θέση στο χάρτη.

- Η συγκεκριμένη λειτουργικότητα βασίζεται στη χρήση του geoLocation που προσφέρεται από browsers που υποστηρίζουν HTML5 και επιστρέφει το στίγμα του χρήστη.
  - Δείτε το [https://www.w3schools.com/html/html5\\_geolocation.asp](https://www.w3schools.com/html/html5_geolocation.asp)
  - Η ακρίβεια που προσφέρει το συγκεκριμένο χαρακτηριστικό της HTML5 σχετίζεται με τη διαθέσιμη υπάρχουσα πληροφορία, πχ. public IP address, Cell tower ID triangulation ή GPS και μας επιστρέφει ένα στίγμα θέσης.
- Μέσω της reverse υπηρεσίας του **nominatim OSM geocode service** μπορείτε από αυτό το στίγμα να βρείτε τα στοιχεία της διεύθυνσης, πόλης και χώρας του χρήστη (αν υπάρχουν).
  - <https://rapidapi.com/GeocodeSupport/api/forward-reverse-geocoding>
- Αν δεν υπάρχουν απλά τυπώνεται κατάλληλο μήνυμα σε κάποιο dom element. Η αυτόματη συμπλήρωση θα παρέχεται μέσω ενός κουμπιού, το οποίο θα είναι ενεργό μόνο αν παρέχεται η συγκεκριμένη λειτουργικότητα από τον browser, διαφορετικά το κουμπί θα πρέπει να είναι ανενεργό.

### **Χρήση OSM maps - geocoding services**

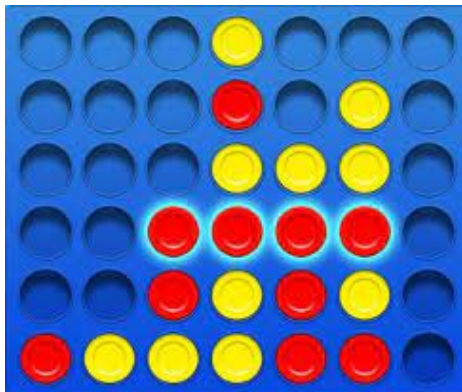
Καλείστε να χρησιμοποιήσετε το API των OSM maps όσο και το geocoding service. Δείτε κυρίως τα παραδείγματα στο elearn και τους παρακάτω βοηθητικούς συνδέσμους:

- [https://www.w3schools.com/html/html5\\_geolocation.asp](https://www.w3schools.com/html/html5_geolocation.asp)
- <https://wiki.openstreetmap.org/wiki/Nominatim>
- <https://rapidapi.com/GeocodeSupport/api/forward-reverse-geocoding>
- [https://wiki.openstreetmap.org/wiki/OpenLayers\\_Marker\\_Example](https://wiki.openstreetmap.org/wiki/OpenLayers_Marker_Example)
- [https://wiki.openstreetmap.org/wiki/Openlayers\\_POI\\_layer\\_example](https://wiki.openstreetmap.org/wiki/Openlayers_POI_layer_example)

**Σημείωση:** Σημαντικό είναι να καταλάβετε τι ακριβώς κάνει ο κώδικας και να αποφύγετε απλά το έτοιμο copy-paste (θυμηθείτε τα αντικείμενα που είδαμε στο μάθημα).

- Προσπαθήστε οι συναρτήσεις που θα φτιάξετε να είναι γενικές έτσι ώστε π.χ. αν στο μέλλον χρειαστεί να ζωγραφίσετε το map σε κάποιο άλλο div να μπορείτε να το κάνετε εύκολα δίνοντας το κατάλληλο όρισμα στην αντίστοιχη συνάρτηση.
- Αντίστοιχα αν χρειαστεί να εισάγετε νέους markers να μπορεί να γίνει μέσω της κατάλληλης κλήσης μιας συνάρτησης, η οποία θα μπορεί να δέχεται το map πάνω στο οποίο θέλουμε να ζωγραφίσουμε τους markers και το αντίστοιχο location.
- Για τα requests που θα χρειαστεί να κάνετε στα OSM services θα χρησιμοποιήσετε ajax requests (δείτε και τον κώδικα που έχει δοθεί στο μάθημα και το σχετικές διαφάνειες). Επίσης αφιερώστε λίγο χρόνο για τη σχεδίαση της σελίδας. Σκεφτείτε ότι ό,τι χτίζεται τώρα θα χρησιμοποιηθεί και στη συνέχεια της εργασίας σας στο μάθημα.

### Άσκηση 3. Υλοποίηση Παιχνιδιού ΣΚΟΡ 4 – [50%+12% bonus]



#### Φτιάξτε μέσω HTML και Javascript το παιχνίδι ΣΚΟΡ 4.

Για να εξοικειωθείτε με τη Javascript, θα φτιάξετε το παιχνίδι **ΣΚΟΡ 4**, χρησιμοποιώντας μόνο **HTML, CSS και Javascript**. Στο παιχνίδι αυτό, υπάρχει ένα ταμπλό 6x7 (6 γραμμές και 7 στήλες).

Κάθε παίκτης έχει στην διάθεσή του 21 πιόνια, τα οποία τοποθετούνται σε οποιαδήποτε στήλη από την κορυφή του ταμπλό και ολισθαίνουν προς τα κάτω. Κάθε φορά παίζει ένας παίκτης (είτε ο κίτρινος είτε ο κόκκινος), και σκοπός του παιχνιδιού είναι να φτιάξει κάποιος μία συνεχόμενη τετράδα από μάρκες του ίδιου χρώματος, είτε κάθετα, είτε διαγώνια, είτε οριζόντια.

Υποθέτουμε ότι οι 2 παίκτες μοιράζονται την ίδια οθόνη και το ίδιο ποντίκι.

#### α) Σχεδίαση Αρχικής HTML σελίδας (10%)

Μπορείτε να σχεδιάσετε όπως θέλετε τη σελίδα σας. Η αρχική σελίδα θα έχει ένα ταμπλό (πχ ένας πίνακας με κουμπιά), και θα πρέπει να έχετε ένα info box (μπορεί να είναι ένα άλλο div) που ο σκοπός του είναι να αναγράφει ποιος παίκτης παίζει, πόσες κινήσεις έχουν γίνει και πόσα κελιά παραμένουν κενά.

#### β) Συναρτήσεις Javascript (40%)

Θα πρέπει να έχετε τις κατάλληλες συναρτήσεις που να ελέγχουν

- **Αρχικοποίηση νέου παιχνιδιού (4 μονάδες) → `newGame()`**
  - ο Δίνει σειρά σε κάποιο παίκτη, αρχικοποιεί το infobox, μεταβλητές κλπ
- **Κίνηση Παίκτη (36 μονάδες)**
  - ο Θα έχετε έναν event handler που θα ενεργοποιείται μόλις πατηθεί κάποιο κουμπί, πχ η επιλογή του παίκτη (**2 μονάδες**) → `play()`
  - ο Συνάρτηση για το ποιος παίκτης παίζει (**1 μονάδα**) → `getPlayerTurn()`
  - ο Έλεγχος για το αν η κίνηση είναι επιτρεπτή (**2 μονάδες**) → `isValidMove()`
  - ο Έλεγχος αν ο παίκτης κέρδισε → `hasPlayerWon()`
    - Έλεγχος νίκης κάθετα (**6 μονάδες**) → `verticalWin()`
    - Έλεγχος νίκης οριζόντια (**6 μονάδες**) → `horizontalWin()`
    - Έλεγχος νίκης διαγώνια (**8 μονάδες**) → `diagonalWin()`
  - ο Ελέγχει αν τελείωσε το παιχνίδι με ισοπαλία (**2 μονάδες**) → `isDraw()`
    - Ελέγχει αν γέμισε όλο το ταμπλό

- ο Σωστή ανανέωση σελίδας HTML → `updatePage ()`
  - Να ανανεώνει το ταμπλό (**2 μονάδες**)
  - Αν η κίνηση είναι σωστή να γράφει ποιος παίκτης έπαιξε και πόση ώρα του πήρε για να κάνει την κίνηση
    - Ποιος παίκτης έπαιξε (**1 μονάδα**)
    - Πόση ώρα του πήρε (**2 μονάδες**)
      - ο Μετρήστε τη διαφορά μέσω του `date`.
- ο Αλλαγή σειράς (**1 μονάδα**) → `changePlayerTurn()`
- ο Όταν έχουμε νικητή θα πρέπει να επιστρέφεται μήνυμα στους χρήστες και να αναβοσβήνει η τετράδα που έδωσε τη νίκη (Δείτε το παράδειγμα του μαθήματος στην τρίλιζα) (**3 μονάδες**)

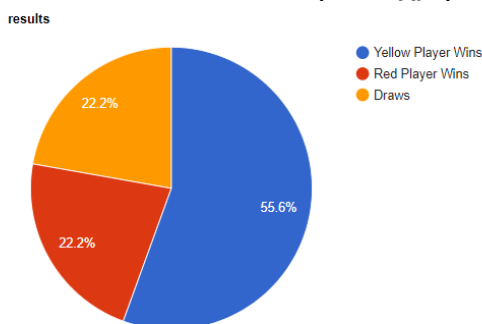
**Μπορείτε να βάλετε όσες παραμέτρους θέλετε, να κάνετε όποιους ελέγχους επιθυμείτε και να ορίσετε νέες συναρτήσεις. Αρκεί να μην χρησιμοποιείται αυτούσιο κώδικα από το internet.**

**Event Handling και ανανέωση HTML σελίδας:** Θα πρέπει να κάνετε σωστά το event handling, έτσι ώστε να γίνονται οι απαραίτητες ενέργειες και η ανανέωση της σελίδας. Μπορείτε να χρησιμοποιήσετε φυσικά και jquery.

#### γ) Bonus (12%)

##### Νέο παιχνίδι ενώ παίζουμε ήδη/ή μετά το τέλος (6 μονάδες)

- Να μπορεί να γίνει **νέο παιχνίδι ανα πάσα στιγμή** (**2 μονάδες**) χωρίς να γίνεται refresh, και φυσικά μετά το τέλος ενός παιχνιδιού
- Να **αποθηκεύονται και να φαίνονται οι νίκες κάθε παίκτη** στο infobox (**2 μονάδες**) → πχ κίτρινος 5 νίκες, κόκκινος 2 νίκες, 2 ισοπαλίες
- **Στατιστικά μέσω οπτικοποίησης** (**2 μονάδες**): Χρησιμοποιήστε Google Charts (πχ pie charts), για να δείχνεις τις νίκες κάθε παίκτη σε κάποιο div (πχ όταν το πατήσει ο χρήστης) παράδειγμα:



Hint: <https://developers-dot-devsite-v2-prod.appspot.com/chart/interactive/docs/gallery/piechart>

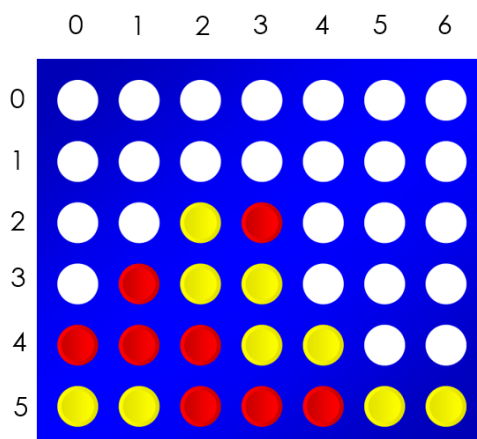
**Παιχνίδι με υπολογιστή (6 μονάδες):** Μπορείτε να υλοποιήσετε ένα παιχνίδι (σε διαφορετικό αρχείο) όπου να παίζει και ο υπολογιστής ενάντια στον παίκτη. Μπορείτε να ορίσετε κάποιους κανόνες έτσι ώστε να υπάρχει μία δυσκολία (να μην είναι τελείως random οι κινήσεις). Όσο πιο δύσκολο είναι το παιχνίδι με τον υπολογιστή, τόσο μεγαλύτερη θα είναι η βαθμολογία. Σε κάθε περίπτωση θα εκτιμηθεί η προσπάθεια.

- **Στρατηγικές**

<https://www.gamesver.com/10-helpful-tips-tricks-and-strategies-to-win-at-connect-4/>

### Hints

- **SOS. Δείτε και το παράδειγμα με την τρίλιζα που δείξαμε στο μάθημα.**
- Ένας τρόπος να φτιάξετε το ταμπλό είναι να έχετε ένα πίνακα διαστάσεων 6\*7.
- Το input μπορείτε να το δίνετε μέσω κουμπιών
  - ο Μπορείτε κάθε κελί να είναι ένα κουμπί, όταν η κίνηση είναι επιτρεπτή αλλάζει με τη σωστή φωτογραφία/χρώμα (Εικόνες έχουν αναρτηθεί στο elearn)
  - ο Επίσης μπορείτε να έχετε ένα πίνακα στη Javascript όπου θα γίνεται η αντιστοίχιση με το γραφικό (Δείτε το παράδειγμα με την τρίλιζα που δείξαμε στο μάθημα.)



### Σημειώσεις:

Μη ξεχνάτε τη χρήση του “use strict”; για την JavaScript. Το parse ενός JSON string γίνεται με χρήση της `JSON.parse(str)` που επιστρέφει το javascript object που αντιστοιχεί στο str, δεδομένου ότι το str είναι μια σωστή αναπαράσταση JSON . Μπορείτε να χρησιμοποιήσετε κάποιον online linter όπως ο `jshint` <http://jshint.com/> για να βελτιώσετε την ποιότητα του js κώδικά σας.

### Τρόπος Παράδοσης

Η Παράδοση θα γίνεται μέσω elearn. Θα πρέπει να παραδώσετε ένα zip που να περιέχει ένα φάκελο A2\_AM, όπου AM ο αριθμός μητρώου σας. Μπορείτε να έχετε ένα φάκελο για τις ασκήσεις 1 και 2, και ένας άλλο φάκελο για την άσκηση 3. Εκπρόθεσμες ασκήσεις **δεν θα γίνονται δεκτές**.

### Αντιγραφή!

Σε περίπτωση αντιγραφής θα μηδενίζονται άμεσα οι εργασίες όλων των εμπλεκόμενων. Άλλωστε κατά την εξέταση θα σας γίνουν πολλές ερωτήσεις που θα δείξουν κατά πόσο κατανοείτε τον κώδικα που έχετε παραδώσει. Οπότε ασχοληθείτε με την άσκηση και παραδώστε ότι καλύτερο μπορείτε και ότι δεν καταλαβαίνετε ρωτήστε στο elearn.

Καλή εργασία