

Πανεπιστήμιο Κρήτης

ΗΥ359 – Διαδικτυακός Προγραμματισμός (Web Programming)

Χειμερινό Εξάμηνο 2022/2023

Διδάσκων: **Μιχαήλ Μουνταντωνάκης**

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

2η Σειρά Ασκήσεων (Ατομική)

Διάρκεια: Τρίτη 25/10 – Παρασκευή 11/11

Αξία: 11% του τελικού σας βαθμού

(Συνολο ασκήσεων: 110 μονάδες)

Θεματική ενότητα: Εξοικείωση με JavaScript – APIs - Ajax - Maps

Άσκηση 1. Βελτίωση Φόρμας (25%)

Αποτελεί μέρος του συνόλου του project

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

a) (4%) 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) (7%) Ασφάλεια Password

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

- Να απαγορεύονται στο password οι ακολουθίες "helmeπα", "uoc" και "tuc" (μέσω javascript)
- 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 το password περιέχει τουλάχιστον 2 διαφορετικά σύμβολα, 1 κεφαλαίο γράμμα και 1 ένα μικρό γράμμα να του εμφανίζει strong password.
 - πχ. wadP0@\$1 (τουλάχιστον 1 μικρό, τουλάχιστον 1 μεγάλο, και σύμβολα @ και \$).
- Σε όλες τις άλλες περιπτώσεις να του εμφανίζει medium password

- πχ. aacdet3@ (δεν είναι ούτε weak ούτε strong).
- **Τέλος να μην επιτρέπει το submit αν το password είναι weak**
- 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) (8%) Έξτρα έλεγχοι για φοιτητή: Αν ο χρήστης είναι φοιτητής

- Εξαφανίζονται όλα τα έξτρα πεδία που έχουν να κάνουν με τον υπεύθυνο βιβλιοθήκης. (δείτε και παρακάτω)
- Θα πρέπει να γίνονται οι εξής έλεγχοι
 - το mail του φοιτητή να είναι ανάλογο με το πανεπιστήμιο που ανήκει (2%)
 - αν είναι στο uoc, το mail να τελειώνει σε uoc.gr
 - αν είναι στο tuc, το mail να τελειώνει σε tuc.gr
 - αν είναι στο helmera, το mail να τελειώνει σε helmera.gr
 - η ημερομηνία λήξης του πάσου να είναι μετά την ημερομηνία έναρξης (1%)
 - η διάρκεια του πάσου να μην ξεπερνά τα 6 χρόνια για προπτυχιακό φοιτητή, 2 χρόνια για μεταπτυχιακό και 5 χρόνια για διδακτορικό (4%)
 - όταν θα γίνει submit, τότε να γίνεται ο έλεγχος ότι όλα αυτά τα στοιχεία για το φοιτητή έχουν συμπληρωθεί (και είναι σωστά) (2%)

d) (5%) Επιλογές υπεύθυνου βιβλιοθήκης: Αν ο χρήστης επιλέξει ότι δεν είναι φοιτητής και είναι υπεύθυνος βιβλιοθήκης, τότε να γίνονται τα εξής:

- Εξαφανίζονται όλες οι επιλογές που έχουν να κάνουν με τους φοιτητές (πάσο κλπ)
 - Συγκεκριμένα
 - Τύπος Φοιτητή
 - Αριθμός Πάσου
 - Ημερομηνία έναρξης πάσου
 - Ημερομηνία λήξης πάσου

- Πανεπιστήμιο/Ανώτατη σχολή στην Κρήτη
- Τμήμα
 - Φυσικά σε αυτήν την περίπτωση δεν είναι τα συγκεκριμένα πεδία required.
- Αν ο χρήστης είναι υπεύθυνος βιβλιοθήκης, τότε να αλλάζει όνομα
 - το πεδίο *Διεύθυνση Οικίας*, και να λέγεται *Διεύθυνση Βιβλιοθήκης*
- Να εμφανίζεται ένα έξτρα text πεδίο “ονομα βιβλιοθήκης” (required)
- Να εμφανίζεται ένα έξτρα textarea πεδίο “Πληροφορίες και ωράριο βιβλιοθήκης” (required)

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 [20%]

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.

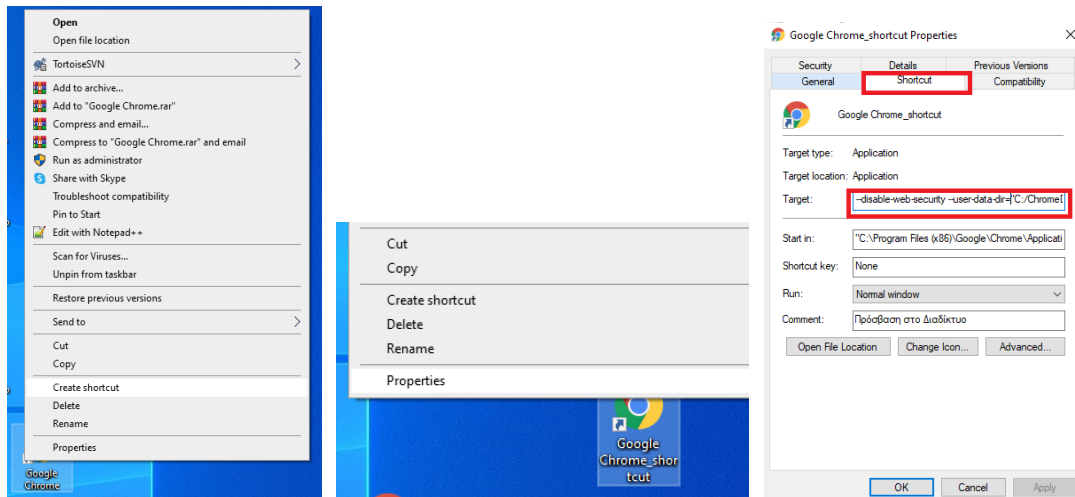
ΠΡΟΣΟΧΗ!!!

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.

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://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_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 Φιδάκι και Σκάλες (CSD Edition) [55%+10% bonus]

71	72 HY-335 Βαθμός 4	73 Πτυχιακή Βαθμός 9	74	75	76	77	78 HY-380 Βαθμός 4	79	80 Πτυχίο!!!
61	62	63	64	65 HY-340 Βαθμός 2	66	67 HY-340 Βαθμός 9	68	69	70
51	52	53	54 HY-345 Βαθμός 7	55	56	57	58 HY-352 Βαθμός 2	59 HY-255 Βαθμός 4	60 Σε πήραν στο MSc!
41	42 HY-359 Βαθμός 10	43	44 HY-280 Βαθμός 3	45	46	47	48	49	50
31	32	33	34	35	36	37 HY-360 Βαθμός 8	38	39	40
21 HY-240 Βαθμός 10	22	23	24	25	26	27	28 HY-225 Βαθμός 4	29	30
11	12	13 Αγγλικά I Βαθμός 3	14	15	16 HY-252 Βαθμός 9	17	18	19	20 HY-180 Βαθμός 3
1 Πέρασες CSD!!!	2	3	4	5 HY-120 Βαθμός 8	6	7	8	9	10

Φτιάξτε μέσω HTML και Javascript το παιχνίδι Φιδάκι και Σκάλες.

Για να εξοικειωθείτε με τη Javascript, θα φτιάξετε το κλασικό παιχνίδι **Φιδάκι και Σκάλες (CSD Edition)**, χρησιμοποιώντας μόνο **HTML, CSS και Javascript**. Στο παιχνίδι αυτό υπάρχουν 80 τετραγωνάκια, και ένα ταμπλό 8x10 (8 γραμμές και 10 στήλες). Όταν ο παίκτης περνάει ένα μάθημα στο csd, τότε ανεβαίνει με τη σκάλα στο αντίστοιχο τετράγωνο. Όταν κόβεται στο μάθημα, συναντάει ένα φίδι και γυρνάει στη θέση που βρίσκεται η ουρά του φιδιού.

Κάθε παίκτης έχει στην διάθεσή του 1 πiónι, το οποίο μετακινείται μόνο με ένα και μοναδικό ζάρι. Ο σκοπός του παιχνιδιού είναι να φτάσει κάποιος στον τερματισμό. Υποθέτουμε ότι οι 2 παίκτες μοιράζονται την ίδια οθόνη και το ίδιο ποντίκι.

Βοηθητικό Υλικό για να ξεκινήσετε

Σας δίνεται κώδικας (και σχετικό presentation) ο οποίος

- αρχικοποιεί και δείχνει το γραφικό του ταμπλό
- αρχικοποιεί τις θέσεις σε ένα object, για να ξέρετε τι είναι κάθε θέση
 - απλή, με σκάλα, με φιδάκι, με python effect
- έχει ένα παράδειγμα για το πως θα φαίνονται τα πiónια όταν κινούνται, ουσιαστικά αλλάζουν οι φωτογραφίες.

α) Σχεδίαση Αρχικής HTML σελίδας (10%)

Μπορείτε να σχεδιάσετε όπως θέλετε τη σελίδα σας. Για διευκόλυνση, σας δίνεται έτοιμος κώδικας που φτιάχνει τα τετραγωνάκια. Επίσης σας συνιστώ όταν κινούνται τα πιόνια, απλά να αλλάζετε τις εικόνες από τα αντίστοιχα τετράγωνα που επηρεάζονται. Σας δίνονται οι εικόνες για όλα αυτά, αλλά και σχετικός κώδικας.

Στην αρχική σελίδα πρέπει να έχετε εκτός από αυτά που δίνονται, ένα info box (μπορεί να είναι ένα άλλο div) που ο σκοπός του είναι να αναγράφει ποιος παίκτης παίζει, να δείχνει το αποτέλεσμα του ζαριού, αν ο παίκτης έχει το python effect, κλπ. Επίσης χρειάζεται ένα κουμπί για το ζάρι, που θα δίνει ένα τυχαίο αριθμό από το 1 έως το 6.

β) Συναρτήσεις Javascript (45%)

Θα πρέπει να έχετε τις κατάλληλες συναρτήσεις που να ελέγχουν

- **Αρχικοποίηση νέου παιχνιδιού (3 μονάδες) → `newGame()`**
 - ο Δίνει σειρά σε κάποιο παίκτη, αρχικοποιεί το infobox, μεταβλητές κλπ
- **Κίνηση Παίκτη (42 μονάδες)**
 - ο Θα έχετε έναν event handler που θα ενεργοποιείται μόλις πατηθεί το ζάρι να μετακινείται το πιόνι στη σωστή θέση με βάση τον αριθμό που έφερε **(20 μονάδες) → `play()`**
 - Απλή κίνηση σε κουτάκι που δεν έχει σκάλα ή φίδι **(6 μονάδες)**
 - Κίνηση σε κουτάκι με σκάλα **(7 μονάδες)**
 - Κίνηση σε κουτάκι με φίδι **(7 μονάδες)**
 - ο Συνάρτηση για το ποιος παίκτης παίζει **(1 μονάδα) → `getPlayerTurn()`**
 - ο Αλλαγή σειράς **(4 μονάδες) → `changePlayerTurn()`**
 - Αν ο παίκτης φέρει 6 ξαναπαίζει **(2 μονάδες)**
 - Αλλιώς αλλαγή σειράς **(2 μονάδες)**
 - ο Έλεγχος αν ο παίκτης κέρδισε **(7 μονάδες) → `hasPlayerWon()`**
 - Για να κερδίσει ο παίκτης πρέπει να φέρει ακριβώς τη ζαριά που χρειάζεται για να φτάσει στο κουτάκι 80. Αλλιώς πηγαίνει προς τα πίσω. **(3 μονάδες)**
 - Τέλος παιχνιδιού όταν φτάνει στο κουτάκι 80 **(2 μονάδες)**
 - Επιστροφή μηνύματος στους χρήστες και να τερματίζει το παιχνίδι **(2 μονάδες)**
 - ο Σωστή ανανέωση σελίδας HTML με κάθε κίνηση **(5 μονάδες) → `updateGUI()`**
 - Να ανανεώνει το ταμπλό, ζάρι, infoBox κλπ
 - ο Έξτρα Κανόνας - Python Effect **(5 βαθμοί)**
 - Αν ο παίκτης πέσει σε κουτάκι με το python της logo, τότε κερδίζει το python effect και γίνεται φίλος με τα φίδια. Αυτό σημαίνει ότι δεν επηρεάζεται πλέον από τα φίδια, οπότε και να πέσει σε κάποιο τέτοιο κουτάκι, παραμένει στη θέση αυτή.

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.

Θεωρείστε ότι τα πιόνια ξεκινάνε απ έξω, όχι από τη θέση 1.

- **πχ την πρώτη φορά που θα παίξει ο παίκτης, αν τύχει 3, θα πάει στη θέση 3.**

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 (10%)

- **Παιχνίδι με Cheat Mode (5 μονάδες):** Να ζητείται ένα όνομα για κάθε παίκτη από τον χρήστη. Αν ο παίκτης δώσει το όνομα "tsitsipas" να μπαίνει σε cheat mode, και ο παίκτης να φέρνει την πρώτη φορά που θα παίξει 5, τη δεύτερη 4 και την τρίτη 4. Έτσι θα φτάνει με 3 κινήσεις κατευθείαν στο τέλος και θα κερδίζει, χωρίς να το καταλάβει ο άλλος παίκτης. Συγκεκριμένα θα γίνεται η εξής διαδρομή:
 - 1η ζαριά→ Φέρνει 5, οπότε πάει στο κουτάκι 5 και με τη σκάλια πάει στο κουτάκι 33.
 - 2η ζαριά→ Φέρνει 4, οπότε πάει στο κουτάκι 37 και με τη σκάλια πάει στο κουτάκι 56.
 - 3η ζαριά→ Φέρνει 4, οπότε πάει στο κουτάκι 60 και με τη σκάλια πάει στο κουτάκι 80, όπου και κερδίζει.
- **Παιχνίδι με υπολογιστή (5 μονάδες):** Μπορείτε να υλοποιήσετε ένα παιχνίδι (σε διαφορετικό αρχείο) όπου να παίζει και ο υπολογιστής ενάντια στον παίκτη.

Hints

- **SOS. Δείτε και το παράδειγμα με τα games που δείξαμε στο μάθημα.**
- Το input μπορείτε να το δίνετε μέσω κουμπιών
 - ο Μπορείτε το ζάρι να είναι ένα κουμπί, όταν η κίνηση είναι επιτρεπτή αλλάζει με τη σωστή φωτογραφία/χρώμα (Εικόνες έχουν αναρτηθεί στο elearn)
- Παρακάτω φαίνονται οι θέσεις που έχουν φιδάκι ή σκάλα και πού οδηγούν, αλλά και οι θέσεις με python effect
- **Έχει δοθεί και σε πίνακα javascript μέσα στον κώδικα**

Ειδικές Θέσεις στο Ταμπλό

Θέση	Κατηγορία	Μεταφορά Από	Μεταφορά Προς
5	Σκάλα	5	33
13	φίδι	13	11
16	Σκάλα	16	36
20	Φίδι	20	10
21	Σκάλα	21	61
28	Φίδι	28	7
29	Python effect	-	-

37	Σκάλα	37	56
42	Σκάλα	42	53
44	Φίδι	44	34
46	Python effect	-	-
54	Σκάλα	54	64
58	Φίδι	58	48
59	Φίδι	59	36
60	Σκάλα	60	80
65	Φίδι	65	25
67	Σκάλα	67	77
72	Φίδι	72	52
73	Σκάλα	73	76
78	Φίδι	78	69

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