

# Γλώσσες και Τεχνολογίες Ιστού - Εργαστήριο 6:

## Αποσφαλμάτωση - Debugging

**Γενική οδηγία:** Ανοίξτε τα αρχεία από τους συνδέσμους που υπάρχουν στην κάθε Άσκηση από το κάτω μέρος της σελίδας, σώστε τα τοπικά, ώστε να μπορείτε να κάνετε debugging στον υπολογιστή σας.

### Άσκηση 1 - Εντοπισμός συντακτικών λαθών

Η σωστή επιλογή και χρήση του browser για τον εντοπισμό συντακτικών λαθών είναι πολύ σημαντικό βήμα στην αποσφαλμάτωση.

Με τη χρήση των διαφόρων browser εντοπίστε και διορθώστε τα λάθη της παρακάτω σελίδας.

#### [lab6-1](#)

Στην παραπάνω σελίδα θα έπρεπε όταν ο χρήστης έκανε click (focus) σε ένα textbox τότε το background του να γινόταν κίτρινο. Όταν έφευγε από αυτό (loosefocus) τότε αυτό να γινόταν ξανά λευκό αν ο χρήστης είχε γράψει κάτι σε αυτό ή κόκκινο αν ο χρήστης δεν είχε γράψει τίποτα σε αυτό. Ωστόσο λόγω των συντακτικών και λογικών σφαλμάτων που περιέχει ο κώδικας θα παρατηρήσετε ότι αυτό δεν είναι το αποτέλεσμα της εκτέλεσης. Διορθώστε τα λάθη.

### Άσκηση 2 - Παρακολούθηση τιμών με window.alert

Με χρήση της window.alert εμφανίζεται ένας διάλογος pop-up στο παράθυρο του browser.

Η παρακάτω σελίδα υλοποιεί ένα παιχνίδι "Βρες τον αριθμό". Το πρόγραμμα φαίνεται να "κολλάει" στο loop εισαγωγής αριθμών. Ανοίξτε την προβολή πηγαίου κώδικα HTML, αντιγράψτε την σελίδα σε δική σας και τροποποιήστε κατάλληλα τον κώδικα εισάγοντας τα alert μηνύματα που χρειάζονται για να εντοπίσετε την εξέλιξη των τιμών των μεταβλητών που σας ενδιαφέρουν και να εντοπίσετε το πρόβλημα.

#### [lab6-2](#)

### Άσκηση 3 - Παρακολούθηση τιμών με χρήση του console object

Η παρακάτω σελίδα υλοποιεί μια αλλαγή εικόνας με την μετακίνηση του δείκτη επάνω στην εικόνα και επαναφορά όταν ο δείκτης φεύγει από αυτήν. Υπάρχει όμως ένα λάθος στην λειτουργία του κώδικα για την λειτουργία της εικόνας δεξιά.

#### [lab6-3](#)

Εικόνες που χρειάζονται στον ίδιο φάκελο:

[SandyBeach.jpg](#)

[PalmBeach.jpg](#)

[Sunset.jpg](#)

Θα χρησιμοποιήσουμε το αντικείμενο console που μας παρέχει τις μεθόδους log, info και error (διάβασε [εδώ](#)) ώστε να κατανοήσουμε την λειτουργία και να διορθώσουμε το λάθος.

Εισάγετε τον κατάλληλο κώδικα στο παρακάτω με χρήση του console ώστε να εμφανίζεται το id του img tag το οποίο αλλάζει, το είδος της αλλαγής (changing / restoring) και η ονομασία του αρχείου εικόνας.

## Άσκηση 4 - Η συνάρτηση eval

Κατανοείτε το παρακάτω παράδειγμα:

lab6-4

**ΕΝΗΜΕΡΩΣΗ ΙΕΕ1405 (2020):** Εντοπίστε την εντολή Javascript που φαίνεται ότι λειτουργεί χωρίς να υπάγεται σε κάποια αντικείμενο (JavaScript Global). Για να μάθετε την λειτουργία της όπως και για κάθε άγνωστη μέθοδο αναζητήστε την στο w3schools. Συγκεκριμένα για τις [global μεθόδους εδώ](#).

Κατεβάστε τον κώδικα της παρακάτω σελίδας που περιέχει δηλώσεις για 4 μεταβλητές

[lab6-4b](#)

και εντάσσοντας την λογική του lab6-4 υπολογίστε τα αποτελέσματα των παρακάτω πράξεων

a+10

b+'def'

c+a

b+d

a+d

## Άσκηση 5 - Διαχείριση λαθών με onerror

Δείτε τον παρακάτω κώδικα ο οποίος φορτώνει δύο φορές την εικόνα "1.gif". Επεκτείνετε το κώδικα για το στοιχείο Img2 ώστε σε περίπτωση δεν υπάρχει το 1.gif να φορτώνει ένα άλλο image.

<HTML>

<HEAD>

<TITLE> Διαχείριση λαθών</TITLE>

</HEAD>

<BODY>

<H1>Διαχείριση λαθών</H1>

<P>Image χωρίς διαχείριση λαθών.

<IMG SRC="1.gif" NAME="Img1">

<P>Image με διαχείριση λαθών.

<IMG SRC="1.gif" NAME="Img2">

</BODY>

</HTML>

## Άσκηση 6 - Διαχείριση λαθών με `window.onerror`

Κατανοείτε πως συλλαμβάνονται τα λάθη με το παρακάτω παράδειγμα :

[lab6-7](#)

Χρησιμοποιήστε την τεχνική που υπάρχει στο 6-7, τοποθετώντας την μέσα στον κώδικα της άσκησης 1, έτσι ώστε να ελέγχετε το πώς εμφανίζονται τα μηνύματα λάθους.

[lab6-1](#)

## Άσκηση 7 - Διαχείριση λαθών με `try - catch`

**ΕΝΗΜΕΡΩΣΗ ΙΕΕ1405 (2020):** Κατανοήστε πως λειτουργούν τα μπλοκ `try-catch` [μέσω αυτού του παραδείγματος](#).

Τροποποιήστε με την χρήση `try - catch` τον κώδικα της παρακάτω σελίδας ώστε σε περίπτωση που ο χρήστης έχει εισάγει αλφαριθμητικό αντί αριθμού στο πλαίσιο η `function rndNum` να επιστρέφει -1 και η `function tellUser` να τυπώνει ένα μήνυμα λάθους.

[lab6-8](#)

## Άσκηση 8 - Χρήση `Debugger`

Χρησιμοποιήστε το debugger από του browser ή του VSCode ώστε να διακόψετε την ροή του παρακάτω προγράμματος σε συγκεκριμένα σημεία και να παρακολουθήσετε την εξέλιξη των μεταβλητών, δημιουργώντας `watches` για τις μεταβλητές, ώστε να εντοπίσετε το λογικό σφάλμα στην εκτέλεσή του.

[lab6-9](#)