

Lab 06 — Εισαγωγή στη JavaScript, DOM, events & φόρμες

Σκοπός και μαθησιακά αποτελέσματα

Σε αυτή την άσκηση θα προσθέσετε **βασική διαδραστικότητα** στον ιστότοπό σας χρησιμοποιώντας **JavaScript**. Θα δείτε πώς:

- συνδέουμε JavaScript σε σελίδα HTML,
- διαβάζουμε/αλλάζουμε στοιχεία της σελίδας μέσω DOM,
- αντιδρούμε σε ενέργειες χρήστη (events),
- κάνουμε έναν απλό έλεγχο φόρμας στην contact.html.

Στο τέλος του εργαστηρίου η σελίδα σας δεν θα είναι πια στατική: θα αντιδρά σε κλικ, θα ελέγχει φόρμες και θα ενημερώνει τον χρήστη.

Με την ολοκλήρωση του εργαστηρίου θα μπορείτε να:

1. Εξηγείτε τον ρόλο της JavaScript στη λειτουργία ενός σύγχρονου ιστότοπου και τη διάκριση μεταξύ δομής (HTML), εμφάνισης (CSS) και λογικής/συμπεριφοράς (JavaScript).
2. Ενσωματώνετε κώδικα JavaScript σε υπάρχουσες σελίδες HTML με σωστή δομή και κατανόηση της ροής εκτέλεσης.
3. Χειρίζεστε το DOM (Document Object Model) για να επιλέγουν, να τροποποιούν ή να προσθέτουν δυναμικά στοιχεία σε μια ιστοσελίδα.
4. Αξιοποιείτε γεγονότα (events) όπως click, input, submit για να δημιουργούν απλές αλληλεπιδράσεις χρήστη-σελίδας.
5. Υλοποιείτε βασικούς ελέγχους φόρμας (validation) με JavaScript, εξασφαλίζοντας ότι τα δεδομένα χρήστη είναι πλήρη ή ορθά πριν αποσταλούν.
6. Αναπτύσσετε διαδραστικά χαρακτηριστικά, όπως αλλαγή θέματος (light/dark mode), με χρήση λογικής ελέγχου (if, classList.toggle()).
7. Αναγνωρίζετε και επιλύετε συνήθη σφάλματα JavaScript μέσω του Developer Console, καλλιεργώντας δεξιότητες debugging.

Σύνδεση με τη θεωρία

Η παρούσα εργαστηριακή άσκηση σχετίζεται με τη διάλεξη «Η γλώσσα Javascript». Ο πίνακας που ακολουθεί κάνει μία σύντομη αναφορά στις κεντρικές έννοιες.

Πού γράφουμε JS

- α) μέσα σε `<script> ... </script>` στο τέλος του `<body>` (για αυτό το εργαστήριο)
- β) ή σε εξωτερικό αρχείο `.js` και το κάνουμε `<script src="app.js"></script>`.

Βασική σύνταξη

```
// μεταβλητές  
let message = "Γεια σας!";  
const year = 2025;  
  
// συνάρτηση  
function sayHello() {  
    console.log("Καλησπέρα από την JS!");  
}
```

DOM = Document Object Model

Μέσω JS μπορούμε να "πιάσουμε" στοιχεία HTML:

```
const title = document.getElementById("main-title");  
title.innerText = "Νέος τίτλος!";
```

Events (Γεγονότα):

```
button.addEventListener("click", function () {  
    alert("Πατήσατε το κουμπί!");  
});
```

Μέρος Α — Προσθήκη JavaScript στο index.html

Βήματα

1. Κατεβάστε όλα τα περιεχόμενα από το φάκελο «Lab06» στο eclass (εκτός της εκφώνησης της εργαστηριακής άσκησης) και τοποθετήστε τα σε ένα νέο φάκελο που θα δημιουργήσετε με όνομα lab06-JS.
2. Ανοίξτε το index.html.
3. Πριν από το </body> προσθέστε

```
<script>
  // απλό sanity check
  console.log("JS ενεργή.");
  alert("Καλωσορίσατε στη διαδραστική έκδοση της σελίδας!");
</script>
</body>
```

4. Ανοίξτε το DevTools → Console και ελέγξτε ότι εμφανίζεται το μήνυμα
5. Για το υπόλοιπο μέρος του εργαστηρίου μπορείτε να τοποθετήσετε τη συνάρτηση alert() σε σχόλιο.

Μέρος Β — Πρόσβαση & αλλαγή στοιχείων DOM

Τώρα θα αλλάξουμε δυναμικά τον τίτλο ή θα εμφανίσουμε την τρέχουσα ημερομηνία κάτω από το header

Βήματα

1. Στη σελίδα index.html, κάτω από το header προσθέστε
 - μία παράγραφο με id «welcome-msg» και κείμενο «Πατήστε το κουμπί για μήνυμα καλωσορίσματος»
 - ένα κουμπί με id «show-welcome» και κείμενο «Δείξε μήνυμα»
2. Προσθέστε στο προηγούμενο script κώδικα ώστε όταν πατηθεί το κουμπί να αλλάζει το κείμενο της παραγράφου που προσθέσατε προηγουμένως εμφανίζοντας το μήνυμα «Καλωσορίσατε, επισκέπτη! Χαιρόμαστε που σας βλέπουμε εδώ.»

Σημείωση: α) πρέπει να γίνει χρήση πρότυπης συμβολοσειράς, β) η λέξη «επισκέπτη», πρέπει να αποτελεί την τιμή μεταβλητής με όνομα «username» η οποία σε πραγματικές συνθήκες θα αποθηκεύει το όνομα χρήστη από μία φόρμα login. Για να γίνει αυτό θα χρειαζόμασταν μία νέα σελίδα login.html, όπου με το αντικείμενο localStorage θα αποθηκεύσουμε προσωρινά το όνομα χρήστη στο browser και θα το εμφανίσουμε στο html με τη φόρτωση της σελίδας index.html (event DOMContentLoaded).

Μέρος Γ — Εξαφάνιση/εμφάνιση header με scrolling

Στόχος είναι να κρύβουμε το header όταν ο χρήστης κάνει scroll προς τα κάτω, και να το επαναφέρουμε όταν κάνει scroll προς τα πάνω.

Βήματα

1. Προσθέστε στο CSS του header τον ακόλουθο κώδικα

```
header { transition: transform .8s ease; } /* στο δικό σου header που είναι ήδη fixed */
```

2. Κάτω από το CSS του header πρόσθεσε τον ακόλουθο κώδικα

```
.header--hidden {transform: translateY(-100%);}
```

3. Στο υπάρχον script προσθέστε τον παρακάτω κώδικα γράφοντας εσείς τις 4 γραμμές κώδικα που λείπουν

```
//Δημιουργεί μια μεταβλητή lastY και της δίνει την τρέχουσα κάθετη θέση κύλισης (scroll) της σελίδας.
```

```
//Η window.scrollY δείχνει πόσα pixel έχεις κάνει scroll προς τα κάτω από την κορυφή της σελίδας.
```

```
//Παράδειγμα: Αν είσαι στην κορυφή → scrollY = 0 Αν έχεις κατέβει 300px → scrollY = 300
```

```
let lastY = window.scrollY;
```

```
//Επιλέγει το στοιχείο <header> από τη σελίδα και το αποθηκεύει στη μεταβλητή header
```

```
Γραμμή κώδικα 1
```

```
//Προσθέτει έναν “ακροατή γεγονότος” (event listener) που ενεργοποιείται κάθε φορά που ο χρήστης κάνει scroll στη σελίδα.
```

```
window.addEventListener('scroll', () => {
```

```
const y = window.scrollY; //Κάθε φορά που γίνεται scroll, παίρνει την τρέχουσα θέση κύλισης και την αποθηκεύει στη μεταβλητή y.
```

```
const goingDown = y > lastY; //Ελέγχει αν ο χρήστης πηγαίνει προς τα κάτω (το y τώρα είναι μεγαλύτερο από το προηγούμενο lastY)
```

```
if (y > 100 && goingDown) {
```

```
Γραμμή κώδικα 2
```

```
Γραμμή κώδικα 3
```

```
Γραμμή κώδικα 4
```

```
}
```

```
lastY = y; //ενημερώνει τη μεταβλητή lastY με τη νέα θέση κύλισης, ώστε στην επόμενη κύλιση να μπορεί να συγκρίνει ξανά
```

```
});
```

Μέρος Δ — Εργασία με φόρμες (contact.html)

Βήματα

1. Ανοίξτε τη σελίδα contact.html
2. Προσθέστε στο στοιχείο <form> το id «contact-form»
3. Προσθέστε μία κενή παράγραφο (χωρίς δηλαδή να περιλαμβάνει κάποιο κείμενο) με id «form-status», κάτω ακριβώς από το στοιχείο <form>.
4. Διατηρήστε τον υπόλοιπο HTML + CSS κώδικα ως έχει
5. Προσθέστε το στοιχείο <script> για να γράψετε κώδικα JavaScript πριν από το </body>
6. Αναθέστε τα στοιχεία της φόρμας και της νέας παραγράφου που δημιουργήσατε στο βήμα 3, σε δύο μεταβλητές με όνομα form και status αντίστοιχα.
7. Δημιουργείτε μία μέθοδο addEventListener για τη μεταβλητή form που θα παρακολουθεί το συμβάν πατήματος του κουμπιού «Αποστολή» (δημιουργήστε δηλαδή ένα “ακροατή” στο event submit της φόρμας)
8. Όταν ο χρήστης πατάει Υποβολή θα πρέπει να εκτελείται μία συνάρτηση με όρισμα το αντικείμενο event
9. Η συνάρτηση function (event) θα κάνει τα εξής:
10. Θα ακυρώνει την προεπιλεγμένη συμπεριφορά του browser (να στείλει τη φόρμα και να γίνει reload)
11. Θα αναθέτει τις τιμές των πεδίων εισαγωγής ονόματος, email και μηνύματος σε τρεις μεταβλητές με όνομα name, email και msg αντίστοιχα (χρησιμοποιείτε τη μέθοδο trim() για να αφαιρέσετε τους κενούς χαρακτήρες πριν και μετά το κείμενο των πεδίων)
12. Θα ελέγχει αν κάποιο από ανωτέρω πεδία είναι κενό
13. Αν είναι κάποιο κενό θα
 - Γράφει το προειδοποιητικό μήνυμα στο στοιχείο status «⚠ Παρακαλούμε συμπληρώστε όλα τα υποχρεωτικά πεδία (Όνομα, Email και Μήνυμα).»
 - Βάζει κόκκινο χρώμα στο στοιχείο status
 - Θα σταματά την εκτέλεση του handler με την εντολή return
14. Αν είναι όλα τα πεδία συμπληρωμένα θα
 - Εμφανίζει το μήνυμα επιτυχίας «☒ Ευχαριστούμε, \${name}! Θα σας απαντήσουμε σύντομα στο \${email}.»
 - Χρησιμοποιεί template literal (backticks `...`) για να “δέσει” τις τιμές name και email μέσα στο κείμενο.
 - Αλλάζει το χρώμα σε πράσινο

Μέρος Ε (προαιρετικό) — Μετρητής χαρακτήρων στο μήνυμα

Βήματα

1. Αναθέστε την τιμή του πεδίου εισαγωγής μηνύματος σε μία μεταβλητή με όνομα `msgField`
2. Δημιουργείστε μία μεταβλητή `maxLen` με τιμή 200 (αντιπροσωπεύει το μέγιστο πλήθος χαρακτήρων που επιτρέπεται στο μήνυμα)
3. Δημιουργείστε δυναμικά ένα νέο στοιχείο `<p>` αποθηκεύστε το στη μεταβλητή `counter`
4. Αρχικοποιείστε το κείμενο του νέου στοιχείου σε «Χαρακτήρες: 0 / `{maxLen}`» χρησιμοποιώντας πρότυπες συμβολοσειρές
5. Προσθέστε 0.25rem κενό πάνω από το νέο στοιχείο για αισθητικούς λόγους
6. Τοποθετείστε το νέο `<p>` (τον μετρητή) ακριβώς κάτω από το πεδίο `msg` (χρησιμοποιείστε την ιδιότητα `parentNode` για να βρείτε το γονά του `msgField` (δηλαδή το `container` του `textarea`), ώστε να προστεθεί το νέο στοιχείο στο τέλος του)
7. Προσθέτει “ακροατή” (event listener) για το γεγονός `input`, δηλαδή κάθε φορά που ο χρήστης πληκτρολογεί ή σβήνει κάτι μέσα στο `textarea`.
8. Ο ακροατής θα εκτελεί μία συνάρτηση που θα
 - αναθέτει το τρέχον πλήθος χαρακτήρων που υπάρχουν στο πεδίο (`value.length`) στη μεταβλητή `len`
 - Ενημερώνει το κείμενο του μετρητή ώστε να δείχνει τον νέο αριθμό χαρακτήρων σε πραγματικό χρόνο