## JavaScript – Μια μικρή Εισαγωγή

Ξεκινώ τοποθετώντας την επόμενη ετικέτα μέσα στο αρχείο index.html και στο εσωτερικό της διπλής ετικέτας <head></head> όπως φαίνεται πιο κάτω.

<link href="css/style.css" rel="stylesheet" /> <link href="css/queries.css" rel="stylesheet" /> <script defer src="js/script.js"></script>

Ανοίγω το αρχείο script.js. Στην κορυφή του αρχείου πληκτρολογώ

console.log("Hello World!");

Σώζω και βλέπω τις αλλαγές.

Μέσα στη σελίδα δεν παρατηρώ κάποια αλλαγή. Αν όμως κάνω δεξί κλικ επιλέξω Έλεγχος' και ανοίξω την καρτέλα **Console** θα δω ότι το κείμενο "Hello World!" εμφανίστηκε στο παράθυρο της καρτέλας.



Η καρτέλα Console χρησιμεύει για να τεστάρω JavaScript κώδικες.

Η JavaScript υποστηρίζει τη χρήση μεταβλητών. Λέμε στον browser να δεσμεύσει μια μεταβλητή να της δώσει ένα όνομα και να εκχωρήσει μέσα σ' αυτή μια τιμή με την πιο κάτω εντολή:

const myName="Mary Fanarioti";

Με την εντολή αυτή λέμε στο browser να δεσμεύσει μια μεταβλητή να της δώσει το όνομα myName και να εκχωρήσει μέσα σ αυτή το κείμενο Mary Fanarioti. Αν στη συνέχεια κάνω χρήση της εντολής

console.log(myName);

θα δω στην κονσόλα το περιεχόμενο της μεταβλητής myName.

Με την επόμενη εντολή λέω στον browser να εντοπίσει μέσα στο έγγραφο της σελίδας εκείνο το στοιχείο που φέρει την κλάση **heading-primary** και να το εκχωρήσει στη μεταβλητή h1.

const h1 = document.querySelector(".heading-primary");

Αν στη συνέχεια κάνω χρήση της εντολής

console.log(h1);

θα δω ότι η πρώτη εντολή εκχωρεί μέσα στην μεταβλητή h1 όλον τον html κώδικα που αφορά το στοιχείο heading-primary , ενώ μέσω της δεύτερης εντολής ο κώδικας αυτός τυπώνεται στην κονσόλα.

	Hello World!	<pre>script.js:1</pre>
	Mary Fanarioti	<pre>script.js:4</pre>
	<pre>     <h1 class="heading-primary">         " A healthy meal delivered to your door,</h1></pre>	<pre>script.js:5 every single</pre>
	day " 	
>	true	<u>script.js:30</u>

Με την επόμενη εντολή μπορώ να αλλάξω το κείμενο του στοιχείου heading-primary

h1.textContent=myName;

Αν σώσω και κάνω ανανέωση τη σελίδα θα δω στη θέση της επικεφαλίδας h1 Το περιεχόμενο της μεταβλητής myName



Με τις επόμενες εντολές αλλάζω το φόντο του στοιχείου και προσθέτω περιμετρικά padding 5rem.

```
h1.style.backgroundColor="red";
h1.style.padding: 5rem;
```



Αν θέλω η αλλαγή να συμβαίνει με το που πατάει ο χρήστης πάνω στην επικεφαλίδα χρησιμοποιώ την επόμενη έκφραση

```
h1.addEventListener("click", function () {
    h1.textContent = myName;
    h1.style.backgroundColor = "red";
    h1.style.padding = "5rem";
});
```

Με αυτή την έκφραση λέω στον browser ότι μόλις ανιχνεύσει πάτημα (click) ποντικιού οπουδήποτε μέσα στο στοιχείο h1 να εκτελέσει τη συνάρτηση function() η οποία περιλαμβάνει τις εντολές που περιγράψαμε πριν και οι οποίες αλλάζουν το περιεχόμενο της h1, το χρώμα του φόντου και τα paddings.

Σώζω και αφού ανανεώσω τον κώδικα της σελίδας πατώ πάνω στην επικεφαλίδα.

Πριν το click

#### Μετά το click

# A healthy meal delivered to your door, every single day

The smart 365-days-per-year food subscription that will make you eat healthy again. Tailored to your personal tastes and nutritional needs.

Start eating well

Learn more ↓



Με τον επόμενο κώδικα εμφανίζω στο υποσέλιδο ακριβώς δίπλα από το Copyright το τρέχων έτος μέσω JavaScript.

const yearEl = document.querySelector(".year"); const currentYear = new Date().getFullYear(); yearEl.textContent = currentYear;

Προηγουμένως στον κώδικα του footer πρέπει να έχω φροντίσει να περικλείσω το έτος στην διπλή ετικέτα <span></span> όπως φαίνεται πιο κάτω

<span class="year">2027</span>

Σώζω και βλέπω το αποτέλεσμα στη σελίδα. Το τρέχον έτος εμφανίζεται δίπλα στο copyright.

Copyright © 2024 by Omnifood, Inc. All rights reserved.

Σ' αυτό το σημείο απομακρύνουμε όλους τους κώδικες που προσθέσαμε στο αρχείο script.js εκτός από τις 3 τελευταίες εντολές που εμφανίζουν το τρέχων έτος μέσα στη σελίδα.

#### **Mobile Navigation**

Το μενού πλοήγησης εμφανίζεται στις κινητές συσκευές οποτεδήποτε η κλάση **nav-open** είναι παρούσα στην ετικέτα αρχής header. Και η κλάση αυτή πρέπει να προστίθεται στην ετικέτα header κάθε φορά που ο χρήστης πατά στο κουμπί μενού. ακολουθούν οι JavaScript κώδικες.

```
const btnNavEl = document.querySelector(".btn-mobile-nav");
const headerEl = document.querySelector("header");
```

```
btnNavEl.addEventListener("click", function () {
    headerEl.classList.toggle("nav-open");
```

**});** 

Η πρώτη εντολή επιλέγει το κουμπί και η δεύτερη το στοιχείο header. Η τρίτη εντολή ανιχνεύει τα κλικ που γίνονται πάνω στο κουμπί πλοήγησης. Σε κάθε κλικ καλείται η συνάρτηση function(). Αυτή περιέχει μια εντολή η οποία στην περίπτωση που η κλάση navopen απουσιάζει από την ετικέτα header την προσθέτει, ενώ στην περίπτωση που η κλάση είναι παρούσα την αφαιρεί.

Σώζω και βλέπω τη λειτουργία. Προσοχή πρέπει να την τεστάρω για τις μικρές οθόνες.

Επίσης στο αρχείο queries.css και στο media query που αφορά τα πλάτη των Tablets -από τα οποία ξεκινά η εμφάνιση του μενού πλοήγησης - εντοπίζω τον πιο κάτω κανόνα και προσθέτω την ιδιότητα z-index με τιμή 1000 προκειμένου το κουμπί μενού να βρίσκεται πάνω από όλα τα στοιχεία της σελίδας.

```
.btn-mobile-nav {
display: block;
z-index: 1000;
}
```

#### Σύνδεσμοι που οδηγούν σε άλλα σημεία της ίδιας σελίδας - Scrolling Effect

Όταν ο χρήστης πατά στους συνδέσμους της μπάρας πλοήγησης θα πρέπει να οδηγείται στις αντίστοιχες ενότητες με χρήση ενός scrolling effect.



Πρόκειται να δημιουργήσουμε συνδέσμους που όταν οδηγούν σε συγκεκριμένα σημεία μέσα στη σελίδα. Έτσι όταν ο χρήστης πατά στο σύνδεσμο How it works πρέπει να οδηγείται στην ενότητα How it works της σελίδας. Όταν πατά στο σύνδεσμο Meals πρέπει να οδηγείται στην ενότητα Meals της σελίδας. Όταν πατά στο σύνδεσμο Testimonials πρέπει να οδηγείται στην ενότητα Testimonials της σελίδας κ.ο.κ.

Για να το πετύχουμε αυτό ακολουθούμε την εξής διαδικασία. Σε κάθε μία από τις ενότητες στις οποίες θέλω να οδηγούν οι σύνδεσμοι όταν πατηθούν προσθέτω ένα μοναδικό όνομα μέσω της παραμέτρου id. Η παράμετρος αυτή τοποθετείται στο εσωτερικό της ετικέτας αρχής <section> της κάθε ενότητας, όπως φαίνεται πιο κάτω.

```
<section class="section-how" id="how">
</section>
</section class="section-meals" id="meals">
</section class="section-meals" id="meals">
</section>
</section class="section-testimonials" id="testimonials">
</section class="section-testimonials"</section class="section-testimonials">
</section class="section-testimonials">
</section class="section-testimonials"</section class="section-testimonials">
</section class="section-testimonials"</section class="section-testimonials">
</section class="section-testimonials"</section class="section-testimonials">
</section class="section-testimonials"</section class="section-testimonials"</section clas
```

</section>

<section class="section-cta" id="cta">

</section>

Στη συνέχεια επιστρέφω στον κώδικα της μπάρας πλοήγησης όπου ως τιμή της παραμέτρου **href** του κάθε κουμπιού δίνω το **id** της ενότητας στην οποία πρέπει να οδηγεί ο σύνδεσμος όταν πατηθεί.

<nav></nav>
<ul></ul>
<li><a class="btn-menu" href="#how">How it works</a></li>
<li><a class="btn-menu" href="#meals">Meals</a></li>
<li><a class="btn-menu" href="#testimonials">Testimonials</a></li>
<li><a class="btn-menu" href="#pricing">Pricing</a></li>
<li><a class="btn-menu" href="#cta">Try for free</a></li>

Προσέξτε ότι πριν από το όνομα της ενότητας προηγείται το σύμβολο #.

Σώζω και τεστάρω τους συνδέσμους της μπάρας.

Συνεχίζω με τα κουμπιά της hero section. Το μεν αριστερό κουμπί πρέπει να οδηγεί στην ενότητα **cta**, το δε δεξί πρέπει να οδηγεί στην ενότητα **how.** Εντοπίζω μέσα στο αρχείο **index.html** τους κώδικες των κουμπιών και ενημερώνω κατάλληλα την παράμετρο **href** αυτών όπως φαίνεται πιο κάτω.

```
<a href="#cta" class="btn btn-left">Start eating well</a>
<a href="#how" class="btn btn-right">Learn more &darr;</a>
```

Σώζω και τεστάρω τους συνδέσμους.

Οι σύνδεσμοι που σαν τιμή της παραμέτρου **href** φέρουν το σύμβολο **#** οδηγούν στην αρχή της σελίδας όταν πατηθούν.

Προσθέτοντας την ιδιότητα scroll-behavior με τιμή smooth στον κανόνα της html πετυχαίνω μια ομαλή μετάβαση προς μια συγκεκριμένη ενότητα της σελίδας μετά το πάτημα ενός συνδέσμου.

# html { font-size: 62.5%; overflow-x: hidden; scroll-behavior: smooth;

}

Δυστυχώς αυτή η ιδιότητα δεν λειτουργεί στον Safari browser που είναι ο μόνος διαθέσιμος browser στα iPhone & iPad. Ακόμη και ο Google Chrome σ' αυτές τις συσκευές βασίζεται στον Safari για να λειτουργήσει.

Σ΄ αυτό το σημείο θα χρειαστεί να προσθέσουμε κάποιο κώδικα JavaScript που να υλοποιεί την ομαλή μετάβαση ανεξαρτήτως browser.

Επίσης αν δω τη σελίδα από κινητό και πατήσω σε ένα σύνδεσμο του μενού πλοήγησης θα μεταφερθώ στην αντίστοιχη ενότητα, επιστρέφοντας όμως στην κορυφή θα δω ότι το μενού παραμένει αφού έχει ρυθμιστεί να κλείνει μόνο όταν ο χρήστης πατήσει στο κουμπί close του μενού. Αν θέλουμε να κλείνει μόλις ο χρήστης πατήσει σε ένα από τα κουμπιά του μενού τότε θα πρέπει να προσθέσουμε JavaScript κώδικα μέσα στο αρχείο script.js

Ακολουθούν οι σχετικοί κώδικες.

```
// Scroll back to top
if (href === "#")
window.scrollTo({
   top: 0,
    behavior: "smooth",
}).
```

});

```
// Scroll to other links
```

```
if (href !== "#" && href.startsWith("#")) {
   const sectionEl = document.querySelector(href);
   sectionEl.scrollIntoView({ behavior: "smooth" });
}
```

```
//Close mobile navigation
if (link.classList.contains("btn-menu"))
headerEl.classList.toggle("nav-open");
});
```

Μ.ΦΑΝΑΡΙΩΤΗ

## **Sticky Navigation Bar**

Στο αρχείο index.html συνδυάζω την κλάση sticky με την ετικέτα αρχής <header>

<header class="sticky">

</header>

Στο αρχείο style.css προσθέτω τον επόμενο κανόνα

/*	* STICKY NAVIGATION */	
.sticky {		
	position: fixed;	
	top: 0;	
	bottom: 0;	
	width: 100%;	
	height: 8rem;	
	padding-top: 0;	
	padding-bottom: 0;	
	background-color: rgba(255, 255, 255, 0.97);	
	z-index: 999;	
	box-shadow: 0 1.2rem 3.2rem rgba(0, 0, 0, 0.03);	
}		

Σώζω και βλέπω τις αλλαγές.



Αν θέλω όμως η sticky bar να εμφανίζεται αφού μετακινηθώ κάτω από την hero section, θα χρειαστώ JavaScript κώδικα ο οποίος θα προσθέτει την κλάση sticky στην ετικέτα αρχής <body> μόλις η hero section βγει έξω από το διαθέσιμο viewport και θα την απομακρύνει μόλις η hero section έρθει ξανά στο προσκήνιο. Ακολουθεί ο κώδικας που πρέπει να προστεθεί μέσα στο αρχείο script.js

**ΠΡΟΣΟΧΗ!** Σ' αυτή την περίπτωση θα πρέπει να απομακρύνουμε την κλάση sticky από την ετικέτα αρχής <header>.

```
//Sticky Navigation
const sectionHeroEl = document.querySelector(".section-hero");
const obs = new IntersectionObserver(
 function (entries) {
  const ent = entries[0];
 if (ent.isIntersecting === false) {
  document.body.classList.add("sticky");
 }
 if (ent.isIntersecting === true) {
  document.body.classList.remove("sticky");
 }
},
 {
 // In the viewport
  root: null,
 threshold: 0,
  rootMargin: "-80px",
}
);
obs.observe(sectionHeroEl);
```