



DESIGN YOUR JOB

Web design education program





ΕΚΠΑΙΔΕΥΤΙΚΟ ΠΡΟΓΡΑΜΜΑ WEB DESIGN

Τίτλοι συνεδριών:

Στόχοι προγράμματος	1-2
HTML fundamentals	2-6
CSS fundamentals	6-8
Frontend Design	8
Mobile First Responsive Design ...	9
Frameworks και Libraries.....	10

.frontendDESIGN {...}

/* επισκόπηση και ενημέρωση της μαθησιακής πορείας */

στόχοι_της_συνεδρίας {}

επισκόπηση του προηγούμενου εργαστηρίου: 15min

ενημέρωση της μαθησιακής πορείας: 15min

q&a: 15min

--

πρόγραμμα και μέθοδοι {}

εισαγωγή της μεθόδου του προγράμματος και ερώτηση των συμμετεχόντων για τις προσδοκίες τους: 15min

--

Απαραίτητα υλικά και Απαιτούμενος χώρος

Φορητός υπολογιστής, γρήγορη πρόσβαση στο Διαδίκτυο, βιντεοπροβολέας

--



Συνιστώμενη ανάγνωση / Πόροι

Η καλύτερη πηγή για μάθηση:

<https://developer.mozilla.org/en-US/>

!σημαντικό (χρησιμοποιήστε το για να βρείτε απαντήσεις):

<https://stackoverflow.com/>

ΒΑΣΙΚΑ:

<https://css-tricks.com/video-screencasts/58-html-css-the-very-basics/>

CSS Reference:

<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

CSS Selectors:

<https://css-tricks.com/how-css-selectors-work/>

Σχετικά:

[How To Use Media Queries For Device Targeting - Vanseo Design](https://1stwebdesigner.com/css3-in-a-nutshell/)

<https://1stwebdesigner.com/css3-in-a-nutshell/>

<https://code.tutsplus.com/tutorials>

<https://tympanus.net/codrops/>

<https://www.awwwards.com/>
<https://24ways.org/>
<https://medium.freecodecamp.org/css-naming-conventions-that-will-save-you-hours-of-debugging-35cea737d849>
<https://www.sarasoueidan.com/blog/css-shapes/>
<https://medium.com/coding-artist/a-beginners-guide-to-pure-css-images-ef9a5d069dd2>
<https://hacks.mozilla.org/2014/09/single-div-drawings-with-css/>
--
https://docs.google.com/document/d/1PdmZUY6-vro9UrxUZxEwM7_5wYNXzisPeB8FPBVkq8s/edit?usp=sharing

--
Έμπνευση:
<https://css-tricks.com/>
<https://codepen.io/>
<http://html5doctor.com/>
<https://www.smashingmagazine.com/>
<https://tympanus.net/codrops/>

Εισαγωγή στο World Wide Web

--

Πληκτρολογήστε τη λέξη "load" και πατήστε το πλήκτρο <enter>!
Ο υπολογιστής θα ανταποκριθεί

...

Coldcut - Let Us Play! - Return to margin

[πώς λειτουργεί αυτό](#)

HTMLfundamentals

- | | |
|--|-----------------------------|
| ☐ syntax | (σύνταξη), |
| ☐ elements | (στοιχεία), |
| ☐ linking | (σύνδεσμοι), |
| ☐ using id and class attr | (χρήση id και class), |
| ☐ Styling text | (στυλ κειμένου), |
| ☐ html forms | (html φόρμες), |
| ☐ Tables | (πίνακες), |
| ☐ images and objects | (εικόνες και αντικείμενα).. |



Η ιστορία του front-end development δεν είναι σύντομη καθώς η ιστορία των υπολογιστών προχωράει, αλλά μόλις πρόσφατα άρχισε να παίρνει τα εργαλεία και την προσοχή που της αξίζει. Η σήμανση γραφής, ως επί το πλείστον αγνοήθηκε επειδή ήταν τόσο “δυσκίνητη” όσο και απλοϊκή. Η πρακτική της δημιουργίας του “the view layer” είναι πιο ευχάριστη και συναρπαστική τώρα, περισσότερο από ποτέ και μας δίνει καλύτερες εμπειρίες χρήσης.

Παραδοσιακά, το μεγαλύτερο μέρος διάταξης (layout) σε HTML έχει γίνει μόνο με λίγες ετικέτες:

- `<table>` : Χρησιμοποιόταν λανθασμένα για χρόνια για να ξεπεράσει τις αδυναμίες των διατάξεων CSS. Το αντίκτυπο εναντίον τους, οδήγησε ορισμένους σχεδιαστές να σταματήσουν να τις χρησιμοποιούν εντελώς. Τα δεδομένα πίνακα ανήκουν σε πίνακες. Μην φοβάστε να τα χρησιμοποιήσετε.
- `<div>` : Αυτό είναι το “catch-all” element και χρησιμοποιείται ως προεπιλογή όταν δεν το κάνουν άλλες ετικέτες.
- `<span>` : Παρόμοιο με το div, αλλά αυτό είναι ένα inline-level element χρήσιμο για κείμενο.

Όπως φαντάζεστε, αυτά τα λίγα στοιχεία δεν μας δίνουν μεγάλη διαφοροποίηση και δεν αντικατοπτρίζουν τη δομή της εφαρμογής και τον σκοπό των τμημάτων τους. Κάνουν τη δουλειά που πρέπει, αλλά θα μπορούσε να είναι και καλύτερα.

HTML5

Το 2014, η HTML5 μας έφερε μια επιρροή νέων ετικετών που συμβάλλουν στη διευκόλυνση της διαφοροποίησης του περιεχομένου και στη δημιουργία σημαντικών σχέσεων μεταξύ των στοιχείων.

Σε γενικές γραμμές υπάρχουν [νέα στοιχεία](#), αλλά αυτά είναι τα στοιχεία για διάταξη και είναι τα πιο συνηθισμένα που θα συναντήσετε.

- `<main>` : Περιέχει το κυρίως περιεχόμενο. Αυτή η ετικέτα συνήθως δεν χρησιμοποιείται από τους σχεδιαστές, αλλά είναι ιδιαίτερα χρήσιμη για “μονής-σελίδας” εφαρμογές που αντικαθιστούν το περιεχόμενο.
- `<article>` : Καθορίζει αυτοαποτελούμενο περιεχόμενο, όπως π.χ. άρθρα.
- `<section>` : Κομμάτια περιεχομένου όπως κεφάλαια ή ακόμη και μέρη περιεχομένου που χωρίζονται οπτικά.
- `<header>` : Εισαγωγικό περιεχόμενο. Μπορεί να είναι ιστός με λογότυπο και συνδέσμους πλοήγησης ή ακόμη και να χρησιμοποιείται σε ένα μικρό κομμάτι περιεχομένου.
- `<footer>` : Περιέχει πληροφορίες σχετικά με τα αποτελούμενα στοιχεία, όπως μεταδεδομένα (metadata) για ένα blog post ή ως footer της σελίδας.
- `<nav>` : Περιέχει συνδέσμους πλοήγησης.
- `<aside>` : Περιεχόμενο που συσχετίζεται με το κυρίως περιεχόμενο, αλλά βρίσκεται σε μια περιοχή όπως μία πλαϊνή μπάρα ή προτεινόμενο περιεχόμενο. Η πλευρική πλοήγηση θα έπρεπε αντ’ αυτού να χρησιμοποιεί το nav element.
- `<figure>` και `<figcaption>` : Περιέχει πολυμέσα όπως εικονογραφήσεις και φωτογραφίες. Η ετικέτα figcaption είναι ένθετη μέσα σε σχήμα για να δώσει επιπλέον πληροφορίες.

Αυτές οι ετικέτες προορίζονται να έχουν νόημα αλλά και ευελιξία στην χρήση τους. Μπορεί να παρατηρήσετε ότι οι ορισμοί αυτών των στοιχείων είναι κάπως ασαφείς, και αυτό είναι σκόπιμο. Δεν έχουν κάποιο δικό τους στυλ και μπορούν να χρησιμοποιηθούν σε οποιονδήποτε συνδυασμό που να έχει νόημα για εσάς. Αυτά τα στοιχεία έχουν εξαιρετική υποστήριξη από τους περιηγητές.

Σημασιολογική σήμανση (Semantic Markup)

Πώς φαίνεται στην πράξη. Ας δούμε μερικά παραδείγματα.

Ένα άρθρο blog μπορεί να χρησιμοποιεί ένα header και ένα footer για να διαχωρίσει τον τίτλο και τις meta-πληροφορίες από την ίδια την ανάρτηση.

```
<main class="content">
  <article class="post">
    <header>
      <h2 class="title">Article Title</h2>
      <time datetime="2015-01-30 13:00">Jan 30th</time>
    </header>
    <section>
      <h3>Section Title</h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
    </section>
    <section>
      <h3>Section Title</h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
    </section>
    <footer>
      <span class="post-comments" data-show="#comments">24 comments</span>
    </footer>
  </article>
</main>
```

Ένα header μπορεί να παρέχει σαφείς συνδέσμους πλοήγησης σε μια ετικέτα nav.

```
<header class="site-header">
  <a href="/" class="logo">Logo</a>
  <nav>
    <ul>
      <li><a href="/register">Register</a> </li>
      <li><a href="/login">Log In</a> </li>
    </ul>
  </nav>
</header>
```

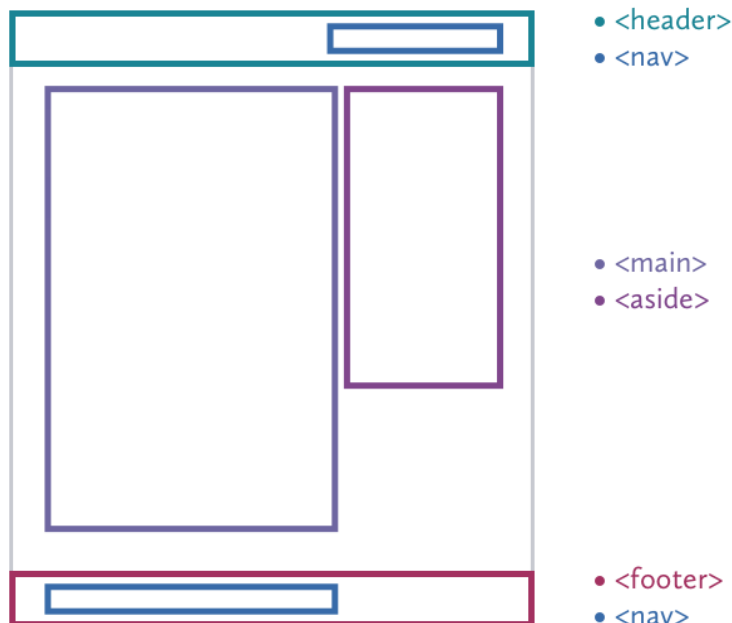
Αυτά τα στοιχεία μπορούν να ομαδοποιηθούν. Αυτό είναι ιδιαίτερα βοηθητικό για section tags.

```
<section>
  <h1>Pangolins</h1>
  <section>
    <h2>Habitat</h2>
    <p>Pangolins live in two regions: Southern Africa and South-East Asia</p>
  </section>
  <section>
    <h2>Diet</h2>
    <p>They are insectivorous, mainly eating ants and termites.</p>
  </section>
  <section>
    <h2>Conservation</h2>
    <p>Pangolins are endangered and protected under international laws.</p>
  </section>
</section>
```

Τα Headers και footers μπορούν να χρησιμοποιηθούν ως γενικά στοιχεία και επίσης μέσα σε άλλα στοιχεία.

```
<header class="site-header">
  <a href="/" class="logo">Logo</a>
</header>
<main class="content">
  <article>
    <header>
      <h1>The Robot Uprising and You</h1>
    </header>
    <section>
      <p> ... </p>
    </section>
    <footer>
      Posted by Ralph Bot on August 29th
    </footer>
  </article>
</main>
<footer class="site-footer">
  &copy; 2015 Businesscorp
</footer>
```

A basic HTML5 layout



Inline και block-level στοιχεία

Θα πρέπει να αφιερώσουμε λίγο χρόνο για να μιλήσουμε για τη [διαφορά](#) μεταξύ block-level και inline-level στοιχείων. Τα Inline-level στοιχεία χρησιμοποιούνται για κείμενο και δεν προσθέτουν (από προεπιλογή) αλλαγή γραμμής (line-break) πριν και μετά. Αυτά είναι tags όπως `<a>`, `<strong>` και `<img>`. Τα Block-level στοιχεία είναι τα πιο συνηθισμένα tags όπως `<div>`, `<header>` και `<h1>`. Το καθένα μπορεί να έχει το styling του άλλου χρησιμοποιώντας την ιδιότητα εμφάνισης στα styles σας.

```
a { display: block; }
```

Υπάρχει επίσης και ο συνδυασμός μεταξύ των δύο παραπάνω, που ονομάζεται inline-block. Αυτό κάνει το στοιχείο να παραμείνει inline, αλλά να αποδεχτεί παραμέτρους όπως `top` και `bottom padding`. Σε ορισμένες περιπτώσεις, αυτή είναι μια απλούστερη προσέγγιση από τη χρήση `floats` και μπορεί να είναι πολύ χρήσιμη όταν πρόκειται για ευθυγράμμιση στοιχείων με κείμενο.

```
img { display: inline-block; vertical-align: middle; }
```

Η σημασία του REL attribute στο `<a>` tag!

Χρησιμοποιούνται για την παροχή πληροφοριών στις μηχανές αναζήτησης σχετικά με τη δομή της σελίδας σας. Για παράδειγμα, μπορείτε να δώσετε στους συνδέσμους ένα REL attribute μετά ή ένα REL attribute πριν για σελιδοποίηση των συνδέσμων (σύνδεσμοι που δείχνουν την επόμενη/τελευταία σειρά αποτελεσμάτων αναζήτησης), ή θα μπορούσατε να δώσετε μία κανονική τιμή για να "πείτε" στις μηχανές αναζήτησης ποιος είναι ο προεπιλεγμένος σύνδεσμος της σελίδας που βλέπει (μερικές φορές οι σελίδες είναι προσβάσιμες μέσω πολλών διαφορετικών συνδέσμων και αυτό αποτρέπει το indexing διπλού περιεχομένου που θα μπορούσε να βλάψει το SEO σας).

Ακολουθούν μερικοί σύνδεσμοι από την Google σχετικά με την καλύτερη χρήση:

<https://support.google.com/webmasters/answer/1663744> <https://support.google.com/webmasters/answer/139394> https://www.w3schools.com/TAGs/att_a_rel.asp

Βασικές αρχές CSS

Cascading Stylesheets — ή CSS — είναι η πρώτη τεχνολογία που πρέπει να αρχίσετε να μαθαίνετε μετά την HTML. Ενώ η HTML χρησιμοποιείται για να καθορίσει τη δομή και τη σημασιολογία του περιεχομένου σας, η CSS χρησιμοποιείται για το στυλ και την εμφάνιση. Για παράδειγμα, μπορείτε να χρησιμοποιήσετε την CSS για να αλλάξετε τη γραμματοσειρά, το χρώμα, το μέγεθος και τα κενά διαστήματα του περιεχομένου σας, να το χωρίσετε σε πολλές στήλες ή να προσθέσετε κινούμενα σχέδια και άλλα διακοσμητικά χαρακτηριστικά.

- | | |
|------------------------|-------------------|
| □ Syntax | (σύνταξη) |
| □ Positional selectors | (επιλογείς θέσης) |
| □ Box layout | (διάταξη κουτιού) |
| □ Styling Text | (στυλ σε κείμενο) |
| □ Styling Forms | (στυλ σε φόρμες) |
| □ CSS Layouts | (CSS διατάξεις) |



Διάταξη CSS

<https://learnlayout.com/no-layout.html>

https://www.w3schools.com/css/css_website_layout.asp

<https://www.smashingmagazine.com/2018/05/guide-css-layout/>

Δεν προσπάθησα να τεκμηριώσω πλήρως κάθε μέθοδο διάταξης, καθώς αυτό θα είχε δημιουργήσει ένα βιβλίο και όχι ένα άρθρο. Αντ' αυτού, δίνω μια επισκόπηση του τι είναι διαθέσιμο σε εσάς, με πολλούς συνδέσμους για να μάθετε περισσότερα.

Normal Flow (Κανονική Ροή)

Αν πάρετε μια σελίδα HTML που δεν εφαρμόστηκε CSS για να αλλάξει η διάταξη, τα στοιχεία θα εμφανίζονται σε normal flow. Στο normal flow, τα κουτιά εμφανίζονται το ένα μετά το άλλο με βάση τον τρόπο γραφής του εγγράφου. Αυτό σημαίνει ότι εάν έχετε οριζόντιο τρόπο γραφής, στον οποίο οι προτάσεις είναι από αριστερά προς δεξιά ή δεξιά προς αριστερά, το normal flow θα εμφανίζει τα κουτιά των block level elements το ένα μετά το άλλο, κάθετα στη σελίδα.

Το Normal flow είναι εκεί που ξεκινάτε με οποιαδήποτε διάταξη: όταν δημιουργείτε μια διάταξη CSS, παίρνετε τα μπλοκ και τα αναγκάζετε να κάνουν κάτι διαφορετικό από την κανονική ροή (normal flow).

ΔΟΜΗΣΤΕ ΤΟ ΑΡΧΕΙΟ ΣΑΣ ΓΙΑ ΝΑ ΠΑΡΕΤΕ ΤΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΟΥ NORMAL FLOW

Μπορείτε να εκμεταλλευτείτε το normal flow, διασφαλίζοντας ότι το αρχείο σας ξεκινά με έναν καλά δομημένο τρόπο.

Φανταστείτε εάν - αντί γι αυτήν την ιδέα του normal flow - το πρόγραμμα περιήγησης συσσωρεύει όλα τα κουτιά σας στη γωνία το ένα πάνω στο άλλο μέχρι να δημιουργήσετε μια διάταξη.

Εάν το CSS σας αποτυγχάνει να φορτωθεί, ο χρήστης μπορεί ακόμα να διαβάσει το περιεχόμενο, και οι χρήστες που δεν λαμβάνουν καθόλου CSS (π.χ. κάποιος που χρησιμοποιεί πρόγραμμα ανάγνωσης οθόνης) θα τους παραδωθεί το περιεχόμενο με τη σειρά που υπάρχει στο αρχείο. Αυτό το καθιστά σημαντικό από την άποψη της προσβασιμότητας ότι το αρχείο HTML σας ξεκινά με σωστή σειρά. Ωστόσο, θα κάνει τη ζωή σας πιο εύκολη ως προγραμματιστής, εάν το περιεχόμενό σας είναι με τη σειρά που ένας χρήστης θα περίμενε να το διαβάσει, ώστε να μην χρειαστεί να κάνετε μαζικές αλλαγές στη διάταξη για να το φτάσετε στο σωστό σημείο.

Με τις νέες μεθόδους θα εκπλαγείτε για το πόσο λίγα χρειάζεται να κάνετε.

Επομένως, πριν σκεφτείτε τη διάταξη, σκεφτείτε τη δομή του αρχείου και τη σειρά με την οποία θέλετε να φαίνεται το περιεχόμενο σας από την κορυφή του αρχείου έως το τέλος.

Box Model (Μοντέλο Κουτιού)

https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/The_box_model
https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Box_Model/Introduction_to_the_CSS_box_model

Το box model είναι μία από αυτές τις έννοιες στο CSS που μπερδεύουν πολλούς νέους front-end developers. Πολλές από τις εφαρμογές του ήταν ασαφείς για εμένα. Αυτό το άρθρο λοιπόν είναι μια προσπάθεια να κατανοήσουμε περισσότερο αυτή την έννοια και επίσης να βοηθήσουμε τους άλλους να έχουν καλύτερη ιδέα για το μοντέλο αυτό.

Τι είναι το Box Model?

Το CSS box model είναι ένα σύνολο από κανόνες που καθορίζουν τον τρόπο απόδοσης κάθε στοιχείου σε μια ιστοσελίδα. Σύμφωνα με το Box model, κάθε στοιχείο σε μια σελίδα είναι ένα ορθογώνιο κουτί με τουλάχιστον περιεχόμενο, πλάτος και ύψος.

Σύμφωνα με το CSS box model, κάθε στοιχείο μέσα σε μια ιστοσελίδα είναι ένα ορθογώνιο κουτί. Το περιεχόμενο βρίσκεται στη μέση και περιβάλλεται από προαιρετικά στοιχεία όπως padding, border και margin. Το box model έχει πέντε κύρια μέρη / ιδιότητες που καθορίζουν το μέγεθος του κουτιού. Οι ιδιότητες αυτές είναι width (πλάτος), height (ύψος), padding (διάστημα), border (περίγραμμα) και margin (περιθώριο).

Εκτός από το width και το height οι υπόλοιπες είναι προαιρετικές. Αυτό σημαίνει ότι μπορούμε να έχουμε ένα κουτί χωρίς padding, χωρίς border και χωρίς margin. Στην επόμενη ενότητα, θα αναλύσουμε λεπτομερώς τα διάφορα μέρη αυτού του κουτιού.

Ένα κουτί έχει γενικά πέντε κύρια μέρη συμπεριλαμβανομένων των width, height, padding, border και margin. Τα Width και height είναι απαραίτητα, είτε τα καθορίζουμε, είτε όχι. Τα Padding, border και margin είναι προαιρετικά.

Content, Width και Height (Περιεχόμενο, Πλάτος και Ύψος)

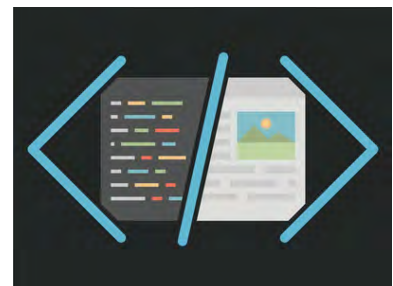
Σε κάθε κουτί, υπάρχει περιεχόμενο στον πυρήνα του. Το περιεχόμενο μπορεί να είναι κείμενο, εικόνες, βίντεο, ήχος κ.λπ. Τα width, height, padding, border, και margin εφαρμόζονται στο περιεχόμενο. Κάθε στοιχείο html έχει προεπιλεγμένο width βάσει δύο παραγόντων: αν το στοιχείο είναι inline ή αν είναι block. Το height σπάνια ορίζεται για στοιχεία. Γενικά, το height του περιεχομένου ορίζεται ως προεπιλεγμένο.

Padding

Το Padding είναι το διάστημα που περιβάλλει το περιεχόμενο, χωρίζοντάς το από το περίγραμμα. Παίρνει το χρώμα από το φόντο του περιεχομένου. Η ιδιότητα padding CSS είναι η συντομογραφία του padding-top, padding-right, padding-bottom και padding-left. Καθορίζει το μέγεθος του διαστήματος μεταξύ του περιεχομένου και του περιγράμματος ή το margin εάν το περίγραμμα δεν έχει οριστεί, π.χ. none existent. Μπορούμε να εφαρμόσουμε padding σε οποιοδήποτε στοιχείο HTML, όμως ποτέ δεν μπορεί να έχει αρνητική αξία.

Frontend Design

- *Analysis of requirements and design files*
- *Setting project*
- *Planning markup*
- *Adopting coding guides*
- *Creating style guides (usage of design systems)*
- *Developing html/css code*



Αρχές

Responsive (Απόκριση σε οθόνες)

Όλα πρέπει να είναι 100% responsive. Η σελίδα σας θα πρέπει να λειτουργεί σωστά, ανεξάρτητα από τη συσκευή ή το μέγεθος της οθόνης του χρήστη.

Readable (Αναγνώσιμο)

Ανεξάρτητα από το φωτισμό ή τη συσκευή, τα μεγέθη των γραμματοσειρών πρέπει να είναι αρκετά μεγάλα και η αντίθεση να είναι αρκετά υψηλή ώστε οι χρήστες σας να διαβάζουν εύκολα το περιεχόμενό σας.

Modular (Με ενότητες)

Προσπαθήστε να μην φτιάξετε ένα μονοκόμματο πλαίσιο, δημιουργήστε μία συλλογή από μικρές ενότητες, που μπορούν να αναμιχθούν και να συνδυαστούν ή να χρησιμοποιηθούν ανεξάρτητα. Χρησιμοποιήστε μόνο ό,τι χρειάζεστε, αφήστε τα υπόλοιπα.

Accessible (προσβάσιμο)

Η προσβασιμότητα είναι πολύ σημαντική. Τεκμηριώστε την εργασία σας, για παράδειγμα, όταν δημιουργείτε μία CSS σύνθεση, γράψτε σχόλια σε τι χρησιμεύει ή γιατί το κάνατε με αυτόν τον τρόπο, σε περίπτωση που χρειαστεί να επιδιορθωθεί κάποιο σφάλμα.

Performant (Αποδοτικό)

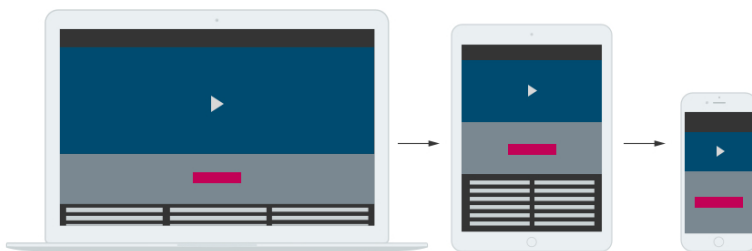
Ο κώδικας πρέπει να βελτιστοποιηθεί για καλύτερη απόδοση. Να είστε προσεκτικοί με το cascading και την κληρονομικότητα (inheriting styles). Αφαιρέστε τα σκουπίδια.

Reusable (Επαναχρησιμοποιήσιμο)

Η σαφής τεκμηρίωση συμβάλλει στη δημιουργία μιας κοινής κατανόησης των προτύπων σχεδιασμού μεταξύ της ομάδας σας. Αυτό βοηθά στην προώθηση της επαναχρησιμοποίησης και μειώνει την ποσότητα πλεονασμού σε μια βάση κώδικα.

Mobile First Responsive Design

Responsive Design



Διαδικασία σχεδιασμού στην Responsive εποχή

Δεν μπορείτε να σχεδιάσετε μία responsive, content-focused, mobile-first σελίδα με τον ίδιο τρόπο που δημιουργούσατε σελίδες εδώ και χρόνια - απλά δεν μπορείτε. Εάν οι στόχοι σας είναι να παράξετε κάτι που είναι μη συγκεκριμένου πλάτους και εξυπηρετεί μικρότερες συσκευές με τα απαιτούμενα styles, γιατί να χρησιμοποιήσετε μια απαρχαιομένη διαδικασία που έρχεται σε αντίθεση με αυτούς τους στόχους?

Θα ήθελα να σας παρουσιάσω ορισμένα προβλήματα που προκαλούνται από τη χρήση παλαιών διαδικασιών στον responsive σχεδιασμό. Ας δούμε μια εξελισσόμενη διαδικασία σχεδιασμού που χρησιμοποιούμε με μερικά πολλά υποσχόμενα deliverables και εργαλεία. Αυτό θα πρέπει να αποτελέσει αφετηρία για να ανανεώσετε τη δική σας διαδικασία και να την φέρετε στην εποχή responsive.

Το πρόβλημα

Τα προβλήματα που προκύπτουν στην προσπάθεια να εξάγεις νέα αποτελέσματα από μία παλιά διαδικασία είναι σημαντικά, παράξενα και όχι άμεσα προφανή. Όλοι μας τα έχουμε συνηθίσει, όπως την ενοχλητική ιδιοτροπία που δεν συνειδητοποιήσαμε ότι είχαμε, έως ότου κάποιος να το επισημάνει. Και από εκείνο το σημείο και μετά, μας κάνει να τρελαινόμαστε.

Για παράδειγμα, όταν δημιουργούμε μία desktop-sized, fixed-width διάταξη σελίδας στο Photoshop και το παραδίδουμε σε έναν προγραμματιστή για να το μεταφράσει σε HTML/CSS, ζητάμε από τον προγραμματιστή να πάρει αρκετές αποφάσεις για τον σχεδιασμό—πιθανότατα χωρίς να το αντιληφθούμε. Παρακάτω είναι ένα μικρό δείγμα:

- Πώς πρέπει να προσαρμόζεται η διάταξη για συσκευές μικρότερου μεγέθους; (Σίγουρα θα ήταν ωραίο να έχουμε μια ιεραρχία σημαντικών στοιχείων σελίδας με βάση τον σκοπό τους)
- Ποια είναι η ιεραρχία του περιεχομένου; (Όλο αυτό το “Lorem Ipsum” δεν το καθιστά προφανές;)
- Πώς ανταποκρίνεται η πλοήγηση σε μικρότερες οθόνες; (Πώς μπορώ να χειριστώ δέκα συνδέσμους με πέντε υποσελίδες που κάθε μία αποκαλύπτεται στο hover με συσκευή αφής 320 × 480;!)

Αυτό μπορεί να προκαλέσει σημαντικά προβλήματα εάν ο προγραμματιστής δεν αισθάνεται σίγουρος για το οπτικό σκηνικό. Ακόμη και σχεδιαστές / προγραμματιστές που αισθάνονται άνετα να πάρουν αυτές τις αποφάσεις μπορεί να βρεθούν σε δύσκολη θέση. Στο τέλος, ο προγραμματιστής αναγκάζεται συχνά να κάνει υποθέσεις όπου τα σχέδια δεν ήταν σαφή εκ των προτέρων - μερικές φορές, ημέρες πριν το feedback από τον σχεδιαστή ή τον πελάτη. Μερικές φορές λειτουργεί αλλά μερικές όχι.

Frameworks και Libraries -- επισκόπηση

--

Bootstrap, foundation, bulma.. Υπάρχουν πολλά παρόμοια frameworks, εναπόκειται σε εσάς και την ομάδα σας να αποφασίσετε ποιο θα χρησιμοποιήσετε.

Το πιο δημοφιλές και με τη μεγαλύτερη βάση κώδικα είναι το Bootstrap. Το foundation έχει επίσης πολλά ενδιαφέρον πράγματα αλλά έχει μικρότερη κοινότητα. Το Bulma είναι απλό, ταχέως αναπτυσσόμενο και με καλογραμμένο κώδικα.

Είναι πιο εύκολο να χρησιμοποιήσετε μερικές βιβλιοθήκες και εξοικονομείτε και χρόνο που μπορεί να είναι πιο πολύτιμος σε άλλα βήματα της διαδικασίας ή σε κρίσιμες καταστάσεις.

Εργασίες

Εργασία - Μέρος 1

Προετοιμάστε τα προσωπικά σας δεδομένα σελίδας (Site Data) και οδηγίες (Guidelines):

Βήμα 1:

Επεξήγηση απαιτήσεων, ποιος είναι ο σκοπός της σελίδας κλπ....
Προετοιμάστε υλικό (γραφικά, κείμενα) και ταξινομήστε το σωστά σε φάκελο (ελέγξτε τις σημειώσεις για οδηγίες).

Βήμα 2:

Δημιουργείτε Οδηγό Περιεχομένου: λίστα σελίδων - site tree (μπορεί να είναι μονής σελίδας), διάγραμμα ροής περιεχομένου εάν υπάρχουν περισσότερες από μία σελίδες.

Βήμα 3:

Δημιουργείτε Οδηγό Στυλ (μία html σελίδα, ένα css αρχείο, όχι js).
Αυτή η σελίδα πρέπει να περιέχει ξεχωριστά μπλοκ για:
Typography (define headings, text rules, create),
content dividers (decorational elements),
Form elements styling (text inputs, textarea, radio buttons, checkboxes),
Button designs (minimum 2),
blockquotes,
inline elements styling.

Τέλος

Σημείωση:

Κάθε βήμα πρέπει να εφαρμοστεί ξεχωριστά στο Google Drive.

Εκτίμηση: 5 ημέρες.

ΤΟ ΕΡΓΟ ΣΥΓΧΡΗΜΑΤΟΔΕΙΤΑΙ ΑΠΟ:



Erasmus+



Co-funded by the
Erasmus+ Programme
of the European Union

Project implemented by



RES POLIS

Centar za omladinski
i društveni razvoj



НАЦИОНАЛНА СЛУЖБА
ЗА ЗАПОШЉАВАЊЕ



egina

(European Grants International Academy S.R.L.)



pôle
emploi

K DIGITALE
DIGITAL TRANSFORMATION INSIGHT

eeStec
LC Tirana

DOR3A
EDUCATIONAL INSTITUTE



Συγγραφέας αυτού του εκπαιδευτικού προγράμματος είναι η CYSD RES POLIS. Αυτά τα εκπαιδευτικά προγράμματα είναι δωρεάν και προορίζονται για μη εμπορική χρήση.