
Speed Optimization στο WordPress.

Οδηγός και τεχνικές για να φορτώνει το site μας σε χρόνο dt.

Παρουσίαση στα πλαίσια του **WordCamp Athens 2017**



Howdy!

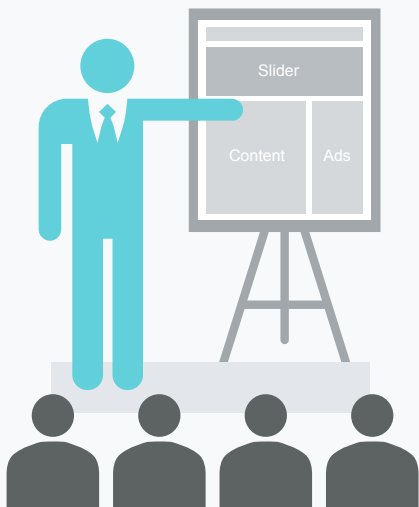


Who Am I;

Ονομάζομαι Αρχιμήδης Μερτζάνος

Εργάζομαι ως **Lead WordPress Engineer** στην
Hellenic Technologies – hta.gr –
mertzanos@hta.gr

Πρόλογος



Σε αυτή την παρουσίαση θα δείτε τρόπους και τεχνικές που θα σας βοηθήσουν να βελτιώσετε την ταχύτητα φόρτωση της WordPress ιστοσελίδα σας. Θα μιλήσουμε για τεχνικές και έννοιες όπως:

- Image Compression
- CSS/JS Minification
- CSS/JS Combinations
- JS Defer Parsing
- Critical CSS
- CDN
- Prefetching / Prerender tags

Πριν ξεκινήσουμε.. Το plugin or not to plugin;

Πολλές από τις έννοιες / τεχνικές που προαναφέρθηκαν μπορούν να υλοποιηθούν και με χρήση WordPress plugin.

Δεν υπάρχει κανένα πρόβλημα με αυτό, εμείς θα δούμε πώς μπορούμε να υλοποιήσουμε διάφορες τεχνικές και να εμβαθύνουμε σε αυτές .

Code Samples & References

<https://pastebin.com/u/ArchimidisM/1/q1DEkwhd>



Εργαλείο μέτρησης αποτελεσμάτων.

Προσωπικά προτείνω το GTmetrix. Υπάρχουν και άλλα όπως το Google PageSpeed, Webpagetest.org , Pingdom.

GTmetrix Features Resources Blog GTmetrix PRO

Welcome Archimidis! | User Settings | Log Out

Features

GTmetrix has a suite of features and options to make optimizing your website clear and easy.

If you're using GTmetrix in a professional capacity and need more monitoring, API credits, or exclusive features, consider [GTmetrix PRO](#).

Get a summary of key performance indicators




Latest Performance Report for: <https://gtmetrix.com/>



ΠΡΟΒΛΗΜΑ ΤΑΧΥΤΗΤΑΣ 1:

Εικόνες μεγάλου μεγέθους και μεγάλων διαστάσεων.

Ίσως το νούμερο 1 πρόβλημα που υπάρχει στα WordPress sites.

-  Ασκοπώς μεγάλα μεγέθη εικόνων για παράδειγμα εικόνες 3MB σε κάποιο slider.
-  Μεγάλο πλήθος εικόνων , για παράδειγμα 15 εικόνες σε ένα carousel.
-  Δεν λαμβάνεται υπόψιν ότι το website θα προβληθεί και σε κινητό.

[Πιθανή λύση](#) >>

Image compression – Συμπίεση εικόνων

Η διαδικασία κωδικοποίησης των δεδομένων μίας εικόνας ή μετατροπής αυτής σε μία που καταλαμβάνει λιγότερο χώρο. Χονδρικά έχει δύο βασικές εκφάνσεις την Lossless και την Lossy compression.

Τρόπος 1: Χειροκίνητα

Συνίσταται για λίγες εικόνες . Μπορείτε να χρησιμοποιήσετε εργαλεία όπως:

- <https://tinypng.com>
- <https://compressjpeg.com>
- <https://compressor.io>
- <https://saerasoft.com/caesium/>

Έπειτα ανεβάζουμε τις εικόνες εκεί που θέλουμε, μέσω του WordPress.

Τρόπος 2: Με χρήση plugin

Μπορείτε να χρησιμοποιήσετε συνδυασμό plugins για να επιτύχετε ακόμα καλύτερο αποτέλεσμα.

<https://wordpress.org/plugins/imagify/>
<https://wordpress.org/plugins/adaptive-images/>

Στην περίπτωση αυτή όχι μόνο κάνουμε συμπίεση εικόνων αλλά στέλνουμε και τις σωστές εικόνες και διαστάσεις στις Mobile συσκευές.

[Αποτελέσματα](#) >>

Image compression –Αποτελέσματα

Χρησιμοποιώντας το Imagify Plugin και μόνο είχαμε τα εξής αποτελέσματα:

ΠΡΙΝ:

Performance Scores



ΜΕΤΑ:

Performance Scores




```
margin: 0;
position: relative;
vertical-align: middle;
width: 1.6rem;
```






```
.avatar.avatar-xs {
font-size: .4rem;
height: .8rem;
width: .8rem;
}
```



```
.avatar.avatar-sm {
font-size: .6rem;
height: 1.2rem;
```

ΠΡΟΒΛΗΜΑ ΤΑΧΥΤΗΤΑΣ 2: Μεγάλου μεγέθους αρχεία CSS/JS

Σημαντικό πρόβλημα που εμφανίζεται σε διάφορα WordPress themes τα οποία «φορτώνουν» και μεγάλα και πολλά CSS/JS αρχεία για να λειτουργήσουν.

-  Αρχεία κώδικα που περιέχουν πολλά και άσκοπα κενά.
-  Μεγάλη ποσότητα σχολίων και formatting στα αρχεία αυτά.
-  Μα γιατί είναι κακό αυτό; Η ύπαρξη σχολίων και το όμορφο formatting , είναι λάθος;

CSS/JS/HTML Minification

Η διαδικασία κατά την οποία αφαιρούμε/αλλάζουμε περιττά στοιχεία από τα CSS/JS/HTML αρχεία μας και έτσι μειώνουμε δραστικά το μέγεθός τους ώστε να "σερβίρονται" πολύ πιο γρήγορα από τον server χωρίς φυσικά να είναι εσφαλμένα.

Τρόποι για να το πετύχουμε (Είτε για CSS, είτε για JS είτε για HTML)

- Χειροκίνητα**
- Με χρήση plugins** όπως το <https://wordpress.org/plugins/autoptimize> η full featured caching plugins
- Με χρήση Node.js και με το Grunt.**

CSS/JS/HTML Minification - Extra Tips 1

- ❑ Στην χειροκίνητη περίπτωση για να βρω τα CSS που γίνονται enqueue : <https://pastebin.com/hJF3dBjM> .
- ❑ Στο αρχείο functions.php χρησιμοποιούμε την *global* μεταβλητή *\$wp_styles*. Το ίδιο και για τα scripts.

Code chunk:

```
global $wp_styles;

foreach( $wp_styles->queue as $style ) :
    $out['styles'][] = $wp_styles->registered[$style]->src . ",";
endforeach;

return $out;
```



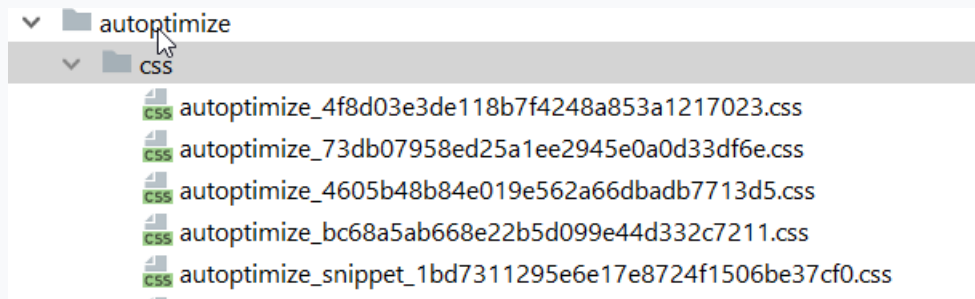
```
[1] => https://mertzanos.gr/wordcamp/wp-content/plugins/js_composer/assets/css/js_composer.min.css;
[2] => https://mertzanos.gr/wordcamp/wp-content/plugins/contact-form-7/includes/css/styles.css;
[3] => https://mertzanos.gr/wordcamp/wp-content/plugins/essential-grid/public/assets/css/settings.css;
...
-----
```

Με μία μικρή αλλαγή κώδικα παίρνω και τα handles

```
[js_composer_front] => https://mertzanos.gr/wordcamp/wp-content/plugins/js_composer/assets/css/js_composer.min.css;
[contact-form-7] => https://mertzanos.gr/wordcamp/wp-content/plugins/contact-form-7/includes/css/styles.css;
```

CSS/JS/HTML Minification - Extra Tips 2

- ❑ Στην περίπτωση με το plugin , δημιουργείται συνήθως κάποιος φάκελος cache και εκεί γίνονται Minified όλα τα αρχεία που επιλέγουμε.



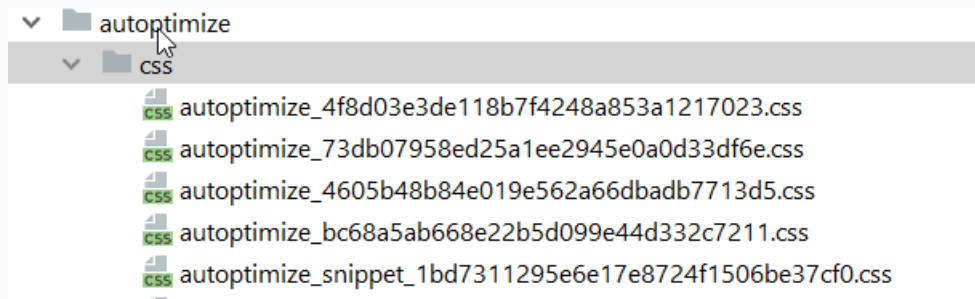
- ❑ Στην περίπτωση Node.js μπορούμε να κατεβάσουμε το <https://www.npmjs.com/package/minifier> και να το τρέξουμε στο local directory μας. Με την παρακάτω εντολή παίρνουμε όλα τα minified .

```
λ minify --clean css
openssl config failed: error:02001003:system library:fopen:No such process
Minification complete
```

- ❑ Χρησιμοποιούμε Grunt Tasks για όλα τα παραπάνω.

CSS/JS/HTML Minification - Extra Tips 2

- ❑ Στην περίπτωση με το plugin , δημιουργείται συνήθως κάποιος φάκελος cache και εκεί γίνονται Minified όλα τα αρχεία που επιλέγουμε.



- ❑ Στην περίπτωση Node.js μπορούμε να κατεβάσουμε το <https://www.npmjs.com/package/minifier> και να το τρέξουμε στο local directory μας. Με την παρακάτω εντολή παίρνουμε όλα τα minified .

```
λ minify --clean css
openssl config failed: error:02001003:system library:fopen:No such process
Minification complete
```

- ❑ Χρησιμοποιούμε Grunt Tasks για όλα τα παραπάνω.

Performance Scores

PageSpeed Score

A (92%) ^

YSlow Score

D (69%) ♦

CSS/JS Combination / Google Fonts Combination

Διαδικασία κατά την οποία όχι ενώνουμε τα διάφορα CSS αρχεία / JS αρχεία σε ένα και μόνο. Αυτό μειώνει κατά πολύ τα HTTP requests.

Τρόποι για να το πετύχουμε (Είτε για CSS, είτε για JS είτε για HTML)

- ❑ Χειροκίνητα
- ❑ Με χρήση plugins όπως το wp-rocket το οποίο λειτουργεί με το autooptimize χωρίς προβλήματα.
- ❑ Με χρήση Node.js και με το Grunt.
- ❑ Προσοχή: Αν τα αρχεία είναι πολύ μεγάλα, ενδέχεται να υπάρξουν τα αντίθετα αποτελέσματα.



CSS/JS/HTML compression/combo errors

Συνήθως εμφανίζονται προβλήματα κατά τη διαδικασία του combination (πολλά αρχεία σε ένα) των JS αρχείων. Πιθανοί λόγοι

- Κακογραμμένος κώδικας χωρίς να υπάρχουν τα σωστά "dependencies" ειδικά σε plugins.
- Reference errors που εμφανίζονται στην κονσόλα μας.

Για να διορθωθούν αυτά συνήθως κάνουμε τα εξής:

- Ανοίγουμε την console με F12 και βλέπουμε το ref error
- Ανοίγουμε νέο tab στον browser με URL της ιστοσελίδας τύπου `http://example.com/?nocache`
- Ψάχνουμε τα αρχεία από το browser για τη μεταβλητή ή τη function που δημιουργεί το error άρα και το URL του αρχείου.
- Το κάνουμε exclude από το minification/combo

Defer JS Parsing

Γίνεται parse του/των συγκεκριμένων JS αρχείων ΑΦΟΥ πρώτα έχει φορτώσει η σελίδα μας. Τα αρχεία αυτά τα δένουμε στο onload event και αφού φορτώσει η σελίδα φορτώνουν και αυτά. Στην ουσία ένα εσωτερικό script καλεί εξωτερικά μετά το load.

- ❑ Προτείνεται για external scripts, πχ google analytics, Facebook scripts κλπ.
- ❑ Δεν προτείνεται για την jQuery αυτή καθ' εαυτή.
- ❑ Θέλει μεγάλη προσοχή στα dependencies για αυτό και προτείνονται εξωτερικά scripts.
- ❑ Συνήθως τα plugins κάνουν τη διαδικασία σωστά.

Κώδικας που μπορείτε να χρησιμοποιήσετε:

<https://pastebin.com/W75v1tPd>

Defer Image Loading - Lazy Loading

Είναι τεχνική φόρτωσης εικόνων κατά τη διάρκεια που ο χρήστης scrolls την σελίδα προς τα κάτω. Μπορεί να κάνει την ταχύτητα φόρτωσης αρκετά μεγάλη όμως έχει διάφορα θέματα, κυρίως performance και διάφορα incompatibilities.



Critical CSS - Τι είναι;



- ✓ Μεταξύ 2 και τέλους του 3 η σελίδα καθυστερεί την πλήρη φόρτωσή της γιατί πρέπει πρώτα να λάβει και να εφαρμόσει τα ζητούμενα CSS. (Render Blocking CSS αλλά και JS φυσικά)
- ✓ Στο διάστημα αυτό ίσως δούμε τη σελίδα για λίγο χωρίς styles.
- ✓ Σε αυτό το διάστημα συμβαίνει το **FOUC** - Flash Of Unstyled Content.

Το Critical CSS είναι το απόλυτα αναγκαίο CSS κομμάτι που πρέπει να φορτώσει πριν φορτώσει πλήρως η σελίδα. Αυτό το κομμάτι είναι το "above the fold" που βλέπετε σαν recommendation στο Google Pagespeed.

Critical CSS - Above the fold content tips

Ένα καλό εργαλείο για να βρούμε το Critical CSS είναι αυτό

<http://www.sitelocity.com/critical-path-css-generator>

Προσοχή, υπάρχει πιθανότητα να πρέπει να αλλάξουμε τα paths των αρχείων σε absolute paths.

Ένα επίσης πολύ καλό εργαλείο που λειτουργεί και με το Autooptimize είναι το <https://criticalcss.com/>

Το CSS που λαμβάνω το τοποθετώ στο ειδικό πεδίο στο Autooptimize ή στο Caching Plugin που χρησιμοποιώ.

DNS Prefetch

Με το Prefetch ο browser μπορεί να «φέρει» resources από ένα URL και να τις αποθηκεύσει στην μνήμη cache που διαθέτει για μελλοντική χρήση

To DNS Prefetch είναι τεχνική βελτιστοποίησης. Μέσω ενός meta tag στην σελίδα μας ο browser «φορτώνει» στο background τα περιεχόμενα κάποιων hosts όπως πχ //fonts.googleapis.com/ κάνοντας DNS Lookups.

```
<link rel="dns-prefetch" href="//fonts.googleapis.com">  
<link rel="dns-prefetch" href="//www.google-analytics.com">  
<link rel="dns-prefetch" href="//cdn.domain.com">
```

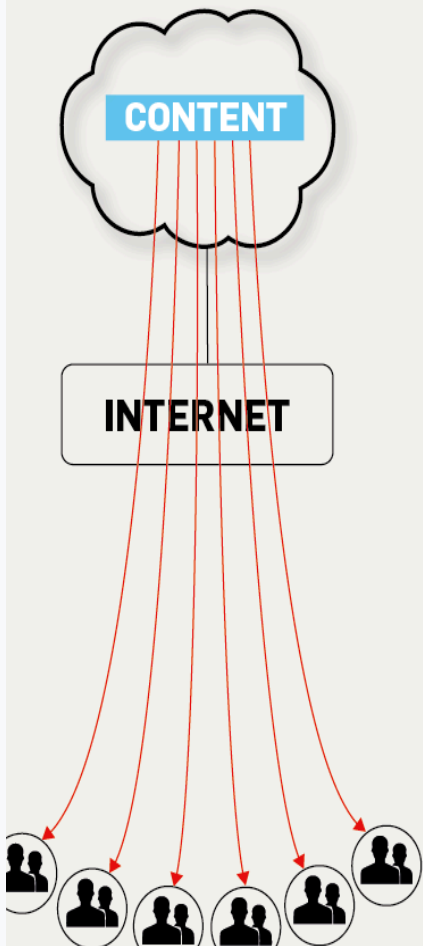
Στο WordPress μπορούμε να χρησιμοποιήσουμε το **wp_resource_hints** filter.

Δεν υποστηρίζεται από τον Opera Mini browser. Οι περισσότεροι browsers το κάνουν αυτόματα.

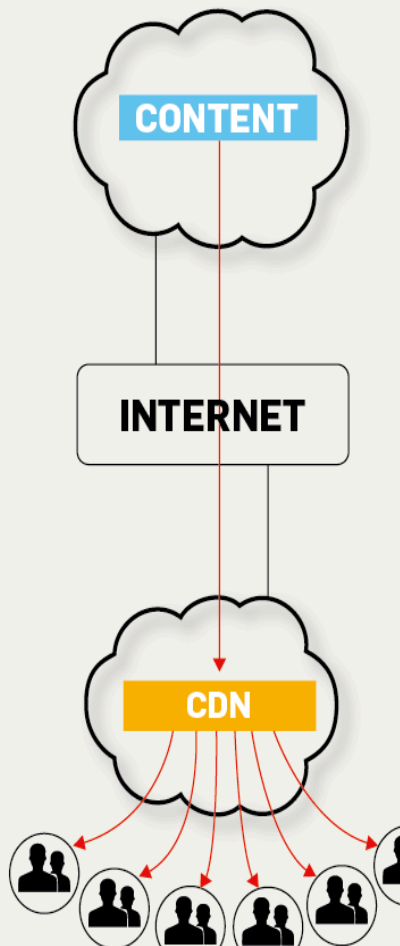
Prerender

1. Το Prerender είναι μία «επέκταση» του DNS prefetch. Όχι μόνο «κατεβάζει» τα δεδομένα του host αλλά τους συμπεριφέρεται σαν να έχουμε ανοίξει ένα δεύτερο tab.

Without CDN



With CDN



CDN – Content Delivery Network

Εξασφαλίζει βέλτιστη διανομή των αρχείων της ιστοσελίδας

Τα αρχεία είναι διαθέσιμα παγκοσμίως

Αντιγράφονται τα αρχεία μας στο server και σερβίρονται όχι πλέον από το δικό μας URL αλλά από το URL του CDN.

Πολλά caching plugins και όχι μόνο υποστηρίζουν τη δυνατότητα CDN.

```
https://wordcamp-9c29.kxcdn.com/wordcamp/wp-content/cache/autoptimize/css/autoptimize_eaf3ffce54f9f4c159d6cb3bcda06932.css
```

Performance Scores

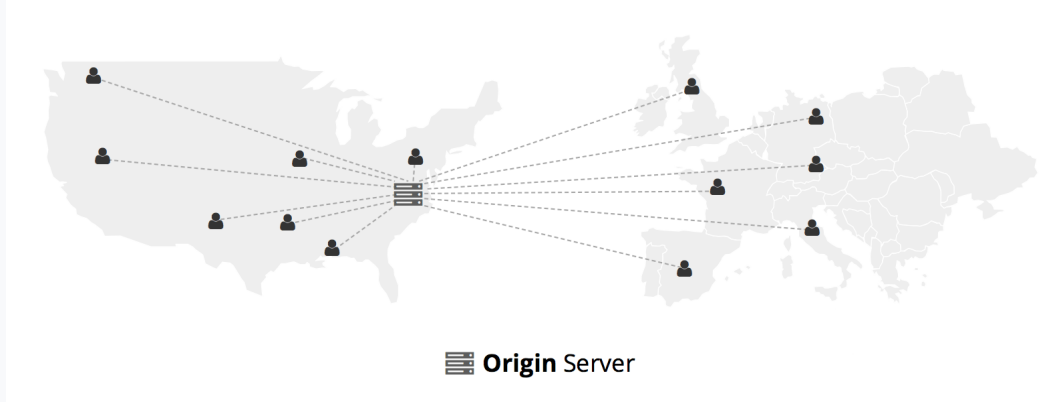
PageSpeed Score

A (94%) ^

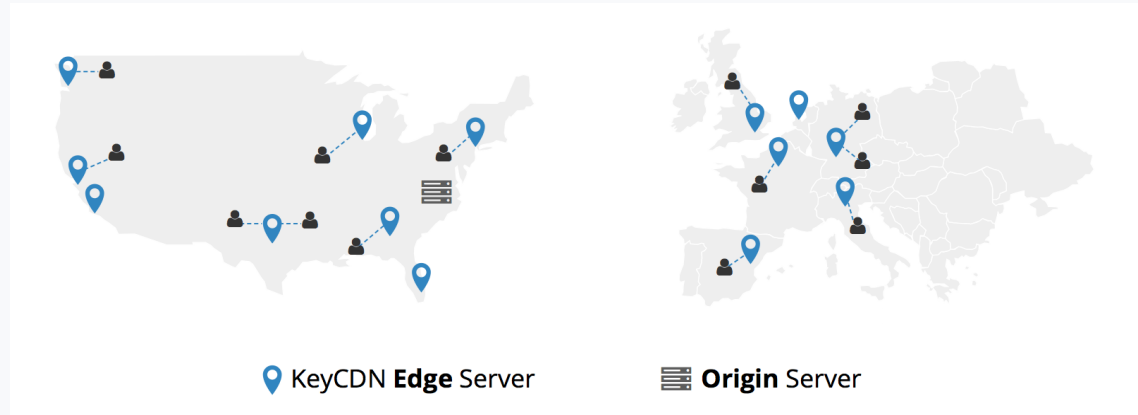
YSlow Score

B (89%) ^

Χωρίς CDN



Με χρήση CDN



Επιπλέον πράγματα που πρέπει να ληφθούν υπόψιν:

- ❑ Ενεργοποίηση Gzip Compression στο server – Πολύ σημαντικό
- ❑ Να μην χρησιμοποιείται @import σε CSS αρχεία. Αυτό γιατί μπορεί να υπάρξει σημαντική καθυστέρηση φόρτωσης.
- ❑ Χρησιμοποίηση Expire Headers. Αυτό ρυθμίζεται είτε από το htaccess , είτε με plugin.
- ❑ Κατά τη φόρτωση της σελίδας να μην υπάρχουν errors, ειδικά «Not found errors».
- ❑ Να χρησιμοποιείτε βέλτιστα image sizes στο WordPress ώστε να σερβίρονται scaled images.
- ❑ Μην κάνετε υπερβολική χρήση WordPress plugins.
- ❑ Σερβίρετε Videos είτε embedded είτε μέσω API και όχι από το server σας.
- ❑ Καλό θα είναι να αφαιρούνται τα query strings από τα resources για παράδειγμα <https://example.com/wp-content/plugins/revslider/public/assets/js/extensions/revolution.extension.layeranimation.min.js?version=5.4.6>
- ❑ Γενικότερα ή βελτιστοποίηση μίας ιστοσελίδας θέλει χρόνο και απαιτεί κόπο. Μην ξεχάσετε αφού την ολοκληρώσετε να κάνετε έλεγχο της ιστοσελίδας ότι λειτουργεί καλά.

ΕΡΩΤΗΣΕΙΣ;

Ευχαριστώ πολύ για την παρακολούθηση.
Μπορείτε να βρείτε αρκετά παραδείγματα κώδικα καθώς και
references εδώ

<https://pastebin.com/u/ArchimidisM/1/q1DEkwhd>

Επίσης μπορείτε να με βρείτε και προσωπικά στο email
mertzanos@hta.gr

THANK YOU