

Τμήμα Μηχανικών Πληροφορικής τ.ε.

Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Δυτικής Ελλάδας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

“ΣΥΓΚΡΙΤΙΚΗ ΑΞΙΟΛΟΓΗΣΗ ΑΠΟΔΟΣΗΣ JAVASCRIPT FRAMEWORKS”

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΩΝ: Χριστίνα Καλαρύτη, ΑΜ:0705

Κατερίνα Παπαδημητρίου, ΑΜ:0816

ΕΠΙΒΛΕΠΩΝ: Σωτήρης Χριστοδούλου

ΑΝΤΙΠΡΟΪΟ 2014

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Αντίρριο, 2014,

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. , Υπογραφή

2. , Υπογραφή

3. , Υπογραφή

Περίληψη

Σκοπός της παρούσας πτυχιακής εργασίας είναι να μελετηθούν και να συγκριθούν οι αποδόσεις των πιο γνωστών Javascript Frameworks σε υπολογιστές και «έξυπνες» κινητές συσκευές, δηλ smartphones αλλά και tablets. Ο λόγος που γίνεται αυτή η αξιολόγηση είναι γιατί πλέον οι περισσότεροι χρήστες διαδικτύου κάνουν χρήση των κινητών τους για να ενημερώνονται για τα πάντα. Λόγω αυτής της ανάπτυξης έχουν αναπτυχθεί αρκετές βιβλιοθήκες java για να μπορούν να υποστηρίξουν τις νέες ανάγκες.

Πιο συγκεκριμένα στο κεφάλαιο 1 θα μελετήσουμε και θα καταγράψουμε τις βασικότερες λειτουργίες που υποστηρίζουν τα 6 πιο γνωστά Javascript Frameworks τα οποία με την σειρά ανάλυσης είναι: JQuery, Dojo, ExtJS, MooTools, Prototype, YUI3. Θα γίνει έλεγχος αν κάποια από τα παραπάνω έχουν mobile εκδόσεις και αναφορά των διαφορών τους σε σχέση με την πλήρη έκδοση.

Στη συνέχεια της εργασίας μέσω ειδικών slickspeed selectors test εκτελέσαμε μετρήσεις οι οποίες αφορούσαν JFrameworks, selectors και browsers. Οι μετρήσεις πραγματοποιήθηκαν από αρκετές συσκευές και οργανώθηκαν σε αρχεία excel ανά Browser και Συσκευή. Δημιουργήσαμε γραφήματα και στο αντίστοιχο κεφάλαιο υπάρχουν τα σχετικά συμπεράσματα.

Όμως ποιος είναι ο λόγος αυτής της εργασίας? Λόγω της ταχείας ανάπτυξης του διαδικτύου και της απλοποίησης των web applications και pages έχει γνωρίσει ιδιαίτερη ανάπτυξη η γλώσσα προγραμματισμού Javascript. Οι προγραμματιστές έχουν υιοθετήσει αρκετές βιβλιοθήκες και πλαίσια εργασίας της Java και αυτό για τη διευκόλυνση του προγραμματισμού των client-side αλληλεπιδράσεων σε εφαρμογές web. Επειδή σε ορισμένες περιπτώσεις, αυτές οι αλληλεπιδράσεις μπορούν να προκαλέσουν μη αμελητέα επιβάρυνση για το χρόνο απόκρισης των εφαρμογών web από κινητές συσκευές, που επηρεάζουν αρνητικά την εμπειρία του χρήστη. Στόχος μας σε αυτήν την εργασία είναι να εξάγουμε στατιστικά συμπεράσματα για κάθε JFramework καθώς και ποιοτικά αποτελέσματα για την λειτουργία και απόδοση κάθε JFs. Όπως επίσης και σε κάθε browser πώς αποδίδει το κάθε JSF.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα θέλαμε να ευχαριστήσουμε θερμά τον καθηγητή μας κ. Σωτήρη Χριστοδούλου, κυρίως για την εμπιστοσύνη που μας έδειξε και την υπομονή που έκανε κατά την διάρκεια υλοποίησης της πτυχιακής εργασίας. Όπως επίσης και για την πολύτιμη βοήθεια και καθοδήγηση του κατά την επίλυση διάφορων θεμάτων.

Θα θέλαμε επίσης να απευθύνουμε τις ευχαριστίες μας και στους γονείς μας, οι οποίοι στήριξαν τις σπουδές μας με διάφορους τρόπους, φροντίζοντας για την καλύτερη δυνατή μόρφωση μας.

Πίνακας περιεχομένων

| | |
|---|-----|
| Περίληψη | iii |
| ΕΥΧΑΡΙΣΤΙΕΣ | iv |
| Κεφάλαιο 1 – Γενικά για την Java | 7 |
| 1.1 Εισαγωγή | 7 |
| 1.2 Java | 9 |
| 1.3 Javascript | 11 |
| 1.4 Διαφορές Java και Javascript | 11 |
| 1.5 Javascript Libraries vs Javascript Frameworks | 12 |
| Κεφάλαιο 2 – Javascripts Frameworks | 13 |
| 2.1 JQuery | 13 |
| 2.1.1 Τι είναι τα jQuery? | 13 |
| 2.1.2 Πλεονεκτήματα της jQuery | 15 |
| 2.1.3 Σύνταξη της jQuery | 16 |
| 2.1.4 jQuery UI | 19 |
| 2.1.5 jQuery Mobile | 19 |
| 2.2 Dojo framework | 21 |
| 2.2.1 Dojo Toolkit | 22 |
| 2.2.2 Dojo Mobile | 22 |
| 2.3 ExtJS | 22 |
| 2.4 MooTools | 24 |
| 2.5 Prototype | 25 |
| 2.6 YUI Javascript | 27 |
| Κεφάλαιο 3 – Αξιολόγηση Απόδοσης JS Framework | 30 |
| 3.1 Μετρήσεις κώδικα | 30 |
| 3.2 Λογισμικά αξιολόγησης και μέτρησης απόδοσης κώδικα | 31 |
| 3.3 Μεθοδολογία μέτρησης απόδοσης JS Framework από κινητές συσκευές | 32 |
| 3.4 Ανάλυση τρόπου λήψης μετρήσεων | 34 |
| Κεφάλαιο 4 – Αποτελέσματα Μετρήσεων | 36 |
| 4.1 Μετρήσεις από Φορητό Υπολογιστή | 36 |
| 4.2 Μετρήσεις από Smartphones | 37 |
| 4.3 Μετρήσεις από Tablet | 40 |
| 4.4 Επιπλέον Αποτελέσματα Έρευνας (1) | 43 |
| 4.4.1 Μετρήσεις από PC/Laptop | 45 |

| | |
|---|----|
| 4.4.2 Μετρήσεις από Smartphone | 46 |
| 4.4.3 Μετρήσεις από Tablet | 47 |
| 4.5 Επιπλέον Αποτελέσματα Έρευνας (2) | 48 |
| 4.5.1. Μετρήσεις από PC/Laptop | 49 |
| 4.5.2 Μετρήσεις από Smartphone | 50 |
| 4.5.3 Μετρήσεις από Tablet | 50 |
| 4.6 Επιπλέον Αποτελέσματα Έρευνας (3) | 51 |
| 4.6.1 Μετρήσεις από PC/Laptop | 52 |
| 4.6.2 Μετρήσεις από Smartphone | 52 |
| 4.6.3 Μετρήσεις από Tablet | 52 |
| Κεφάλαιο 5 – Συμπεράσματα | 54 |
| Βιβλιογραφία..... | 56 |
| Ευρετήριο Εικόνων | 57 |
| Ευρετήριο Πινάκων..... | 57 |
| Ευρετήριο Γραφημάτων..... | 57 |

Κεφάλαιο 1 – Γενικά για την Java

1.1 Εισαγωγή

Η πιο δημοφιλής γλώσσα προγραμματισμού για τα προγράμματα περιήγησης και τις εφαρμογές web είναι σήμερα η Java. Πιο συγκεκριμένα έχει αναπτυχθεί ιδιαίτερα η χρήση των Javascript. Μαζί με την ανάπτυξη των απαιτήσεων για πιο ολοκληρωμένα περιβάλλοντα χρήστη (UI), αυξάνεται το μέγεθος και η πολυπλοκότητα των εφαρμογών διαδικτύου. Από την άλλη πλευρά, η JavaScript γίνεται επίσης μια υπολογιστική πλατφόρμα γενικής χρήσης για προγράμματα περιήγησης, για εφαρμογές γραφείου, για RIAA Frameworks¹ (όπως το Google Web Toolkit, Qooxdoo.org, Cap-puccino.org) ακόμη και περιβάλλοντα ανάπτυξης προγραμμάτων (όπως lively-kernel.org). Τα τελευταία χρόνια το Web γίνεται όλο και πιο προσβάσιμο από φορητές και ασύρματες συσκευές και το 2008 η κινητή πρόσβαση στο Internet είχε υπερβεί την πρόσβαση για πρώτη φορά την πρόσβαση από υπολογιστή. Το διαδίκτυο μέσω κινητού (mobile internet) είναι το όνομα του νέου αυτού τομέα των εφαρμογών διαδικτύου και η JavaScript αναμένεται να παίξει σημαντικό ρόλο στην ανάπτυξή του.

Λόγω της πληθώρας σε εφαρμογές όπου εξυπηρετεί η JavaScript και η ποικιλία των αναγκών προγραμματισμού, τα JavaScript Frameworks (JFs) έχουν αναπτυχθεί προκειμένου να διευκολυνθεί το έργο των διαδικτυακών προγραμματιστών. Ο στόχος αυτών των Frameworks είναι να γίνουν ένα χρήσιμο εργαλείο για την απλούστευση του κώδικα JavaScript Frameworks και επαναχρησιμοποίηση τμημάτων κώδικα γράφοντας μόνο μερικές γραμμές κώδικα. Επιπλέον παρέχουν σαφέστερη δομή, νέα χαρακτηριστικά, υποστήριξη μεταξύ περιηγητών, εφαρμογές και επιπρόσθετα (plugins), έτοιμη για χρήση. Για αυτούς τους λόγους, οι JavaScript βιβλιοθήκες ή/και τα Frameworks, έχουν γίνει πιο δημοφιλή, και κερδίζουν ένα συνεχώς αυξανόμενο αριθμό φίλων σήμερα. Σήμερα, τα πιο δημοφιλή JavaScript Frameworks, είναι τα εξής: ExtJS, Dojo, jQuery, MooTools, Prototype και YUI. Αλλά ποιο JF πρέπει να επιλέξει ένας προγραμματιστής;

¹ Μια πλούσια διαδικτυακή εφαρμογή (RIA) είναι μια διαδικτυακή εφαρμογή που έχει πολλά από τα χαρακτηριστικά των desktop εφαρμογών λογισμικού, συνήθως παραδίδονται μέσω ενός site-και ενός προγράμματος περιήγησης, ενός browser plug-in. Adobe Flash, JavaFX, και το Microsoft Silverlight είναι σήμερα οι τρεις πιο κοινές πλατφόρμες, με τον επιτραπέζιο ποσοστά διείσδυσης του προγράμματος περιήγησης περίπου 96%, 76% και 66%, αντίστοιχα (από τον Αύγουστο του 2011). Πλέον η τάση της Google δείχνει (από τον Σεπτέμβριο του 2012) ότι τα plug-ins που βασίζονται σε Frameworks βρίσκονται σε διαδικασία θα αντικατάστασης από HTML5/Javascript και άλλες εναλλακτικές λύσεις.

Η γλώσσα προγραμματισμού JavaScript χρησιμοποιείται ευρέως για διαδικτυακό προγραμματισμό όλο και περισσότερο. Από την αύξηση της δημοτικότητάς του και την έναρξη της εποχής web 2.0, πολλά JavaScripts Frameworks έχουν γίνει διαθέσιμα για να διευκολυνθεί ο προγραμματισμός των αλληλεπιδράσεων-πελάτη σε εφαρμογές διαδικτύου. Για τους προγραμματιστές web, είναι σημαντικό να επιλέξουν το κατάλληλο Framework που εξυπηρετεί τις τρέχουσες ανάγκες του έργου τους και να φτιάξουν έναν κώδικα υψηλής ποιότητας και καλής απόδοσης. Η εξέλιξη των τεχνολογιών, των προγραμμάτων περιήγησης (web browsers), οι απαιτήσεις των χρηστών για νέες λειτουργίες, αφού τα διάφορα λειτουργικά συστήματα και οι πιο διακριτικές κινητές διαδικτυακές πλατφόρμες, δυσκολεύουν την επιλογή του κατάλληλου framework. Επιπλέον, για τους διαδικτυακούς προγραμματιστές, είναι σημαντικό να επιλέξουν το κατάλληλο Framework που όχι μόνο εξυπηρετεί τις τρέχουσες ανάγκες του έργου, αλλά παρέχει επίσης κώδικα υψηλής ποιότητας και καλή απόδοση. Υπάρχουν πολλοί παράγοντες που μπορούν να επηρεάσουν την επιλογή του κατάλληλου framework, όπως η ενεργή υποστήριξη, η συντήρηση και η ισχύ του πηγαίου κώδικα, οι επιδόσεις σε φορητές συσκευές, κ.λπ. Σκοπός αυτής της πτυχιακής είναι να παρέχει μια λεπτομερή αξιολόγηση της ποιότητας και των επιδόσεων από τα πιο δημοφιλή JavaScript Frameworks, λαμβάνοντας υπόψη ποιοτικούς παράγοντες και δοκιμές επιδόσεων σε καλά εδραιωμένα λογισμικά. Το κύριο αποτέλεσμα αυτής της εργασίας είναι να αναδείξει τα πλεονεκτήματα και τα μειονεκτήματα των JavaScript Frameworks σε διάφορους τομείς ενδιαφέροντος και να προσδιορίσει τα προβληματικά τους σημεία.

Κυρίως, εστιάζουμε στην ποιότητα του κώδικα της βασικής βιβλιοθήκης των JFs και θα διεξάγουμε μια στατιστική ανάλυση λαμβάνοντας υπόψη ποιοτικές μετρήσεις από λογισμικά. Η ποιότητα είναι ένας βασικός παράγοντας που εξετάζουμε στην περίπτωση αυτών των Frameworks, ειδικότερα όπως αυτά χρησιμοποιούνται από μία πληθώρα έργων σήμερα. Για πρώτη φορά αυτά τα JFs μελετώνται και δοκιμάζονται από την ποιοτική άποψη, προσπαθώντας να βρεθεί ποιο από αυτά είναι αρκετά ώριμο για να παράγει μεγάλα έργα. Δεύτερον, μετράμε την απόδοση αυτών των Frameworks και προσπαθούμε να βρούμε βασικούς παράγοντες που μπορεί να τα επηρεάσουν. Τρίτον, διεξάγουμε δοκιμές επικύρωσης και αναζητούμε ευπαθή θέματα ασφαλείας για κάθε Framework.

Το κύριο αποτέλεσμα αυτής της εργασίας είναι να αναδείξει τα πλεονεκτήματα και τα μειονεκτήματα του κάθε JF σε διάφορους τομείς ενδιαφέροντος και να δηλώσουμε που και ποια είναι τα προβληματικά σημεία του κώδικά τους, που κατά πάσα πιθανότητα πρέπει να βελτιωθούν στις επόμενες εκδόσεις.

Πριν ξεκινήσουμε την ανάλυση του κάθε Javascript Framework, θα παραθέσουμε το τι είναι η γλώσσα προγραμματισμού Java και τι η Javascript. Στη συνέχεια θα μελετήσουμε τα χαρακτηριστικά που έχει το κάθε Javascript Framework.

1.2 Java

Στις αρχές του 1991, η Sun αναζητούσε το κατάλληλο εργαλείο για να αποτελέσει την πλατφόρμα ανάπτυξης λογισμικού σε μικρο-συσκευές (έξυπνες οικιακές συσκευές έως πολύπλοκα συστήματα παραγωγής γραφικών). Τα εργαλεία της εποχής ήταν γλώσσες όπως η C++ και η C. Μετά από διάφορους πειραματισμούς προέκυψε το συμπέρασμα ότι οι υπάρχουσες γλώσσες δεν μπορούσαν να καλύψουν τις ανάγκες τους. Ο "πατέρας" της Java, James Gosling, που εργαζόταν εκείνη την εποχή για την Sun, έκανε ήδη πειραματισμούς πάνω στη C++ και είχε παρουσιάσει κατά καιρούς κάποιες πειραματικές γλώσσες (C++ ++) ως πρότυπα για το νέο εργαλείο που αναζητούσαν στην Sun. Τελικά μετά από λίγο καιρό κατέληξαν με μια πρόταση για το επιτελείο της εταιρίας, η οποία ήταν η γλώσσα Oak. Το όνομά της το πήρε από το ομώνυμο δένδρο (βελανιδιά) το οποίο ο Gosling είχε έξω από το γραφείο του και έβλεπε κάθε μέρα.

Η Oak ήταν μία γλώσσα που διατηρούσε μεγάλη συγγένεια με την C++. Παρόλα αυτά είχε πολύ πιο έντονο αντικειμενοστρεφή (object oriented) χαρακτήρα σε σχέση με την C++ και χαρακτηριζόταν για την απλότητα της. Σύντομα οι υπεύθυνοι ανάπτυξης της νέας γλώσσας ανακάλυψαν ότι το όνομα Oak ήταν ήδη κατοχυρωμένο οπότε κατά την διάρκεια μιας εκ των πολλών συναντήσεων σε κάποιο τοπικό καφέ αποφάσισαν να μετονομάσουν το νέο τους δημιούργημα σε Java που εκτός των άλλων ήταν το όνομα της αγαπημένης ποικιλίας καφέ για τους δημιουργούς της. Η επίσημη εμφάνιση της Java στη βιομηχανία της πληροφορικής έγινε το Μάρτιο του 1995 όταν η Sun την ανακοίνωσε στο συνέδριο Sun World 1995. Ο πρώτος μεταγλωττιστής (compiler) της ήταν γραμμένος στη γλώσσα C από τον James Gosling. Το 1994, ο A. Van Hoff ξαναγράφει τον μεταγλωττιστή της γλώσσας σε Java, ενώ το Δεκέμβριο του 1995 πρώτες οι IBM, Borland, Mitsubishi Electronics, Sybase και Symantec ανακοινώνουν σχέδια να χρησιμοποιήσουν τη Java για την δημιουργία λογισμικού. Από εκεί και πέρα η Java ακολουθεί μία ανοδική πορεία και είναι πλέον μία από τις πιο δημοφιλείς γλώσσες στον χώρο της πληροφορικής. Στις **13 Νοεμβρίου του 2006** η Java έγινε πλέον μια **γλώσσα ανοιχτού κώδικα** (GPL) όσον αφορά το μεταγλωττιστή (javac) και το πακέτο ανάπτυξης (JDK, Java Development Kit).

Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε *Java* τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή (Intel x86, IBM, Sun SPARC, Motorola) αλλά και λειτουργικού συστήματος (Windows, Unix, Linux, BSD, MacOS). Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής. Ο συμβολικός κώδικας (*assembly*) που μεταφράζεται και εκτελείται σε Windows είναι διαφορετικός από αυτόν που μεταφράζεται και εκτελείται σε έναν υπολογιστή Macintosh. Η λύση δόθηκε με την ανάπτυξη της *Εικονικής Μηχανής*.

Αφού γραφεί κάποιο πρόγραμμα σε Java, στη συνέχεια μεταγλωττίζεται μέσω του μεταγλωττιστή *javac*, ο οποίος παράγει έναν αριθμό από αρχεία *.class* (κώδικας *byte* ή *bytecode*). Ο κώδικας *byte* είναι η μορφή που παίρνει ο πηγαίος κώδικας της Java όταν μεταγλωττιστεί. Όταν πρόκειται να εκτελεστεί η εφαρμογή σε ένα μηχάνημα, το *Java Virtual Machine* που πρέπει να είναι εγκατεστημένο σε αυτό θα αναλάβει να διαβάσει τα αρχεία *.class*. Στη συνέχεια τα μεταφράζει σε γλώσσα μηχανής που να υποστηρίζεται από το λειτουργικό σύστημα και τον επεξεργαστή, έτσι ώστε να εκτελεστεί (αυτό συμβαίνει με την παραδοσιακή Εικονική Μηχανή (*Virtual Machine*)). Πιο σύγχρονες εφαρμογές της εικονικής Μηχανής μπορούν και μεταγλωττίζουν εκ των προτέρων τμήματα *bytecode* απευθείας σε κώδικα μηχανής (εγγενή κώδικα ή *native code*) με αποτέλεσμα να βελτιώνεται η ταχύτητα. Χωρίς αυτό δε θα ήταν δυνατή η εκτέλεση λογισμικού γραμμένου σε Java.

Ακόμα μία ιδέα που βρίσκεται πίσω από τη *Java* είναι η ύπαρξη του συλλέκτη απορριμμάτων (*Garbage Collector*). Συλλογή απορριμμάτων είναι μία κοινή ονομασία που χρησιμοποιείται στον τομέα της πληροφορικής για να δηλώσει την ελευθέρωση τμημάτων μνήμης από δεδομένα που δε χρειάζονται και δε χρησιμοποιούνται άλλο. Αυτή η απελευθέρωση μνήμης στη Java είναι αυτόματη και γίνεται μέσω του συλλέκτη απορριμμάτων. Υπεύθυνη για αυτό είναι και πάλι η εικονική μηχανή η οποία μόλις «καταλάβει» ότι ο σωρός (*heap*) της μνήμης κοντεύει να γεμίσει ενεργοποιεί το συλλέκτη απορριμμάτων. Έτσι ο προγραμματιστής δε χρειάζεται να ανησυχεί για το πότε και αν θα ελευθερώσει ένα συγκεκριμένο τμήμα της μνήμης, ούτε και για σφάλματα δεικτών. Αυτό είναι ιδιαίτερα σημαντικό γιατί είναι κοινά τα σφάλματα προγραμμάτων που οφείλονται σε λανθασμένο χειρισμό της μνήμης.

Παρόλο που η εικονική μηχανή προσφέρει όλα αυτά (και όχι μόνο) τα πλεονεκτήματα, η *Java* αρχικά ήταν πιο αργή σε σχέση με άλλες προγραμματιστικές γλώσσες υψηλού επιπέδου (high-level) όπως η *C* και η *C++*. Εμπειρικές μετρήσεις στο παρελθόν είχαν δείξει ότι η *C++* μπορούσε να είναι αρκετές φορές γρηγορότερη από την *Java*. Ωστόσο γίνονται προσπάθειες από τη *Sun* για τη βελτιστοποίηση της εικονικής μηχανής, ενώ υπάρχουν και άλλες υλοποιήσεις της εικονικής μηχανής από διάφορες εταιρίες (όπως της *IBM*), οι οποίες μπορεί σε κάποια σημεία να προσφέρουν καλύτερα και σε κάποια άλλα χειρότερα αποτελέσματα. Επιπλέον με την καθιέρωση των μεταγλωττιστών JIT (Just In Time), οι οποίοι μετατρέπουν τον κώδικα byte απευθείας σε γλώσσα μηχανής, η διαφορά ταχύτητας από τη *C++* έχει μικρύνει κατά πολύ. [1]

1.3 Javascript

Η JavaScript (JS) είναι διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται.

Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη *C*. Η JavaScript αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη *Java*, αλλά γενικά οι *δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία*. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού *Self* και *Scheme*. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το *Node.js*) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side). [2]

1.4 Διαφορές Java και Javascript

Η κύρια διαφορά μεταξύ Java και Javascripts είναι ότι μπορούμε να γράψουμε αρχεία μόνο με κώδικα Java, ενώ η Javascript περιλαμβάνεται σαν μέρος κώδικα σε ένα αρχείο HTML. Μία άλλη σημαντική διαφορά είναι το πώς παρουσιάζονται αυτά τα αρχεία στον τελικό χρήστη. Τα αρχεία Java θα πρέπει να συγκεντρώνονται ως "γλώσσα μηχανής" πριν να τα «τρέξει» στο Web. Ο κώδικας Javascript γράφεται σαν αρχείο κειμένου μέσα σε ένα αρχείο html και εκτελείται μέσω του browser. Τα αρχεία Java applets λειτουργούν ανεξάρτητα από το έγγραφο HTML πιθανόν τα καλεί (επίσης πολλά αρχεία Java τρέχουν από πολλές συσκευές και κινητές συσκευές, και δεν απαιτούν ένα πρόγραμμα περιήγησης στο web). Κάτι που δεν ισχύει στην Javascript. [3]

1.5 Javascript Libraries vs Javascript Frameworks

Κάτι αρκετά σημαντικό πριν ξεκινήσουμε την ανάλυση των Javascript Frameworks είναι να καταλάβουμε τον λόγο ύπαρξης Javascript Libraries και των Javascript Frameworks.

Ένα Framework περιγράφει την δομή τους πώς θα πρέπει να είναι ο κώδικας. Λειτουργεί σαν πρότυπο καθώς περιέχει πρότυπα για υλοποίηση συγκεκριμένων προβλημάτων. Τέτοια παραδείγματα είναι το "Backbone", "requireJS", "socketIO".

Ένα Library από την άλλη περιέχει μία ολόκληρη εργαλειοθήκη και συναρτήσεις, που χρησιμοποιούμε για προγραμματισμό σε διάφορα επίπεδα. Χάρη στις βιβλιοθήκες οι προγραμματιστές μπορούν να δημιουργήσουν τα frameworks.

Κεφάλαιο 2 – Javascrpts Frameworks

2.1 JQuery

2.1.1 Τι είναι τα jQuery?

Το jQuery είναι γρήγορο, μικρό και διαθέτει πλούσια βιβλιοθήκη JavaScript. Κάνει τα πράγματα όπως ένα HTML έγγραφο, χειρίζεται τα γεγονότα και τα κινούμενα σχέδια, καθώς και τα Ajax στοιχεία πολύ πιο απλά, με τη βοήθεια ενός εύκολου στη χρήση API που λειτουργεί σε ένα πλήθος από browsers. Με ένα συνδυασμό ευελιξίας και επεκτασιμότητας, το jQuery έχει αλλάξει τον τρόπο που εκατομμύρια άνθρωποι γράφουν πλέον σε γλώσσα JavaScript. Εξάλλου το motto της jQuery είναι "write less, do more" - "γράψε λιγότερα, κάνε περισσότερα". [4]

Η jQuery είναι μία βιβλιοθήκη (**library**) JavaScript που χρησιμοποιείται από προγραμματιστές για τη ταχεία ανάπτυξη ιστοσελίδων και διαδικτυακών εφαρμογών που χρειάζονται μεγάλη **ευχρηστία** και **διαδραστικότητα** (interactivity). Η jQuery πρωτοεμφανίστηκε τον Ιανουάριο του 2006 στο BarCamp από τον John Resig. Πρόκειται για μια βιβλιοθήκη Javascript ανοιχτού κώδικα, υπό τις άδειες MIT License και την GNU General Public License.

Την βιβλιοθήκη jQuery μπορούμε να την κατεβάσουμε από επίσημη ιστοσελίδα που βρίσκεται στην διεύθυνση <http://jquery.com/>.

Η Microsoft και η Nokia ανακοίνωσαν τα σχέδιά τους να συνδυάσουν jQuery στις πλατφόρμες τους. Η Microsoft το υιοθέτησε αρχικά μέσα από το Visual Studio για χρήση σε ASP.NET AJAX πλαίσιο της Microsoft και ASP.NET MVC πλαίσιο, ενώ η Nokia το είχε εντάξει στην Web Run-Time πλατφόρμα ανάπτυξης. Η jQuery έχει επίσης χρησιμοποιηθεί σε MediaWiki από την έκδοση 1.16. [5]

Λόγω της αρχιτεκτονικής του jQuery, αρκετοί προγραμματιστές μπορούν να χρησιμοποιήσουν τις δομές της για να δημιουργήσουν plug-in κώδικα για να επεκταθεί η λειτουργικότητά του framework. Σήμερα υπάρχουν χιλιάδες jQuery plug-ins διαθέσιμα στο διαδίκτυο, που καλύπτουν ένα ευρύ φάσμα λειτουργιών, όπως η Ajax βοηθοί, οι υπηρεσίες web, τα datagrids, οι δυναμικές λίστες, τα XML και XSLT εργαλεία, η διαχείριση των cookies, τα συνδυασμένα παράθυρα.

Παρακάτω αναγράφουμε το πλήθος των εκδόσεων και πλαισίων της jquery.

| Version number | Release date | Additional notes |
|----------------|------------------------------------|--|
| 1.0 | August 26, 2006 | First stable release |
| 1.1 | January 14, 2007 | |
| 1.2 | September 10, 2007 | |
| 1.3 | January 14, 2009 | Sizzle Selector Engine introduced into core |
| 1.4 | January 14, 2010 | |
| 1.5 | January 31, 2011 | Deferred callback management, ajax module rewrite |
| 1.6 | May 3, 2011 | Significant performance improvements to the attr() and val() functions |
| 1.7 | November 3, 2011 | New Event APIs: .on() and .off(), while the old APIs are still supported. |
| 1.8.0 | August 9, 2012 | Sizzle Selector Engine rewritten, improved animations and \$(html, props) flexibility. |
| 1.9.0 | January 15, 2013 | Removal of deprecated interfaces and code cleanup |
| 1.9.1 | February 04, 2013 | |
| 2.0.0 | early 2013 | Dropping IE6-8 support for performance improvements and reduction in filesize |

Πίνακας 1 - Εκδόσεις JQuery

Το QUnit είναι ένα πλαίσιο αυτοματοποίησης δοκιμής που χρησιμοποιούν για τον έλεγχο των framework της jQuery. Η ομάδα της jQuery το χρησιμοποιεί για να ελέγξει τον κώδικα και τα plugins του, αλλά μπορεί να ελέγξει οποιοδήποτε γενικό κώδικα JavaScript, συμπεριλαμβανομένου και του κώδικα server-side JavaScript.

Από το 2011, η Ομάδα Ελέγχου jQuery χρησιμοποιεί το QUnit και το TestSwarm για να δοκιμάσει κάθε jQuery framework. [5]

Παρακάτω ακολουθεί ένας πίνακας ελέγχων συμβατότητας των Framework jQuery με τους περιηγητές Ιστού.

| | Internet Explorer | Chrome | Firefox | Safari | Opera |
|------------|-------------------|---------------------|---------------------|---------------------|---------------------|
| jQuery 1.x | 6+ | Current – 1 version | Current – 1 version | Current – 1 version | Current – 1 version |
| jQuery 2.x | 9+ | | | | |

Πίνακας 2 - Συμβατότητα Frameworks με Browsers

Κάθε πρόβλημα με τα jQuery frameworks στα παραπάνω προγράμματα περιήγησης θα πρέπει να θεωρείται και αναφέρεται ως ένα bug της jQuery. Πιο συγκεκριμένα η έκδοση "current -1 version" δηλώνει ότι θα υποστηρίξει την τρέχουσα σταθερή έκδοση του browser και τις εκδόσεις που προηγήθηκαν. Για παράδειγμα, αν η τρέχουσα έκδοση του προγράμματος περιήγησης είναι 24.x, υποστηρίζει τις 24.x και εκδόσεις 23.x.[6]

Η jQuery είναι μια βιβλιοθήκη με συναρτήσεις γραμμένες σε **JavaScript**. Το πλεονέκτημα της jQuery είναι ο **συμπαγής κώδικας** και η **ευκολία** στην **εκμάθηση** αλλά και στην **συγγραφή**.

Απαραίτητες γνώσεις που πρέπει να κατέχει κάποιος πριν αρχίσει να ασχολείται με την jQuery είναι:

- HTML
- CSS
- Java

2.1.2 Πλεονεκτήματα της jQuery

- *Ακολουθεί την αρχή KISS (Keep It Simple Stupid):* Η βιβλιοθήκη JQuery προσπαθεί να υπεραπλουστεύσει τον προγραμματισμό σε Javascript προσφέροντας πραγματικά απλούς μηχανισμούς και εντολές μέσω του framework της.

- **Παρέχει πλήρη και αναλυτικότητα τεκμηρίωση** που συμπληρώνεται από την εκτεταμένη παρουσία ηλεκτρονικών βοηθημάτων. Εκτός από την πολύ καλοδουλεμένη τεκμηρίωσή του JQuery, οι ενδιαφερόμενοι μπορούν να ανατρέξουν και στην σελίδα του Visual JQuery όπου μπορούν να βρουν μία εναλλακτική αλλά πολύ βολική, από άποψη δομής, τεκμηρίωση.
- **Υποστηρίζεται από μία πάρα πολύ ενεργή κοινότητα:** Όπως και τα περισσότερα open source έργα λογισμικού, έτσι και για το JQuery η ύπαρξη μιας κατά το μέγιστο δυνατό ενεργής κοινότητας αποτελεί τον ακρογωνιαίο λίθο για την ανάπτυξη και ευημερία του.
- **Μικρό μέγεθος:** Το γεγονός ότι το βασικό πακέτο της JQuery είναι μόλις 20Kb αφενός επιβεβαιώνει την πρώτη παρατήρηση, ότι δηλαδή η φιλοσοφία της έγκειται στην απλότητα και αφετέρου κάνει πολύ εύκολη την κατανόηση της αρχιτεκτονικής της.
- **Ποικιλία χαρακτηριστικών:** Η JQuery δίνει τη δυνατότητα στον χρήστη να χρησιμοποιήσει σχεδόν το σύνολο των δυνατοτήτων που προσφέρει η γλώσσα JavaScript. Από απλά χαρακτηριστικά που σχετίζονται με βασικές λειτουργίες εμφάνισης / απόκρυψης, ως Ajax κλήσεις και σύνθετα εφέ.
- **Επεκτασιμότητα:** Η λογική με την οποία είναι φτιαγμένη η JQuery είναι απλή πράγμα που αντικατοπτρίζεται και στον ίδιο της τον κώδικα. Αυτό κάνει πολύ εύκολη την επέκτασή / τροποποίησή της. [7]

2.1.3 Σύνταξη της jQuery

Την βιβλιοθήκη jQuery μπορούμε να την ενσωματώσουμε στην ιστοσελίδα μας, εισάγοντας τον παρακάτω κώδικα στο head τμήμα μιας ιστοσελίδας HTML:

```
<script type="text/javascript" src="PATH/jquery.js"></script>
```

ή υπάρχει κι ο online τρόπος, με έμμεση αναφορά, δηλ:

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"
type="text/javascript"></script>
```

Παρακάτω ακολουθεί ολοκληρωμένη η σελίδα html:


```

<html>
<head>
  <script type="text/javascript" src="PATH/jquery.js"></script>
  <script type="text/javascript">
    // κώδικας Javascript
  </script>
</head>
<body>
  <!-- το περιεχόμενο της σελίδας μας -->
</body>
</html>

```

Σημείωση: Όταν λέμε **PATH** εννοούμε την αναφορά σε σχέση με το html έγγραφο. Αν το αρχείο jquery.js είναι στο ίδιο επίπεδο/φάκελο, τότε μπορούμε να το παραλείψουμε. Αν είναι στο φάκελο {js} τότε γράφουμε ...src="js/jquery.js". Το ιδανικό είναι να υπάρχουν τα αρχεία javascript σε ξεχωριστό φάκελο με όνομα js.

Η JQuery διαβάζει και χειρίζεται το DOM model (Document Object Model). Προϋπόθεση πριν αρχίσουμε να γράφουμε κώδικα είναι να αρχικοποιήσουμε ένα event για το έγγραφο (document) ως εξής:

```

$(document).ready(function() {
  // κώδικας
});

```

Προσθέτοντας έναν υπερσύνδεσμο στο κείμενο Link :

```

<a href="">Link</a>

```

Και ενημερώνοντας τον κώδικα μας:

```

$(document).ready(function() {
  $("a").click(function() {
    alert("Hello world!"); }); });

```

Όταν ο χρήστης πατήσει τον υπερσύνδεσμο, θα εμφανιστεί ένα παραθυράκι που θα λέει Hello World!

Selectors και events

Τα στοιχεία της σελίδας μας (html elements) όπως οι παράγραφοι ή οι σύνδεσμοι μπορούν να επιλεγθούν από την JQuery και να κάνουμε αλλαγές πάνω τους.

Ας δημιουργήσουμε μία λίστα με id “months”.

```
<ul id="months">
  <li>Ιανουάριος</li>
  <li>Φεβρουάριος</li>
  <li>Μάρτιος</li>
</ul>
```

Για να επιλέξουμε την λίστα, στην Javascript έχουμε: document.getElementById(“months”).

Με την JQuery γίνεται ως εξής:

```
$(document).ready(function() {
  $("#months").addClass("red");
});
```

Έτσι προσθέσαμε την κλάση red στην λίστα μας. Και μπορούμε να την μορφοποιήσουμε.

Τώρα, ας προσθέσουμε την κλάση blue σε ένα στοιχείο (li) της λίστας.

```
$(document).ready(function() {
  $("#months > li").addClass("blue");
});
```

Αυτός ο κώδικας επιλέγει όλα τα υπό - στοιχεία δηλ. τους μήνες ουσιαστικά και προσθέτει την κλάση blue.

Τα παραδείγματα μελετήθηκαν και υλοποιήθηκαν με την βοήθεια της ιστοσελίδας www.ituts.gr. [8]

Γενικά το framework jquery μπορούμε να το χρησιμοποιήσουμε για να κάνουμε πιο λειτουργικές και αποτελεσματικές τις σελίδες μας καθώς και τα web application.

2.1.4 jQuery UI

Η jQuery από μόνη της παρέχει πληθώρα εργαλείων για να χτίσετε τη σελίδα σας όπως ακριβώς τη φανταζόσασταν. Παρ' όλα αυτά, υπάρχει και μια βιβλιοθήκη-επέκταση με το όνομα jQuery UI, η οποία ουσιαστικά προσθέτει νέες λειτουργίες, animations και εφέ στην ήδη υπάρχουσα βιβλιοθήκη της jQuery. Επίσης, σας δίνει τη δυνατότητα να χρησιμοποιήσετε έτοιμα widgets όπως ημερολόγιο, λειτουργίες για πίνακες, αλλά και να δημιουργήσετε θέματα χρωμάτων για τα widgets σας. Μπορείτε να μάθετε περισσότερα γι' αυτήν στο www.jqueryui.com.

2.1.5 jQuery Mobile

Το jQuery Mobile είναι ο ευκολότερος τρόπος για να οικοδομήσουμε ιστοσελίδες και εφαρμογές που έχουν πρόσβαση σε όλες τις δημοφιλείς συσκευές smartphone, tablet και desktop.

Το jQuery Mobile έχει ευρεία υποστήριξη και συντριπτική πλειοψηφία σε όλες τις σύγχρονες desktop, smartphone, tablet και e-reader πλατφόρμες. Επιπλέον, τα κινητά τηλέφωνα και παλαιότερα προγράμματα περιήγησης που υποστηρίζονται λόγω της προοδευτικής προσέγγισης του jQuery Mobile.

Το jQuery Mobile χρησιμοποιεί ένα 3-επίπεδο σύστημα διαβαθμισμένης υποστήριξης πλατφόρμας: A (πλήρες), B (πλήρης μείον Ajax), C (βασική HTML). Η οπτική πιστότητά του και η ομαλότητα της μετάβασης στη σελίδα εξαρτώνται σε μεγάλο βαθμό από τις δυνατότητες απόδοσης CSS της συσκευής και της πλατφόρμας. Αλλά μην ξεχνάμε ότι αυτή είναι και η φύση του διαδικτύου. [9]

Παρακάτω βλέπουμε ποιες πλατφόρμες υποστηρίζουν ποιό σύστημα:

A-grade - Full enhanced experience with Ajax-based animated page transitions.

- **Apple iOS 3.2*-6.0** - Tested on the original iPad (4.3 / 5.0), iPad 2 (4.3), iPad 3 (5.1 / 6.0), original iPhone (3.1), iPhone 3 (3.2), 3GS (4.3), 4 (4.3 / 5.0), and 4S (5.1 / 6.0)
- **Android 2.1-2.3** - Tested on the HTC Incredible (2.2), original Droid (2.2), HTC Aria (2.1), Google Nexus S (2.3). Functional on 1.5 & 1.6 but performance may be sluggish, tested on Google G1 (1.5)
- **Android 3.2 (Honeycomb)** - Tested on the Samsung Galaxy Tab 10.1 and Motorola XOOM

- **Android 4.0 (ICS)** - Tested on a Galaxy Nexus. Note: transition performance can be poor on *upgraded* devices
- **Android 4.1 (Jelly Bean)** - Tested on a Galaxy Nexus and Galaxy 7
- **Windows Phone 7-7.5** - Tested on the HTC Surround (7.0) HTC Trophy (7.5), LG-E900 (7.5), Nokia Lumia 800
- **Blackberry 6.0** - Tested on the Torch 9800 and Style 9670
- **Blackberry 7** - Tested on BlackBerry® Torch 9810
- **Blackberry Playbook (1.0-2.0)** - Tested on PlayBook
- **Palm WebOS (1.4-2.0)** - Tested on the Palm Pixi (1.4), Pre (1.4), Pre 2 (2.0)
- **Palm WebOS 3.0** - Tested on HP TouchPad
- **Firefox Mobile 15** - Tested on Android 2.3 and 4.1 devices
- **Chrome for Android 18** - Tested on Android 4.0 and 4.1 devices
- **Skyfire 4.1** - Tested on Android 2.3 device
- **Opera Mobile 11.5-12**: Tested on Android 2.3
- **Meego 1.2** - Tested on Nokia 950 and N9
- **Tizen (pre-release)** - Tested on early hardware
- **Samsung Bada 2.0** - Tested on a Samsung Wave 3, Dolphin browser
- **UC Browser** - Tested on Android 2.3 device
- **Kindle 3 and Fire** - Tested on the built-in WebKit browser for each
- **Nook Color 1.4.1** - Tested on original Nook Color, not Nook Tablet
- **Chrome Desktop 11-21** - Tested on OS X 10.7 and Windows 7
- **Safari Desktop 4-5** - Tested on OS X 10.7 and Windows 7
- **Firefox Desktop 4-15** - Tested on OS X 10.7 and Windows 7
- **Internet Explorer 7-10** - Tested on Windows XP, Vista and 7
- **Opera Desktop 10-12** - Tested on OS X 10.7 and Windows 7

B-grade - Enhanced experience except without Ajax navigation features.

- **Blackberry 5.0***: Tested on the Storm 2 9550, Bold 9770

- **Opera Mini 7** - Tested on iOS 5.1 and Android 2.3
- **Nokia Symbian^3** - Tested on Nokia N8 (Symbian^3), C7 (Symbian^3), also works on N97 (Symbian^1)

C-grade - Basic, non-enhanced HTML experience that is still functional

- **Blackberry 4.x** - Tested on the Curve 8330
- **Windows Mobile** - Tested on the HTC Leo (WinMo 5.2)
- **All older smartphone platforms and featurephones** - Any device that doesn't support media queries will receive the basic, C grade experience

2.2 Dojo framework

Το framework Dojo είναι ένα open-source λογισμικό. Το σύνολο των εργαλείων από το οποίο αποτελείτε είναι διαθέσιμο στην επίσημη ιστοσελίδα όπου μπορεί ο χρήστης να το κατεβάσει ως ένα ZIP αρχείο στο παρακάτω σύνδεσμο <http://dojotoolkit.org/download/>, καθώς επίσης και στο Google CDN. Η εργαλειοθήκη περιλαμβάνει περίπου τρεις χιλιάδες JavaScript modules, εκτός από εικόνες και άλλους πόρους.

Το Dojo Toolkit είναι οργανωμένο σε διάφορα μέρη:

- **dojo** περιέχει τον πυρήνα και πιο μη-οπτικές ενότητες,
- **dijit** είναι μια βιβλιοθήκη της διεπαφής χρήστη για ενότητες και widgets,
- **dojox** κατέχει διάφορα modules που δεν έχουν ακόμη καθιερωθεί στο μέρος dojo ή dijit,
- **util** περιλαμβάνει την κατασκευή εργαλείων, όπως τη βελτιστοποίηση, την τεκμηρίωση, το στυλ ελέγχου και δοκιμών. [10]

Τα Dojo widgets είναι ενθέματα που συνδυάζουν κώδικα JavaScript, HTML και CSS και αυτό επιτρέπει να είναι συμβατό με αρκετούς Browsers. Μερικά από τα πιο βασικά widgets τα χρησιμοποιούμε για την δημιουργία:

- Μενού
- Πίνακες
- Δυναμικά πεδία
- Δυσδιάστατα σχέδια

- Εφέ κίνησης, διαφάνειες slide.
- Ποικιλία φορμών και φορμών επικύρωσης.
- Δημιουργία ημερολογίου.
- Δημιουργία χαρτών. [10]

2.2.1 Dojo Toolkit

Πλεονεκτήματα:

- Το Debug του κώδικα γίνεται από το Firebug των Browsers
- Το Dojo μπορεί να ενσωματωθεί σε ένα Zend Framework
- Ιεραρχική σειρά φόρτωσης των πακέτων συστήματος
- Το Dojo είναι ένα Framework – Πλαίσιο εργασίας

Μειονεκτήματα:

- Το Dojo αποτυγχάνει στην on-line τεκμηρίωση. [11]

2.2.2 Dojo Mobile

Το Dojo Mobile είναι μια κατηγορία πλαισίου JavaScript που επιτρέπει την ταχεία ανάπτυξη των κινητών εφαρμογών web για τις σύγχρονες webkit-enabled φορητές συσκευές όπως το iPhone, το iPod Touch, iPad, Android και τα RIM smartphones και tablets. Είναι το βασικότερο εργαλείο mobile που χρησιμοποιούνται για την προεπισκόπηση της IBM Mobile Technology και το Web 2. Όπως και πολλά frameworks, έτσι και το Dojo παρέχει μια μηχανή JavaScript που σας επιτρέπει να γράφετε τις επιλογές CSS. [12]

2.3 ExtJS

Το ExtJS Framework είναι ένα πλαίσιο εφαρμογής JavaScript για την δημιουργία διαδραστικών web εφαρμογών. Χρησιμοποιεί Ajax, DHTML και DOM στοιχεία και έχει σαν βάση το YUI Framework. Λειτουργεί σε όλα τα σύγχρονα προγράμματα περιήγησης από IE6 με την πιο πρόσφατη έκδοση του Chrome. Δίνει τη δυνατότητα να δημιουργήσουμε τις καλύτερες cross-platform εφαρμογές χρησιμοποιώντας ένα πρόγραμμα περιήγησης.

Το Ext JS περιλαμβάνει μια σειρά από GUI-based στοιχείων (ή «widgets») για χρήση σε εφαρμογές web:

- έλεγχος συμπλήρωσης πεδίων κειμένου και textarea
- πεδία ημερομηνίας με ένα pop-up επιλογή
- αριθμητικά πεδία
- πλαίσιο λίστας και σύνθετα πλαίσια
- κουμπιά ελέγχου
- html editor
- δημιουργία tabs
- γραμμές εργαλείων
- desktop applications
- sliders
- διαγράμματα και διανυσματικά γραφικά

Πολλά από αυτά τα χαρακτηριστικά μπορούν να επικοινωνούν με έναν web server, χρησιμοποιώντας Ajax στοιχεία.

Υπάρχουν αρκετές εκδόσεις του ExtJs Framework. Παρακάτω αναλύουμε τις βασικότερες:

Το Ext JS έκδοση 2.0 κυκλοφόρησε στις 4 Δεκεμβρίου 2007. Αυτή η έκδοση προήχθη ως παροχή διασύνδεσης και χαρακτηριστικά παρόμοια με εκείνα που παραδοσιακά συνδέονται με desktop εφαρμογές.

Το Ext JS έκδοση 3.0 κυκλοφόρησε στις 6 Ιουλίου 2009. Αυτή η προστιθέμενη επικοινωνιακή υποστήριξη REST και μια νέα πλατφόρμα server Ext.Direct έκδοση. Στα νέα στοιχεία συγκαταλέγεται η διαχείριση flash αντικειμένων και.

Η έκδοση 4.0 του πλαισίου Ext κυκλοφόρησε στις 26 Απριλίου 2011. Περιλαμβάνει μια πλήρως αναθεωρημένη ταξική δομή, ένα αναθεωρημένο πακέτο δεδομένων, μια κινούμενη εικόνα και ένα προαιρετικό στοιχείο αρχιτεκτονικής που παρέχει ένα μοντέλο-view-controller (MVC).

Στις 15 Ιουνίου 2010 ανακοινώθηκε η συγχώνευση των Ext JS με jQTouch και Raphaël σαν μία νέα οργάνωση που ονομάζεται Sencha. Το Ext JS εξακολουθεί να είναι διαθέσιμο ως κύριο προϊόν για την νέα ιστοσελίδα Sencha. [13]

Το Framework Ext JS 4 υποστηρίζει όλα τα σημαντικά προγράμματα περιήγησης στο Web, από τον Internet Explorer 6 μέχρι και την πιο πρόσφατη έκδοση του Google Chrome. Κατά τη διάρκεια της ανάπτυξης, όμως, σας προτείνουμε να επιλέξετε ένα από τα ακόλουθα προγράμματα περιήγησης για την καλύτερη εμπειρία εντοπισμού σφαλμάτων:

Google Chrome 10 +

Apple Safari 5 +

Mozilla Firefox 4 + με το Firebug Plugin Web Development. [14]

2.4 MooTools

Το MooTools είναι ένα συμπαγές Object-Oriented JavaScript πλαίσιο που έχει σχεδιαστεί για προγραμματισμό JavaScript. Μας επιτρέπει να γράψουμε ένα ισχυρό, ευέλικτο, και cross-browser κώδικα με κομψό, καλά τεκμηριωμένο, και συνεκτικό API.

Το MooTools τηρεί αυστηρές προδιαγραφές και δεν έχει "warmings". Είναι ευρέως τεκμηριωμένο και έχει μεταβλητές οριζόμενες: είναι ευχάριστο στην περιήγησή του και εύκολο στο προγραμματισμό του.

- Το MooTools περιλαμβάνει έναν αριθμό συστατικών, αλλά δεν χρειάζονται όλα να φορτωθούν για κάθε εφαρμογή. Ορισμένες από τις κατηγορίες συστατικών είναι:
- Core: Μια συλλογή από βοηθητικές λειτουργίες που απαιτούν όλα τα υπόλοιπα στοιχεία
- More: Μία επίσημη συλλογή add-ons που επεκτείνει το Core και παρέχει αυξημένη λειτουργικότητα.
- Class: Η βασική βιβλιοθήκη για αντικείμενα Classes.
- Native: Μια συλλογή από JavaScript Native βελτιώσεις αντικειμένων, που προσθέτουν λειτουργικότητα, συμβατότητα, και οι νέες μέθοδοι που απλοποιούν την κωδικοποίηση.
- Element: Περιέχει ένα μεγάλο αριθμό από βελτιώσεις και την τυποποίηση της συμβατότητας με το αντικείμενο HTML Element.
- Fx.: Μία συλλογή από εφέ-API για να τονώσουν τα στοιχεία της σελίδας.
- Request: Περιλαμβάνει XHR interface, Cookie, JSON, HTML και ανάκτησης-ειδικά εργαλεία για τους προγραμματιστές προς εκμετάλλευση.
- Window: Παρέχει ένα cross-browser interface για τον πελάτη-συγκεκριμένες πληροφορίες, όπως οι διαστάσεις του παραθύρου. [15]

Το MooTools framework είναι συμβατό και ελέγξιμο με τους browsers:

- Safari 3+
- Internet Explorer 6+

- Mozilla Firefox 2+
- Opera 9+
- Chrome 4+

Το MooTools δίνει έμφαση στην μοντελοποίηση και στην επαναχρησιμοποίηση κώδικα. Κάθε πλαίσιο JavaScript έχει τη φιλοσοφία του και το MooTools προσπαθεί να επωφεληθεί πλήρως από την ευελιξία και τη δύναμη της JavaScript με έναν τρόπο που δίνει έμφαση στην μεγαλύτερη προσαρμοστικότητα και στην επαναχρησιμοποίηση κώδικα. Το MooTools επιτυγχάνει αυτούς τους στόχους με τρόπο που είναι διαισθητικό σε έναν προγραμματιστή που προέρχεται από μια κατηγορία με βάση τη γλώσσα κληρονομιάς, όπως Java με το αντικείμενο Class MooTools.

Το Class είναι ένα αντικείμενο του ζεύγους κλειδιού / τιμής που μπορεί να περιέχει είτε ιδιότητες ή μεθόδους (συναρτήσεις). Η επαναχρησιμοποίηση του κώδικα επιτυγχάνεται μέσω της μεγιστοποίησης της ισχύος της πρωτότυπης κληρονομιάς JavaScript, αλλά σε μια σύνταξη αντικείμενου Class πιο εξοικειωμένη με την κλασική μοντέλα κληρονομιά. [15][16]

2.5 Prototype

Το πλαίσιο Prototype JavaScript είναι ένα πλαίσιο JavaScript που δημιουργήθηκε από τον Sam Stephenson τον Φεβρουάριο του 2005, ως μέρος του ιδρύματος για την υποστήριξη Ajax στην γλώσσα προγραμματισμού Ruby. Υλοποιείται ως ένα ενιαίο αρχείο του κώδικα JavaScript, που συνήθως ονομάζεται Prototype.js. Το Prototype framework χρησιμοποιείται κατά 2,9% επί του συνόλου των ιστοχώρων.

Τα χαρακτηριστικά του Prototype JS Framework κυμαίνονται από συντομεύσεις προγραμματισμού για βασικές λειτουργίες μέχρι την αντιμετώπιση μεγάλων μεθόδων τύπου XMLHttpRequest. Παρέχει επίσης λειτουργίες δυνατότητες για την υποστήριξη τάξεων και αντικείμενων που η γλώσσα JavaScript στερείται. Στην Javascript μία συνάρτηση η οποία δημιουργεί ένα αντικείμενο μπορεί να έχει μία ιδιότητα τύπου prototype και κάθε φορά που κάποιο αντικείμενο του ανατίθεται η ιδιότητα αυτή, τότε θα χρησιμοποιηθεί σαν πρωτότυπο για την δημιουργία του αντικειμένου. Το πλαίσιο Prototype δεν πρέπει να συγχέεται με αυτό το χαρακτηριστικό της γλώσσας.

Μερικές από τις βασικές μεθόδους του πλαισίου Prototype είναι η μέθοδος του δολαρίου, \$(), μπορεί να χρησιμοποιηθεί ως συντομογραφία για τη λειτουργία getElementById. Οπότε

για να αναφερθούμε σε ένα DOM στοιχείο της σελίδας HTML, η συνήθης λειτουργία αναγνώριση ενός στοιχείου είναι:

```
document.getElementById("id_of_element").style.color = "#ffffff";
```

Με την χρήση της μεθόδου `$()` ο κώδικας μειώνεται αρκετά και για τις ίδιες εντολές έχουμε το παρακάτω:

```
$("#id_of_element").setStyle({color: '#ffffff'});
```

Σε ένα διαφορετικό παράδειγμα θα δούμε:

```
var domElement = document.getElementById("id_of_element");  
prototypeEnhancedDomElement = $(domElement);
```

Τέλος το πλαίσιο Prototype κάνει χρήση αρκετών στοιχείων Ajax. Σε μια προσπάθεια να μειώσει την ποσότητα του κώδικα που απαιτείται για να εκτελεστεί μια λειτουργία cross-browser XMLHttpRequest, το Prototype παρέχει το αφηρημένο (abstract) αντικείμενο Ajax στα διάφορα προγράμματα περιήγησης. Έχει δύο κύριες μεθόδους: `Ajax.Request()` και `Ajax.Updater()`. Υπάρχουν δύο μορφές του αντικειμένου Ajax. Η `Ajax.Request` επιστρέφει την παραγωγή πρώτων XML αρχείων από μια κλήση AJAX, ενώ η `Ajax.Updater` θα επιστρέφει την επιστροφή μέσα σε ένα καθορισμένο αντικείμενο DOM. Η `Ajax.Request`, στο παρακάτω παράδειγμα, βρίσκει τις τρέχουσες αξίες των στοιχείων εισόδου σε μορφή HTML, εκδίδει μια αίτηση HTTP POST στο διακομιστή με τα ζεύγη ονόματος / τιμής, και τρέχει μια προσαρμοσμένη συνάρτηση (που ονομάζεται `showResponse`), όταν η απόκριση HTTP έχει ληφθεί από τον διακομιστή:

```
new Ajax.Request("http://localhost/server_script", {  
  parameters: {  
    value1: $F("form_element_id_1"),  
    value2: $F("form_element_id_2")  
  },  
  onSuccess: showResponse,  
  onFailure: showError  
});
```

Το πλαίσιο Prototype παρουσιάζει αρκετά προβλήματα που έχουν να κάνουν κυρίως με την συμβατότητα με τους browsers και θέματα απόδοσης. Στις επόμενες εκδόσεις του framework οι προγραμματιστές θα πρέπει να το έχουν βελτιώσει για να μπορεί να ανταποκρίνεται επάξια μαζί με τα άλλα Javascript πλαίσια που υπάρχουν. [17]

2.6 YUI Javascript

Η Yahoo User Interface Βιβλιοθήκη (YUI) είναι ένα open - source JavaScript που δημιουργήθηκε από τους μηχανικούς της Yahoo! το 2005 και δόθηκε στο κοινό για ελεύθερη χρήση τον Φεβρουάριο του 2006. Ο λόγος ανάπτυξής της ήταν για την δημιουργία πλούσιων διαδραστικών web εφαρμογών με τη χρήση στοιχείων Ajax, DHTML και DOM scripting. Η βιβλιοθήκη YUI περιλαμβάνει αρκετά στοιχεία CSS.

Τον Σεπτέμβριο του 2009, κυκλοφόρησε το Yahoo! YUI 3, μια νέα έκδοση του YUI ανοικοδομήθηκε από το μηδέν για τον εκσυγχρονισμό της υπάρχουσας βιβλιοθήκης και να ενσωματώσει τα σφάλματα από YUI 2. Μεταξύ των βελτιώσεων περιλαμβάνεται ένας CSS selector, όπως στην jQuery, για την ανάκτηση στοιχείων DOM, δίνεται μεγαλύτερη έμφαση στην ανάλυση των modules, ένα μικρότερο αρχείο σπόρων φορτώνει άλλες μονάδες, όταν είναι αναγκαίο, καθώς και μια ποικιλία συντακτικών τροποποιήσεων με στόχο να κάνουν το κώδικα γρηγορότερο και ευκολότερο.

Στα χαρακτηριστικά της YUI βιβλιοθήκης συγκαταλέγονται ο πηγαίος κώδικας της που είναι μικρό πακέτο εργαλείων για την διαχείριση των events και των DOM στοιχείων, η ύπαρξη YUI καθολικού αντικειμένου που περιλαμβάνει βοηθητικά προγράμματα, javascript loader κά, συλλογή DOM στοιχείων για την διαχείριση CSS στοιχείων. Στα βοηθητικά στοιχεία μπορούμε να αναφέρουμε τα στοιχεία που επιτρέπουν κίνηση, διαχείριση ιστορικού περιήγησης, πηγή δεδομένων, φόρτωση εικόνων, επιλογέας κίνησης κά. [16] Επίσης αν συνδεθούμε στην επίσημη σελίδα της YUI βιβλιοθήκης θα δούμε τους 4 βασικούς λόγους επιλογής της, γρήγορη, ολοκληρωμένη, επαγγελματική και ελεύθερη και «ανοικτή» για τους προγραμματιστές. [18]

Στις επόμενες εικόνες απεικονίζεται η αναζήτηση πληροφοριών και κώδικα για τις βιβλιοθήκες jQuery, Dojo, Mootools, YUI και EXTJS στην πάροδο του χρόνου και τα πιο δημοφιλή Javascript Framework.

Εξερεύνηση τάσεων

Δημοφιλείς αναζητήσεις

Ενδιαφέρον με την πάροδο του χρόνου

Ο αριθμός 100 αναπαριστά το μέγιστο ενδιαφέρον αναζητήσεων

Τίτλοι ειδήσεων Πρόβλεψη ?

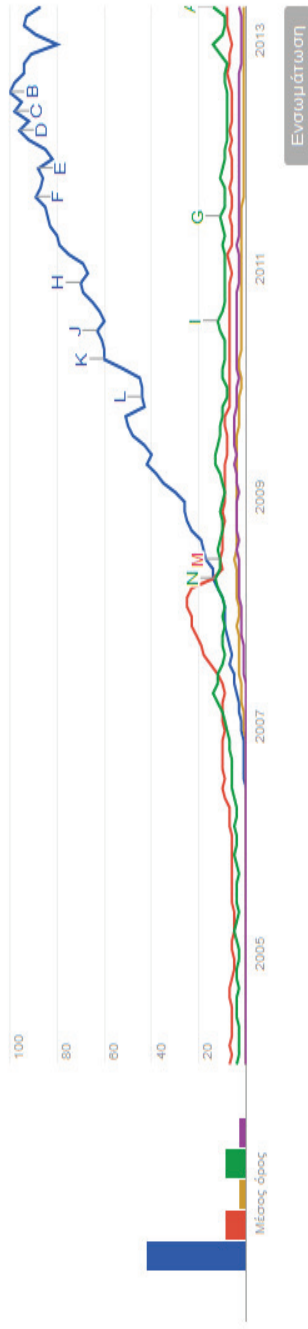
Όροι αναζήτησης

- jquery
- dojo
- mootools
- yui
- extjs

▶ Άλλες συγκρίσεις

Περιορισμός σε

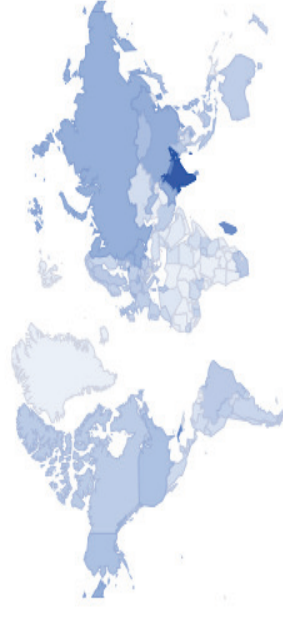
- Αναζήτηση Ιστού ▶
- Παγκοσμίως ▶
- 2004 - παρόν ▶
- Όλες οι κατηγορίες ▶



Ενσωμάτωση

jquery dojo mootools yui extjs

Τοπικό ενδιαφέρον



0 100

Περιοχή | Πόλη

▶ Προβολή αλλαγών με την πάροδο του χρόνου ?

Ενσωμάτωση

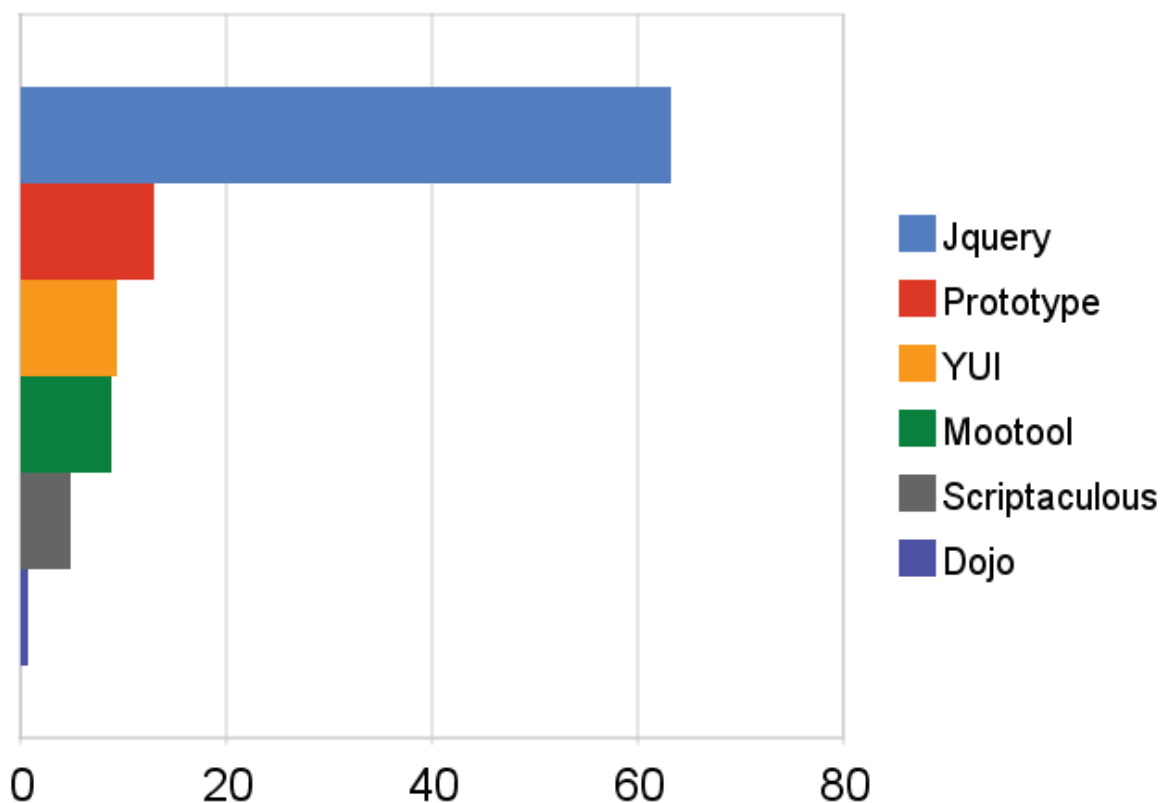
Σχετικοί όροι

| Κορυφές | Ανερχόμενες |
|---------------|-------------|
| javascript | 100 |
| ajax jquery | 60 |
| ajax | 60 |
| jquery css | 55 |
| jquery ui | 55 |
| jquery get | 40 |
| jquery on | 30 |
| jquery form | 30 |
| jquery select | 30 |
| jquery mobile | 30 |

Ενσωμάτωση

Εικόνα 1 – Αναζήτηση πληροφοριών για βασικές JavaScript βιβλιοθήκες

Javascript Framework Popularity



Εικόνα 2 - Δημοφιλή Javascript Frameworks

Τρέχοντας τις 100.000 πιο δημοφιλείς ιστοσελίδες με τον μηχανισμό Alexa καταλήξαμε στο συμπέρασμα ότι το 63% αυτών των σελίδων χρησιμοποιούν πλαίσιο το JQuery framework. Η δεύτερη επιλογή τους είναι το Prototype με ποσοστό 13%, το τρίτο και τέταρτο είναι τα YUI και Mootool με ποσοστό 9%. Αντιλαμβανόμαστε ότι το κυρίαρχο πλαίσιο είναι το JQuery. Το οποίο είναι και το πιο ευρέως χρησιμοποιήσιμο.

Κεφάλαιο 3 – Αξιολόγηση Απόδοσης JS Framework

Στο προηγούμενο κεφάλαιο αναλύσαμε τα JS Framework καθώς και την εξέλιξή τους. Στο παρόν κεφάλαιο θα αναλύσουμε τον λόγο μέτρησης της απόδοσης των πλαισίων καθώς και την μεθοδολογία που ακολουθήσαμε για το πρακτικό μέρος της εργασίας.

3.1 Μετρήσεις κώδικα

Για να μετρηθεί η ποιότητα του κώδικα του κάθε πλαισίου (framework), θα έπρεπε να βασιστούμε σε γνωστές μετρήσεις ποιότητας για το μέγεθος, την πολυπλοκότητα και την διατηρησιμότητα του κώδικα σαν σύνολο και ανά λειτουργία.

Οι μετρήσεις μεγέθους είναι ένα σύνολο από μετρήσεις που προσπαθούν να ποσοτικοποιήσουν το λογισμικό "μέγεθος". Ως προς το πόσο μεγάλο είναι το πρόγραμμα ή μία λειτουργία - μέθοδος. Το πιο αντιπροσωπευτικό μετρικό μέγεθος είναι οι γραμμές κώδικα (LOC). *Όσο μεγαλύτερος είναι ο αριθμός των LOC, τόσο πιο δύσκολο είναι να βρεθούν και να διορθωθούν τα λάθη του.* Άλλες σημαντικές μετρήσιμες ιδιότητες είναι ο αριθμός των δηλώσεων μεταβλητών, αντικειμένων κ.α., τις γραμμές σχολίων και η αναλογία μεταξύ των σχολίων και κώδικα. Η αναλογία αυτή πρέπει να είναι τουλάχιστον 10% έτσι ώστε να μπορεί να δείξει και να σχολιάσει μια λειτουργία που είναι δύσκολο να κατανοήσουμε.

Οι μετρήσεις πολυπλοκότητας μετρούν τη δυσκολία της κατανόησης του κώδικα και βρίσκουν ένα λογικό σφάλμα μέσα σε αυτό. Η πιο σημαντική μέτρηση σε αυτή την κατηγορία είναι η *McCabe Cyclomatic* πολυπλοκότητα, το οποίο μετρά τον αριθμό των γραμμικά ανεξάρτητων διαδρομών μέσα από ένα module (ένθετο) του προγράμματος. Είναι μία από τις πιο ευρέως αποδεκτές μετρήσεις λογισμικού και είναι ανεξάρτητη από τη γλώσσα προγραμματισμού και τη μορφή της. Άλλοι σημαντικοί παράγοντες πολυπλοκότητας λειτουργίας που δείχνουν ότι μέρος του κώδικα είναι δύσκολο να διατηρηθεί είναι τα *Branches* (ο αριθμός των μονοπατιών σε λογική ροή κατά την εκτέλεση μιας συνάρτησης) και το βάθος - Depth (το μέγιστο πλήθος των δομών **αν-τότε-αλλιώς** μέσα σε μια συνάρτηση) *Branches* προκαλούνται από τη χρήση του if, else, case, default, catch, finally, for, while, do-while, && και ||. Η μέτρηση της *Cyclomatic Πολυπλοκότητας* (CC) δείχνει τον αριθμό των ακμών που σχηματίζονται από ένα Branche (E), μείον τον αριθμό των κόμβων του Branche (N), καθώς και ο αριθμός των σημείων εξόδου στη Μέθοδο (R) (return ή throw) ($CC = E - N + R + 1$). Τέλος, μια μέτρηση πολυπλοκότητας για το σύνολο του λογισμικού είναι ο αριθμός του σε ενότητες ή λειτουργίες.

Οι μετρήσεις συντηρησιμότητας χρησιμοποιούνται κυρίως για να καθορίσουν εάν το πρόγραμμα ή μια συνάρτηση έχει ένα υψηλό, μέσο ή χαμηλό βαθμό δυσκολίας να διατηρηθεί. Οι Halstead μετρήσεις και ο Δείκτης συντηρησιμότητας (MI) είναι δημοφιλείς μετρήσεις σε αυτή την κατηγορία. Ο MI είχε επιλεγεί, δεδομένου ότι είναι μια ενιαία τιμή για τον αριθμό εκτίμησης της σχετικής συντήρησης του κώδικα και ενσωματώνει τις Halstead μετρήσεις. Ο MI υπολογίζεται με ορισμένους τύπους από τις γραμμές-του-κώδικα (LOC), τη McCabe και τις Halstead μετρήσεις. Μετράμε το MI για κάθε λειτουργία και για κάθε αρχείο της Java και πιο συγκεκριμένα του JFs.

3.2 Λογισμικά αξιολόγησης και μέτρησης απόδοσης κώδικα

Δυστυχώς, δεν υπάρχει ένα ενιαίο εργαλείο για τη μέτρηση σε όλες τις παραμέτρους ποιότητας που χρειαζόμαστε και πολλά τέτοια εργαλεία δεν υποστηρίζουν την JavaScript γλώσσα ή δεν υποστηρίζουν μετρήσεις για τον κώδικα σε πολλαπλά αρχεία. Για να κάνουμε μετρήσεις ποιότητας, θα πρέπει να χρησιμοποιήσουμε διάφορα εργαλεία, όπως: το JSmeter (jsmeter.info), το Cloc (cloc.sourceforge.net) και το Understand (scitools.com). Ας αναφέρουμε μερικά λόγια για κάθε μία από τις παραπάνω εφαρμογές.

Το λογισμικό JsMeter είναι μια εφαρμογή που υπολογίζει μετρήσεις κώδικα JavaScript για προγράμματα και μεθόδους. Με αυτό το λογισμικό μπορούμε να πάρουμε μετρήσεις για: γραμμές κώδικα, γραμμές των σχολίων, % των γραμμών που περιέχουν σχόλια, τον αριθμό των δηλώσεων, ο αριθμός των Branches, η μονάδα Cyclomatic Complexity, το βάθος κώδικα, ο Halstead Δείκτης Πρόγραμμα επίπεδο / όγκο και δυνατότητα συντήρησης.

Το λογισμικό Cloc μετράει κώδικα, γραμμές σχόλια, παρατηρήσεις κενές γραμμές και φυσικές γραμμές πηγαιού κώδικα. Λαμβάνοντας υπόψη δύο εκδόσεις κώδικα, το Cloc μπορεί να μετρήσει τις αλλαγές στο κενό, σχόλια και τις γραμμές κώδικα. Είναι γραμμένο εξ ολοκλήρου σε Perl και είναι υπό την άδεια GNU. Χρησιμοποιήθηκε για να συλλέξει τις μετρήσεις μεγέθους.

Το λογισμικό Understand είναι ένα στατικό εργαλείο ανάλυσης για τη διατήρηση, τη μέτρηση και ανάλυση κώδικα JavaScript.

Στην παρούσα πτυχιακή εργασία χρησιμοποιήσαμε την τεχνική του slickspeed selector test. Πιο συγκεκριμένα «τρέξαμε» online με την βοήθεια browsers τα test και αποθηκεύαμε σε online αρχεία τα αποτελέσματα. Πιο συγκεκριμένα στα test είχαμε ορίσει κάποια selectors

και μετράγαμε για κάθε javascript framework το χρόνο που κάνει για την εκτέλεση του κάθε selector.

3.3 Μεθοδολογία μέτρησης απόδοσης JS Framework από κινητές συσκευές

Στην εργασία μας καλούμαστε να μετρήσουμε την απόδοση των JS Frameworks. Στα tests που εκτελέσαμε ελέγχαμε τον χρόνο που χρειάζεται κάθε javascript framework για να εκτελέσει κάθε DOM selector.

Όλα τα javascript πλαίσια περιέχουν ένα πλήθος από DOM στοιχεία (selectors) τα οποία βασίζονται αρκετά στις προδιαγραφές CSS και προσφέρουν ένα ισχυρό εργαλείο για την αντιστοίχιση στοιχείων σε μια ιστοσελίδα. Οι DOM selectors μπορεί να κυμαίνονται από απλά ονόματα στοιχείων έως σύνθετες παραστάσεις. Στον επόμενο πίνακα έχουμε ταξινομήσει τους selectors όπως περιγράφονται από την προδιαγραφή W3C για τους CSS selectors.

| Simple Selectors | | | | |
|----------------------------------|-----------------|----------------------------|---------------|------|
| Universal Selector | * | | | |
| Tag Selectors | tag | | | |
| ID Selectors | #id | tag#id | | |
| Class Selectors | .class | tag.class | | |
| Attribute Selectors | [att] | [att=val] | [att =val] | etc. |
| | tag[att] | tag[att=val] | tag[att =val] | etc. |
| Multiple Attribute Selectors | [att1][att2] | [att1=val1][att2=val2] | | etc. |
| | tag[att1][att2] | tag[att1=val1][att2 =val2] | | etc. |
| UI element States Pseudo-classes | :enabled | :disabled | :checked | |
| Structural Pseudo-classes | :root | :empty | | |
| | :nth-child(val) | :nth-of-type(val) | | etc. |
| The negation pseudo-class | :not(selector) | | | |
| Other pseudo-classes | :target | :lang(val) | | |
| Combinators | | | | |
| Descendant combinator | selector1 | selector2 | | |
| Child combinator | selector1 | > selector2 | | |
| Adjacent sibling combinator | selector1 | + selector2 | | |
| General sibling combinator | selector1 | ~ selector2 | | |
| Group of selectors | | | | |
| Multiple Selectors | Selector1, | selector2, | selector3 | ... |

Πίνακας 1 – Ταξινόμηση Selectors

Προκειμένου να μελετήσουμε την απόδοση των JS Framework συλλέξαμε κάποιες «έξυπνες» συσκευές και tablet προκειμένου να τις χρησιμοποιήσουμε για να τρέξουμε τα slickspeed selectors test. Οι συσκευές που χρησιμοποιήθηκαν είναι:

| |
|---|
| Κινητές συσκευές – smartphone -tablet: |
|---|

- Nokia Lumia 820, Win Mobile 8.0, Dual-core 1.5 GHz Krait with 1GB RAM
- Samsung GT-S6810P, Google Android,
- Iphone 5, iOS 7.0.4, Dual-core 1.3 GHz Swift (ARM v7-based) with 1GB RAM
- Zte s8q, Android , 1200 MHz, Quad - Core, 1 GB RAM.

Στην εργασία μας κάναμε μετρήσεις και από φορητούς υπολογιστές για να εξάγουμε συγκριτικά αποτελέσματα και συμπεράσματα. Συγκεκριμένα οι σταθεροί υπολογιστές είχαν λειτουργικό σύστημα Win XP και Win7.

Οι browsers που χρησιμοποιήθηκαν από κάθε συσκευή ήταν:

- Default browser for Android
- Default browser for iOS 7
- Internet Explorer 10
- Safari
- Fox
- Google Chrome
- Firefox 25.0
- Dolphin

Ενώ οι browsers που χρησιμοποιήθηκαν από τους φορητούς υπολογιστές είναι:

- Firefox 25.0
- Google Chrome
- Internet Explorer 11

Κατά την εκτέλεση των tests οι συσκευές μας ήταν πλήρως φορτισμένες, εκτελούσαμε μόνο το test και καμία άλλη εφαρμογή στο προσκήνιο και φροντίζαμε να μην κλείνει η οθόνη από την προφύλαξη οθόνης. Ο χρόνος εκτέλεσης κάθε test κυμαινόταν από 2 λεπτά έως και μισή ώρα για κάθε test. Αυτό συνέβαινε γιατί σε κάθε test ελέγχαμε διαφορετικό πλήθος selectors.

Τα τεστ που εκτελέσαμε από κάθε συσκευή, είτε πρόκειται για smartphone είτε για laptop, είναι τα παρακάτω:

- 1) <http://alife.hpclab.ceid.upatras.gr:100/slickspeed-big/>
- 2) <http://alife.hpclab.ceid.upatras.gr:100/slickspeed-med/>
- 3) <http://alife.hpclab.ceid.upatras.gr:100/slickspeed-small/>
- 4) <http://alife.hpclab.ceid.upatras.gr:100/slickspeed-perf-big/>
- 5) <http://alife.hpclab.ceid.upatras.gr:100/slickspeed-perf-med/>
- 6) <http://alife.hpclab.ceid.upatras.gr:100/slickspeed-perf-small/>

Τα τρία πρώτα τεστ υλοποιήθηκαν για μετρήσεις των διαφορετικών εκδόσεων JQuery. Ενώ τα υπόλοιπα 3 για τα Javascript Framework JQuery, MooTools, Extjs, YUI, Dojo, Prototype.

Τα τεστ που εκτελέσαμε από Iphone, είναι τα παρακάτω:

<http://jsperf.com/q-tests#run>

<http://jsperf.com/q-tests/2#run>

<http://jsperf.com/q-tests/3#run>

<http://jsperf.com/a-f-tests#run>

<http://jsperf.com/a-f-tests/2#run>

<http://jsperf.com/a-f-tests/3#run>

<http://jsperf.com/g-j-tests#run>

<http://jsperf.com/g-j-tests/2#run>

<http://jsperf.com/g-j-tests/3#run>

<http://jsperf.com/klmnop-tests#run>

<http://jsperf.com/klmnop-tests/2#run>

<http://jsperf.com/klmnop-tests/3#run>

3.4 Ανάλυση τρόπου λήψης μετρήσεων

Για να υλοποιήσουμε τις μετρήσεις αξιοποιήσαμε τις δυνατότητες του Slickspeed test. Είμαι μία τεχνική που μας βοηθάει να μετράμε την ταχύτητα των DOM selectors σε JFs. Με βάση την εφαρμογή Slickspeed χρησιμοποιήσαμε μια νέα βελτιωμένη έκδοση που ονομάζεται slick-speed_enhanced και παρέχει:

- Online αποθήκευση των αποτελεσμάτων στον server της εφαρμογής:

- Στο κάτω μέρος του τεστ υπάρχει ένα πλαίσιο κειμένου στο οποίο ο χρήστης μετά την εκτέλεση του τεστ συμπληρώνει το όνομα της συσκευής και τον τύπο, καθώς και τον περιηγητή που χρησιμοποίησε. Πατώντας το κουμπί της αποθήκευσης αποτελεσμάτων στον διακομιστή τότε αποστέλλονται τα δεδομένα και αποθηκεύονται σε αρχείο στον διακομιστή της εφαρμογής.
- Όλα τα αποτελέσματα αποθηκεύονται σε ένα αρχείο `showresult.php` που βρίσκεται σε αντίστοιχο φάκελο στον server.
- Σε κάθε σελίδα προβολής αποτελεσμάτων υπάρχει και ένα κουμπί “Toggle found” το οποίο κρύβει ανεπιθύμητα αποτελέσματα ή μη χρήσιμα αποτελέσματα για τους DOM selectors και επιτρέπει στον χρήστη να αντιγράψει τα αποτελέσματα σαν αριθμούς.
- Μεγαλύτερο και οργανωμένο σύνολο από 263 selectors.
- Τρεις διαφορετικές υλοποιήσεις του slickspeed. Με αυτές τις τρεις εκδοχές οι χρήστες μπορούν να εκτελέσουν 3 διαφορετικά αρχεία όπου το καθένα περιέχει διαφορετικό αριθμό DOM στοιχείων (140, 192 και 2068).
- Τέλος, το τεστ μπορεί να μετρήσει όχι μόνο τον χρόνο για ένα DOM selector αλλά και για μία JF εντολή ή κομμάτι κώδικα. Χρησιμοποιήσαμε αυτό το εργαλείο για την σύγκριση της απόδοσης των 49 jQuery επεκτάσεων Selectors με 35 ισοδύναμες εντολές jQuery χρησιμοποιώντας λειτουργίες jQuery.

Συνολικά εκτελέσαμε **88** test από όλες τις συσκευές. Στο επόμενο κεφάλαιο παρουσιάζονται τα αποτελέσματα των μετρήσεων.

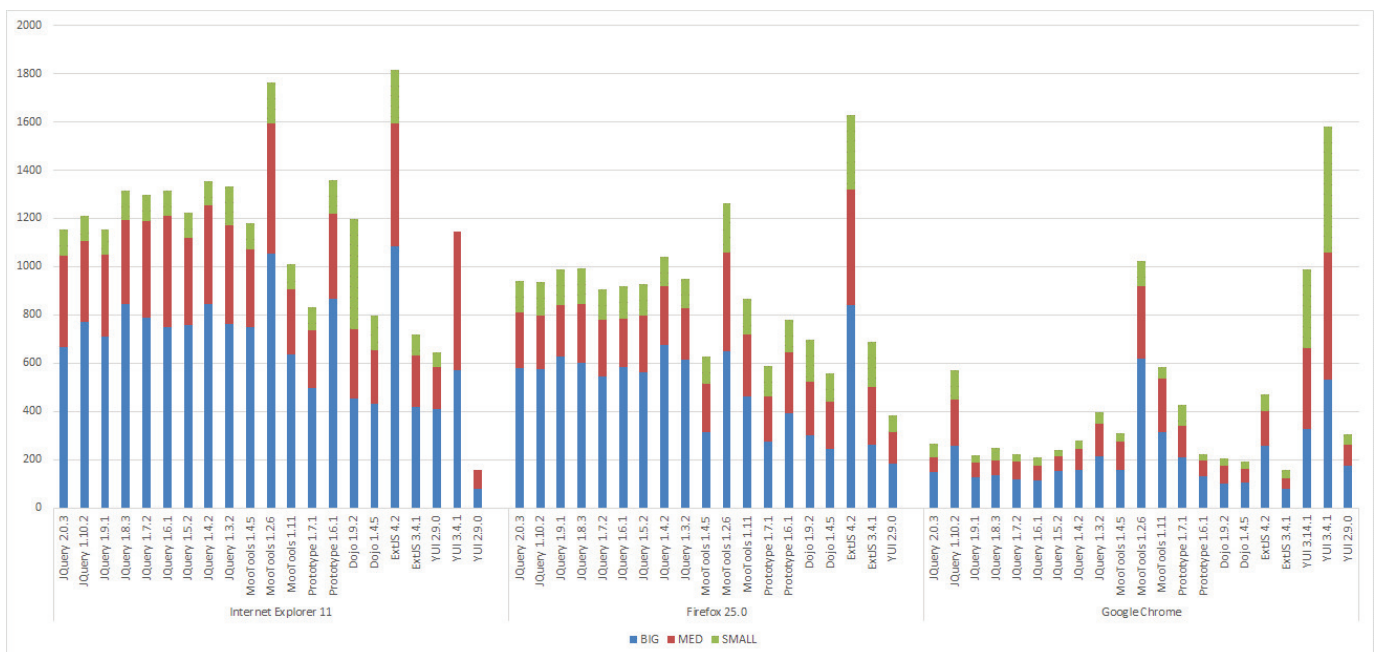
Κεφάλαιο 4 – Αποτελέσματα Μετρήσεων

Στο παρόν κεφάλαιο θα παραθέσουμε πλήθος γραφημάτων για τα αποτελέσματα των μετρήσεων που υλοποιήσαμε. Στο αντίστοιχο CD της εργασίας υπάρχουν αναλυτικά όλα τα αρχεία μετρήσεων.

4.1 Μετρήσεις από Φορητό Υπολογιστή

Η μέτρηση απόδοσης των JSFrameworks από φορητό υπολογιστή αφορά υπολογιστές με λογισμικό Windows XP, Windows 7 και Windows 8.1. Οι περιηγητές που χρησιμοποιήθηκαν είναι Mozilla Firefox 25.0, Google Chrome και Internet Explorer 11.

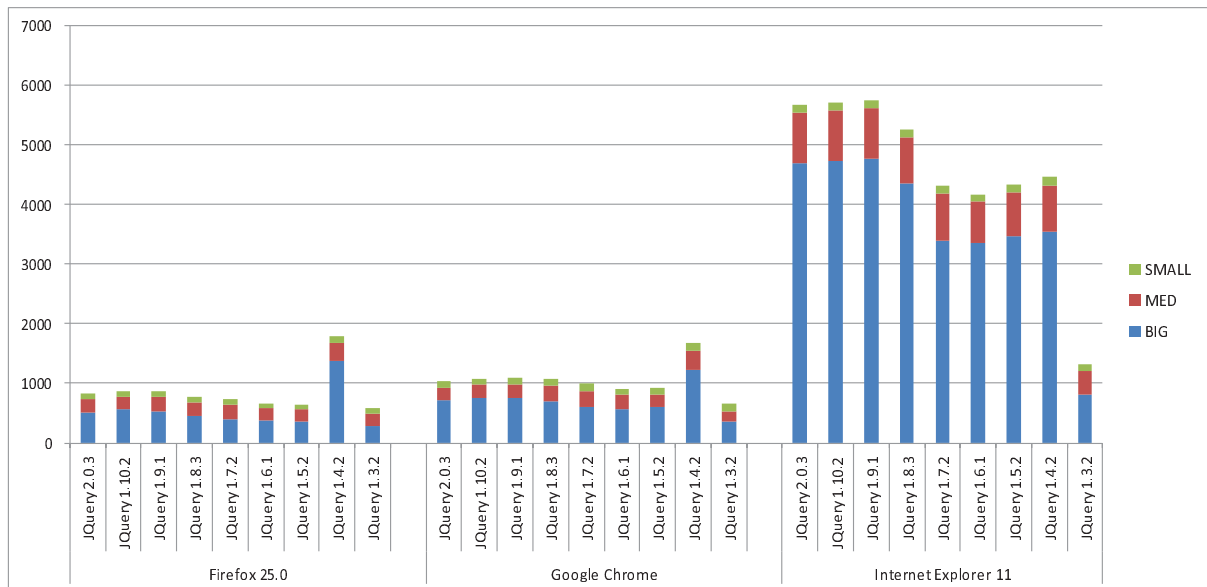
Στο πρώτο γράφημα απεικονίζουμε τους χρόνους εκτέλεσης των JS Frameworks που αναλύσαμε σε προηγούμενο κεφάλαιο, ενώ στο 2ο γράφημα απεικονίζουμε τους χρόνους εκτέλεσης των διαφορετικών εκδόσεων JQuery Framework.



Γράφημα 1 - Μετρήσεις JFs από PC

Παρατηρούμε ότι όσο νεότερες είναι οι εκδόσεις των JFs τόσο λιγότερος είναι ο χρόνος προσπέλασης τους. Με μόνη εξαίρεση το πλαίσιο YUI 3.4.1 και ExtJS 4.2. Επίσης η εκτέλεση από τον προεπιλεγμένο browser των Windows Internet Explorer είναι πιο αργή σε σχέση με τον Mozilla Firefox, που και αυτός σε σύγκριση με την εκτέλεση του Google Chrome είναι αρκετά αργός.

Η μικρότερη τιμή και για τις εκτελέσεις των 3 τεστ είναι η τιμή 180ms. Ενώ η μεγαλύτερη τιμή είναι 1800ms. Σε αυτό το σημείο θα προσθέσουμε ότι οι τιμές έχουν προκύψει από την μέση τιμή εκτέλεσης των τεστ, διότι κάθε τεστ το «τρέξαμε» παραπάνω από μία φορά.



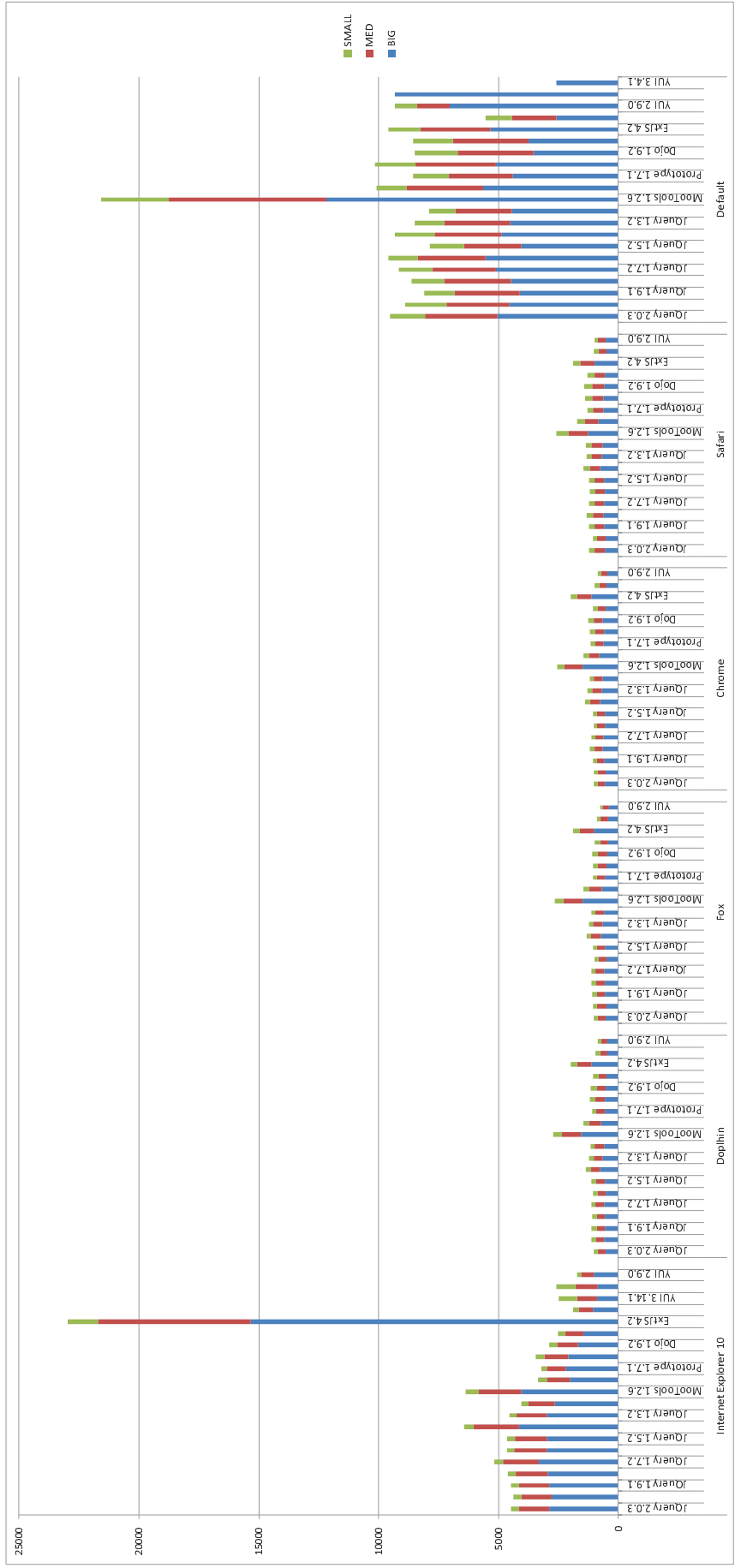
Γράφημα 2 - Μετρήσεις JQuery εκδόσεων από PC

Τα αποτελέσματα των τεστ από σταθερούς υπολογιστές για τα JQuery frameworks είναι λίγο διαφορετικά από το προηγούμενο γράφημα. Εδώ οι τιμές είναι αρκετά υψηλές και βλέπουμε ότι το Mozilla Firefox αποδίδει καλύτερα όσο αφορά τον χρόνο εκτέλεσης, με δεύτερο τον Google Chrome και τελευταίο τον Internet Explorer. Εδώ η ελάχιστη τιμή είναι περίπου 500ms και η μέγιστη 5700ms.

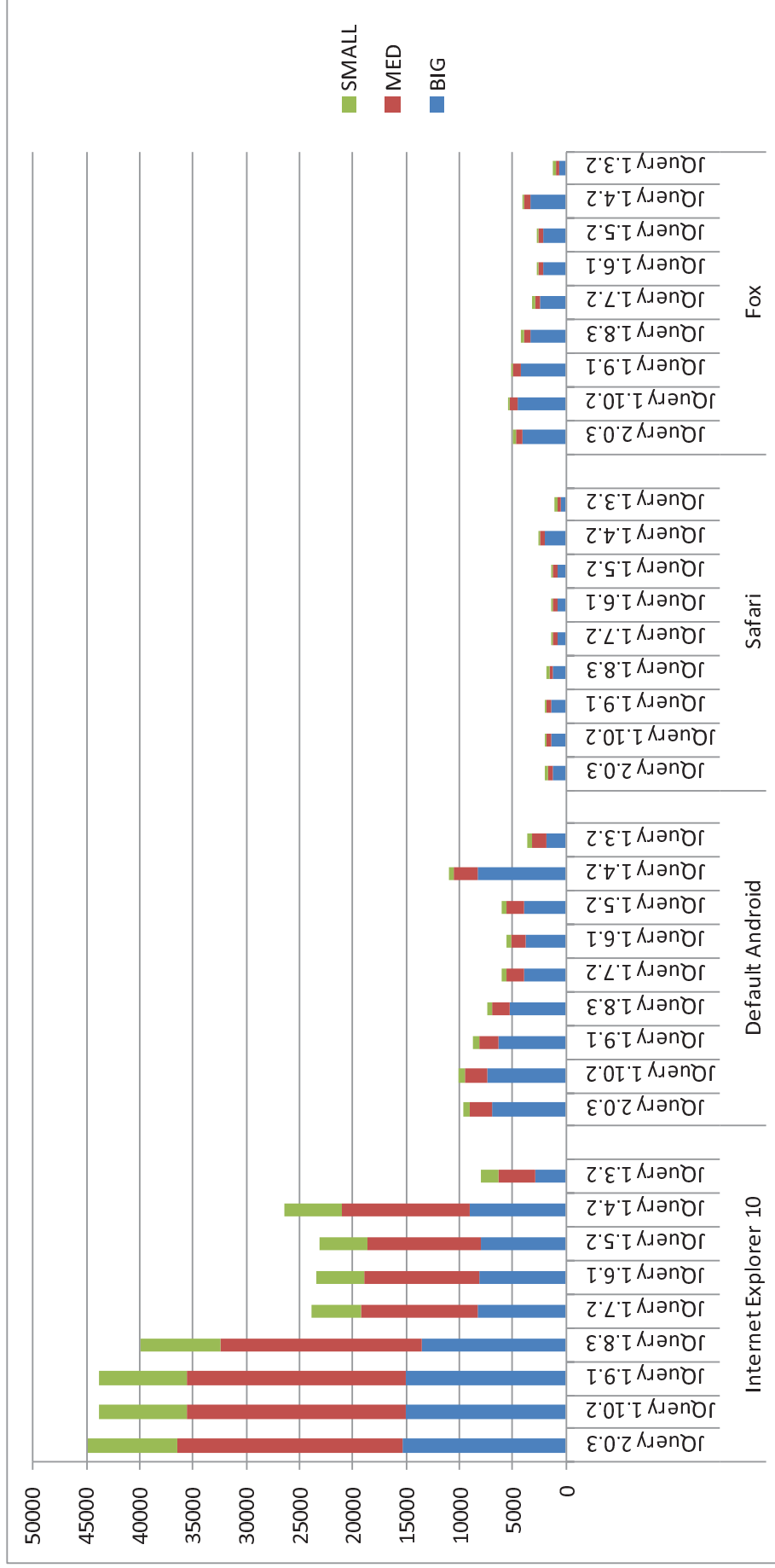
4.2 Μετρήσεις από Smartphones

Η μέτρηση απόδοσης των JS Frameworks από κινητές συσκευές αφορά smartphone με λογισμικό Android 4.1, Win Mobile 8.0 και iOS 7.0.4. Οι περιηγητές που χρησιμοποιήθηκαν είναι Default browser, Dolphin, Fox, Safari και Internet Explorer 10.

Στο πρώτο γράφημα απεικονίζουμε τους χρόνους εκτέλεσης των JS Frameworks που αναλύσαμε σε προηγούμενο κεφάλαιο, ενώ στο 2ο γράφημα απεικονίζουμε τους χρόνους εκτέλεσης των διαφορετικών εκδόσεων JQuery Framework.



Γράφημα 3 - Μετρήσεις JFS από Smartphones



Γράφημα 4 - Μετρήσεις JQuery εκδόσεων από Smartphone

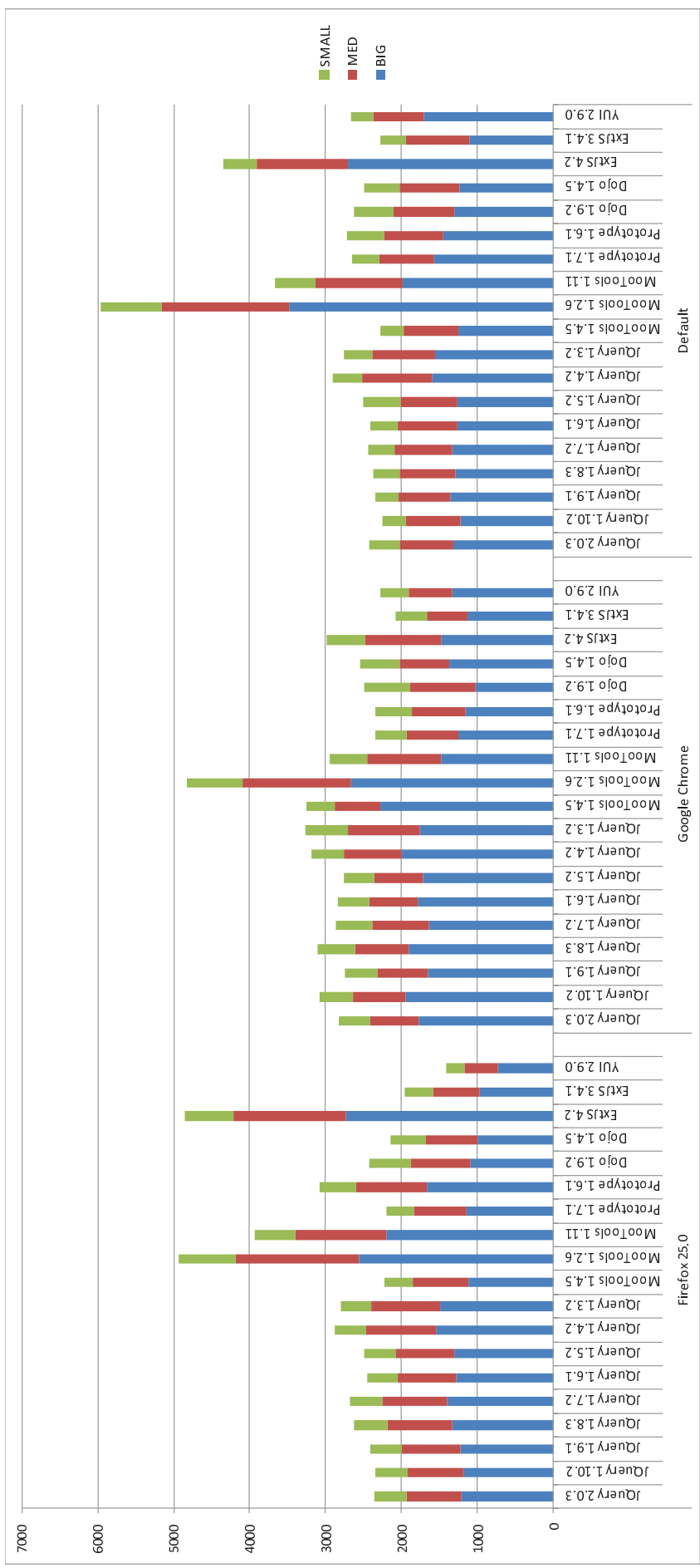
Μελετώντας τα παραπάνω γραφήματα βλέπουμε ότι οι χρόνοι εκτέλεσης είναι αρκετά μεγαλύτεροι σε σχέση με τους χρόνους από τις μετρήσεις των σταθερών υπολογιστών. Μάλιστα οι χρόνοι διαφέρουν και αρκετά μεταξύ τους ανάλογα με τους περιηγητές που έχουμε χρησιμοποιήσει. Οπότε με βάση αυτές τις παρατηρήσεις καταλήγουμε στα παρακάτω:

1. Οι περιηγητές Dolphin, Fox, Safari και Google Chrome έχουν τους καλύτερους χρόνους σε σχέση με τον Internet Explorer και τον Default Android Browser που έχουν οι περισσότερες κινητές συσκευές.
2. Αν συγκρίνουμε τις τιμές του Google Chrome από τις κινητές συσκευές με τον Google Chrome από τους σταθερούς υπολογιστές οι χρόνοι είναι σχεδόν τριπλάσιοι, γεγονός που αποδεικνύει ότι τα JSF θέλουν ακόμα αρκετή δουλειά για να αποδίδουν εξίσου καλά με την εκτέλεσή τους μέσω Ηλεκτρονικών Υπολογιστών.
3. Οι καινούργιες εκδόσεις των JSFs αποδίδουν καλύτερα σε σχέση με τις παλαιότερες εκδόσεις ανεξαρτήτου περιηγητή ιστού.

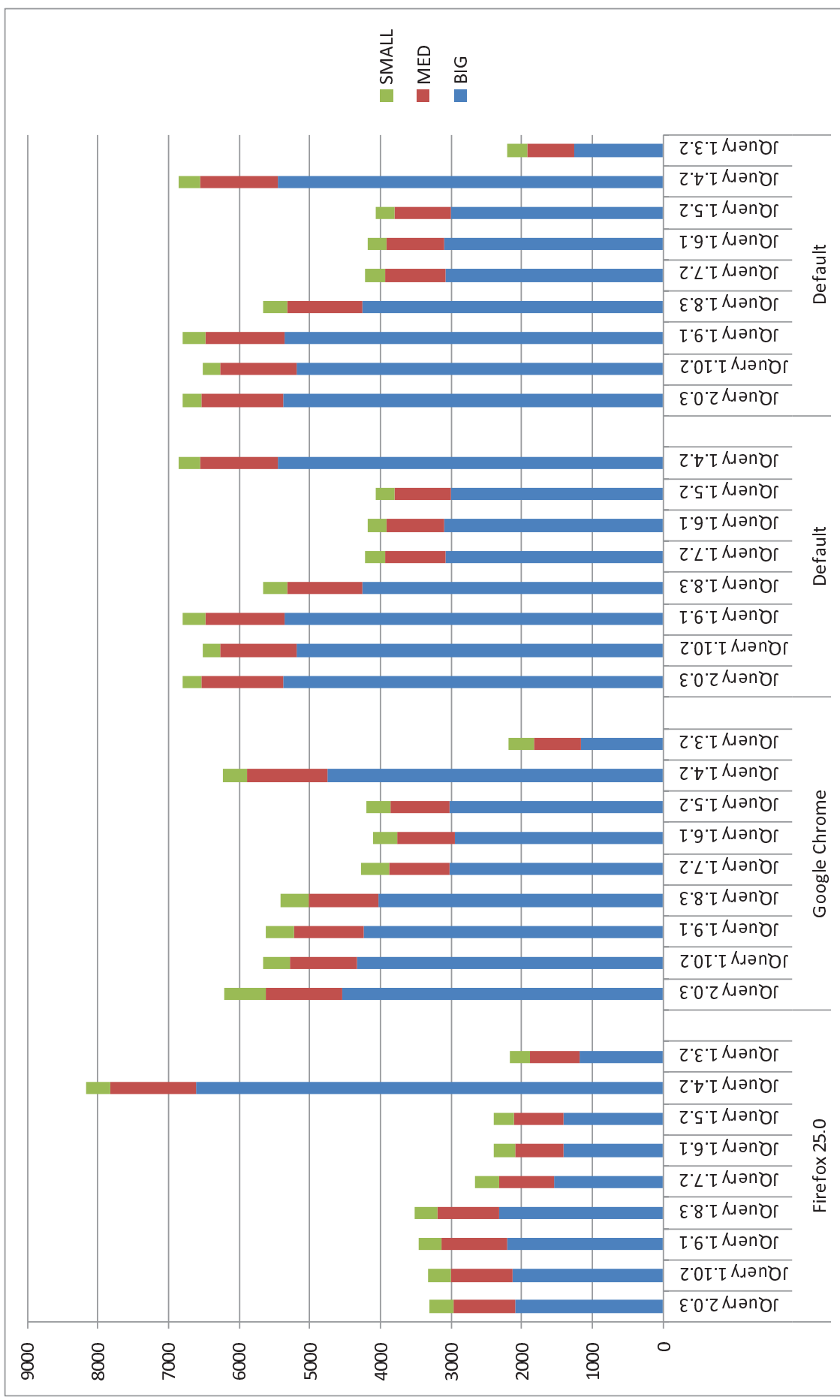
4.3 Μετρήσεις από Tablet

Η μέτρηση απόδοσης των JSFrameworks από tablet με λογισμικό Android 4.1 και τους περιηγητές Default browser for Android, Google Chrome και Mozilla Firefox 25.0.

Στο πρώτο γράφημα απεικονίζουμε τους χρόνους εκτέλεσης των JS Frameworks που αναλύσαμε σε προηγούμενο κεφάλαιο, ενώ στο 2ο γράφημα απεικονίζουμε τους χρόνους εκτέλεσης των διαφορετικών εκδόσεων JQuery Framework.



Γράφημα 5 - Μετρήσεις JSFs από Tablet



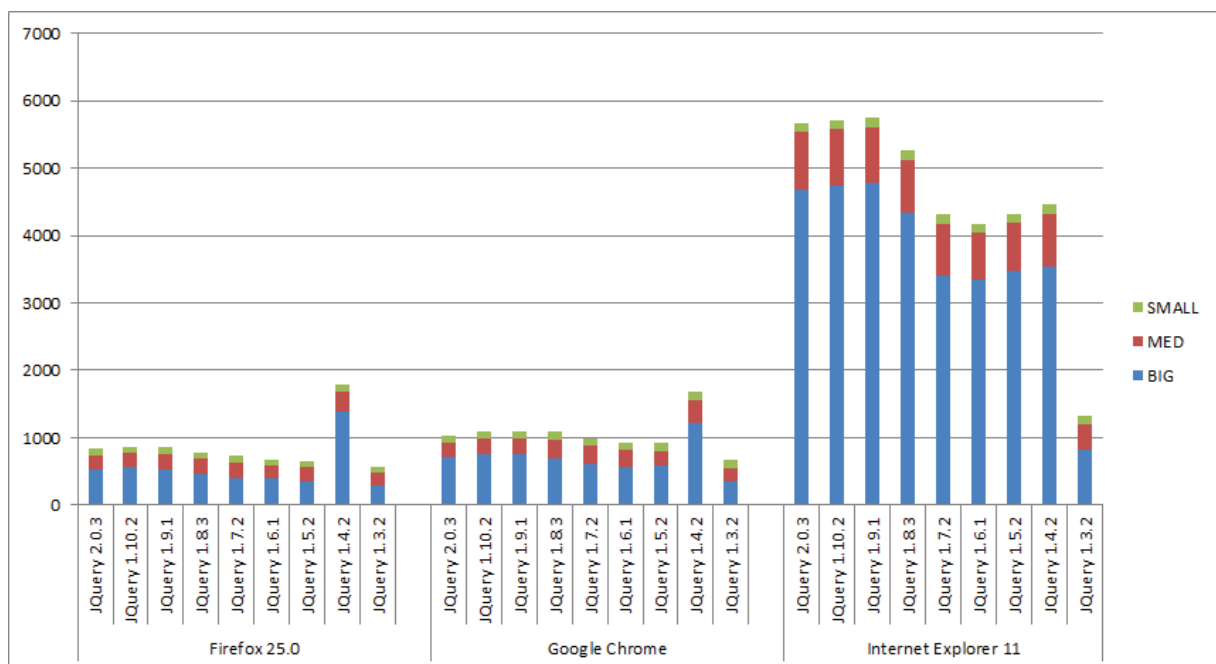
Γράφημα 6 - Μετρήσεις jQuery εκδόσεων από Tablet

Βλέποντας τα γραφήματα και των τριών κατηγοριών βλέπουμε ότι οι τιμές που λάβαμε από τις μετρήσεις που γίνανε από tablet είναι ενδιάμεσες των μετρήσεων από smartphone και σταθερών υπολογιστών. Γεγονός που μας δείχνει ότι όσο πιο κοντά είναι μία συσκευή με την έννοια του υπολογιστή ή όσο πιο πολλές κοινές εργασίες με έναν υπολογιστή εκτελεί μία συσκευή τόσο καλύτερες αποδόσεις έχουμε από τα JSFs.

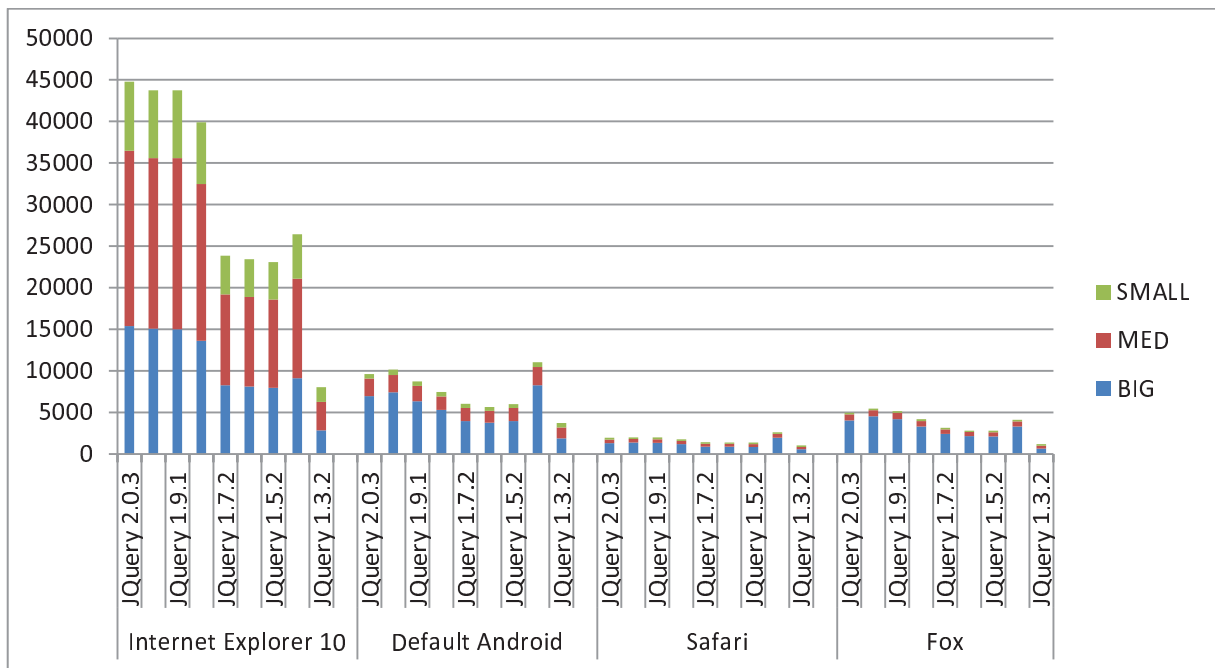
4.4 Επιπλέον Αποτελέσματα Έρευνας (1)

Στις επόμενες σελίδες παραθέτουμε αναλυτικά τα αποτελέσματα που πήραμε εκτελώντας τα test μέτρησης απόδοσης των javascript framework. Πιο συγκεκριμένα συγκρίνουμε τις αποδόσεις των διαφορετικών jQuery εκδόσεων και από τις 3 κατηγορίες συσκευών. Στη συνέχεια συγκρίνουμε αποδόσεις συγκεκριμένων DOM selectors των jQuery frameworks.

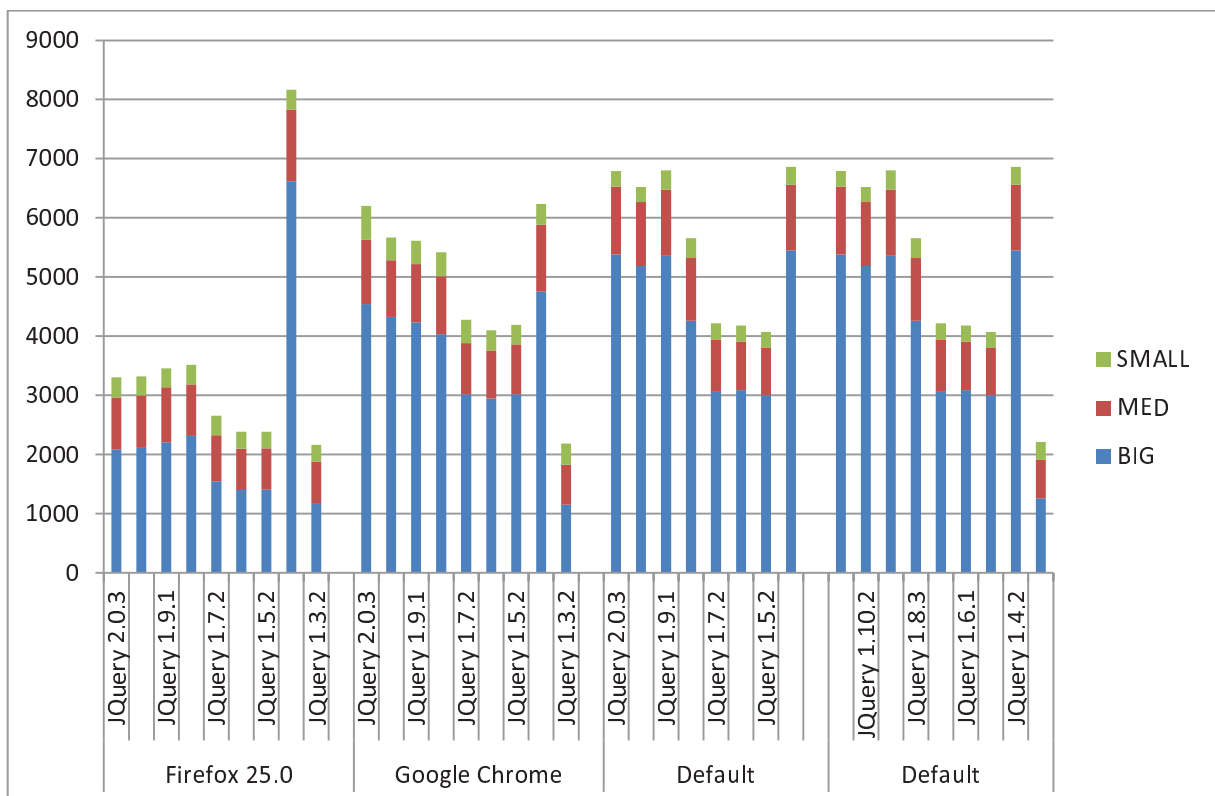
Στο παρακάτω γράφημα απεικονίζουμε τα αποτελέσματα από την εκτέλεση 3 διαφορετικών slickspeed tests και τα έχουμε οργανώσει σε big, medium & small.



Γράφημα 7 - Μετρήσεις από Laptop & PC



Γράφημα 8 - Μετρήσεις από Smartphone Android & iPhone



Γράφημα 9 - Μετρήσεις από tablet Android

Παρατηρώντας τα γραφήματα βλέπουμε:

- ότι οι χρόνοι εκτέλεσης είναι μικρότεροι όταν τα εκτελούμε από σταθερούς υπολογιστές και tablet και ειδικά όταν ο browser είναι ο Firefox Mozilla και ο Google Chrome.

- στην ίδια κατηγορία συσκευών παρατηρούμε τους υψηλότερους χρόνους εκτέλεσης όταν ο περιηγητής που χρησιμοποιούμε είναι ο Internet Explorer που είναι ο προεπιλεγμένος για τους σταθερούς υπολογιστές και laptop ή ο προεπιλεγμένος του συστήματος Android από τα tablet.
- στην ίδια κατηγορία εκτέλεσης των jquery frameworks παρατηρούμε ότι όσο πιο παλιά είναι η έκδοση του jquery framework τόσο περισσότερο χρόνο εκτέλεσης απαιτεί, ενώ όσο πιο καινούργια έκδοση είναι το jquery framework τόσο λιγότερο χρόνο εκτέλεσης χρειάζεται. Αυτό παρατηρείται πάλι όταν οι περιηγητές είναι ο Mozilla Firefox & Google Chrome.
- στους προεπιλεγμένους περιηγητές, είτε αναφερόμαστε σε desktop, laptop είτε tablet παρατηρείτε το αντίστροφο, δηλ όσο πιο καινούργια είναι η έκδοση τόσο περισσότερο χρόνο εκτέλεσης χρειάζεται, από ότι στις παλιές εκδόσεις.

Στην κατηγορία έξυπνων κινητών τηλεφώνων τα αποτελέσματα είναι λίγο διαφορετικά. Καταρχάς σε αυτήν την κατηγορία έχουμε να ξεχωρίσουμε τις συσκευές με Android λειτουργικό και τις συσκευές με iOS λειτουργικό.

- Στην πρώτη κατηγορία παρατηρούμε ότι οι χρόνοι εκτέλεσης είναι μεγαλύτεροι στις καινούργιες εκδόσεις των jquery frameworks με μία εξαίρεση στην έκδοση 1.4.2 όπου ο χρόνος εκτέλεσης είναι αρκετά μεγαλύτερος από την προγενέστερη έκδοση 1.3.2.
- Στην δεύτερη κατηγορία συσκευών παρατηρούμε τους πιο μικρούς χρόνους εκτέλεσης οι οποίοι ακολουθούν ακριβώς την ίδια πορεία με τις συσκευές Android.

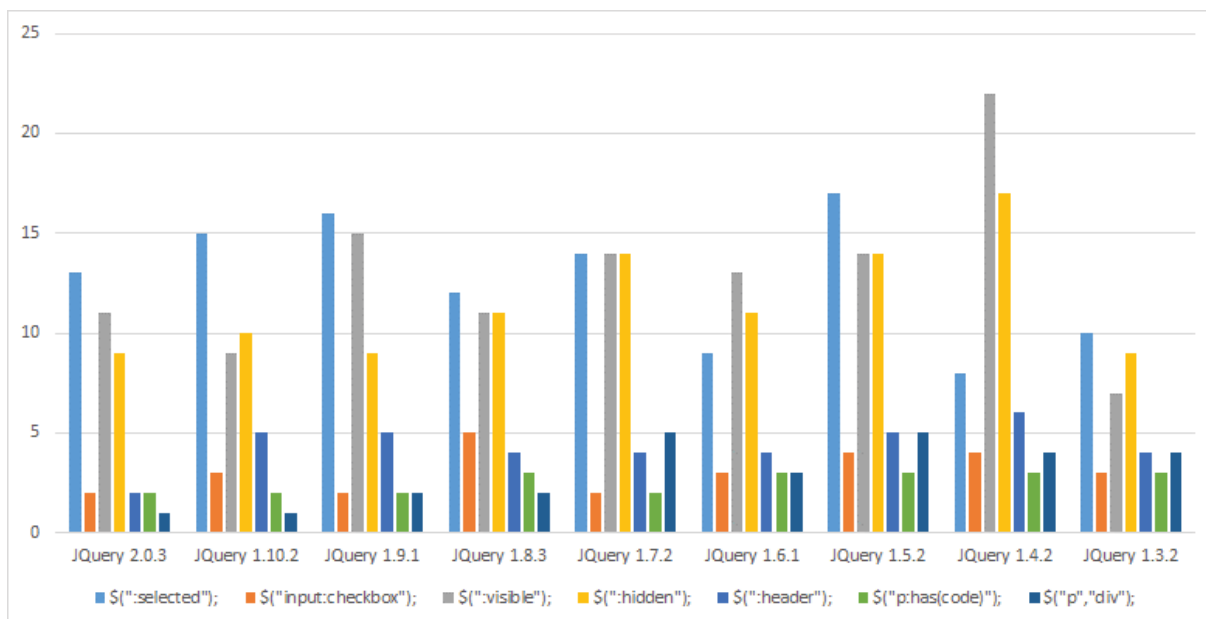
Στην ίδια κατηγορία μετρήσεων επιλέξαμε κάποιους βασικούς selectors των jquery frameworks και μετρήσαμε τον χρόνο εκτέλεσής τους και σε έναν πίνακα παραθέτουμε τις μέσες τιμές τους.

4.4.1 Μετρήσεις από PC/Laptop

| selectors | JQuery 2.0.3 | JQuery 1.10.2 | JQuery 1.9.1 | JQuery 1.8.3 | JQuery 1.7.2 | JQuery 1.6.1 | JQuery 1.5.2 | JQuery 1.4.2 | JQuery 1.3.2 |
|------------------------------------|--------------|---------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| <code>\$(":selected");</code> | 13 | 15 | 16 | 12 | 14 | 9 | 17 | 8 | 10 |
| <code>\$("input:checkbox");</code> | 2 | 3 | 2 | 5 | 2 | 3 | 4 | 4 | 3 |
| <code>\$(":visible");</code> | 11 | 9 | 15 | 11 | 14 | 13 | 14 | 22 | 7 |
| <code>\$(":hidden");</code> | 9 | 10 | 9 | 11 | 14 | 11 | 14 | 17 | 9 |
| <code>\$(":header");</code> | 2 | 5 | 5 | 4 | 4 | 4 | 5 | 6 | 4 |
| <code>\$("p:has(code)");</code> | 2 | 2 | 2 | 3 | 2 | 3 | 3 | 3 | 3 |
| <code>\$("p","div");</code> | 1 | 1 | 2 | 2 | 5 | 3 | 5 | 4 | 4 |

Πίνακας 3 - Μετρήσεις απόδοσης Selectors από PC

Με κόκκινο έχουμε επισημάνει την μέγιστη τιμή ενώ με πράσινο την ελάχιστη τιμή. Η παρατήρηση μας είναι ότι οι μέγιστη τιμή αντιστοιχεί σε αρκετά παλαιότερη βιβλιοθήκη, ενώ οι ελάχιστες σε νεότερες εκδόσεις. Παρακάτω βλέπουμε και το αντίστοιχο γράφημα των παραπάνω μετρήσεων.



Γράφημα 10 - Μετρήσεις απόδοσης Selectors από PC

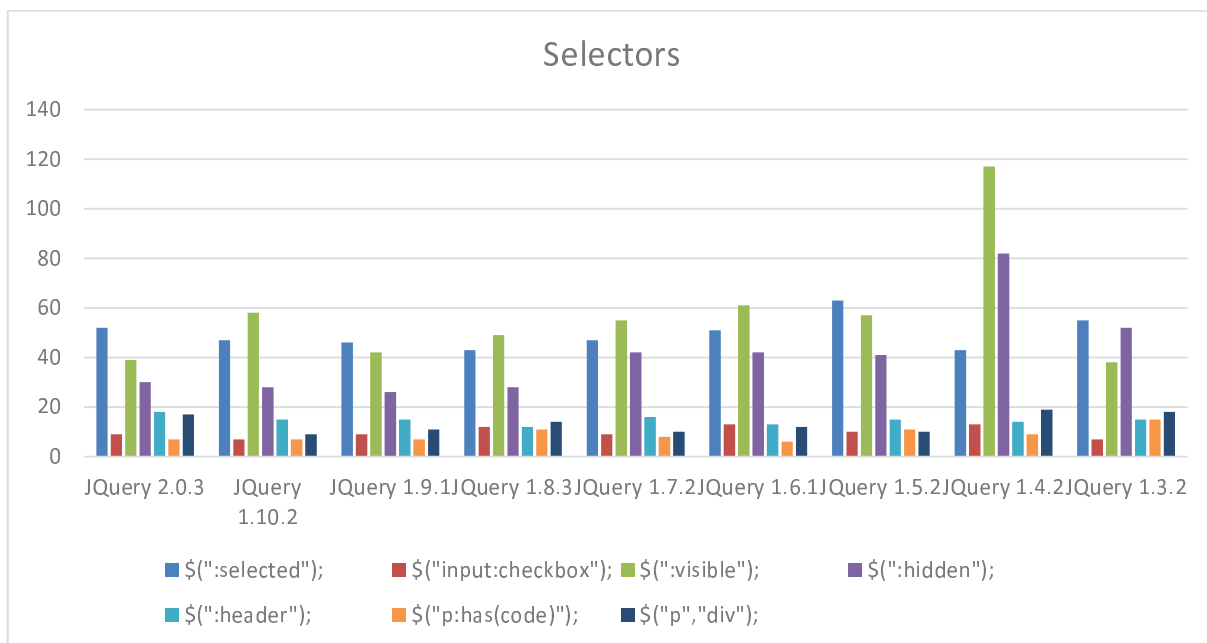
4.4.2 Μετρήσεις από Smartphone

Επιλέξαμε τους ίδιους selectors και για τα Smartphone και τα αποτελέσματα φαίνονται παρακάτω.

| selectors | jQuery 2.0.3 | jQuery 1.10.2 | jQuery 1.9.1 | jQuery 1.8.3 | jQuery 1.7.2 | jQuery 1.6.1 | jQuery 1.5.2 | jQuery 1.4.2 | jQuery 1.3.2 |
|----------------------------|--------------|---------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| \$(":selected"); | 52 | 47 | 46 | 43 | 47 | 51 | 63 | 43 | 55 |
| \$("input:checkbox"); | 9 | 7 | 9 | 12 | 9 | 13 | 10 | 13 | 7 |
| \$(":visible"); | 39 | 58 | 42 | 49 | 55 | 61 | 57 | 117 | 38 |
| \$(":hidden"); | 30 | 28 | 26 | 28 | 42 | 42 | 41 | 82 | 52 |
| \$(":header"); | 18 | 15 | 15 | 12 | 16 | 13 | 15 | 14 | 15 |
| \$("p:has(code)"); | 7 | 7 | 7 | 11 | 8 | 6 | 11 | 9 | 15 |
| \$("p","div"); | 17 | 9 | 11 | 14 | 10 | 12 | 10 | 19 | 18 |

Γράφημα 11 - Μετρήσεις απόδοσης Selectors από Smartphone

Αντίστοιχα και στις συσκευές smartphone η μέγιστη τιμή εντοπίζεται σε παλαιότερη έκδοση ενώ οι χαμηλότερη σε νεότερη βιβλιοθήκη. Μάλιστα η μικρότερη τιμή έχει να κάνει με τον selector της παραγράφου (.p). Παρακάτω ακολουθεί το αντίστοιχο γράφημα.



Γράφημα 12 - Μετρήσεις απόδοσης Selectors από Smartphone

4.4.3 Μετρήσεις από Tablet

Επιλέξαμε τους ίδιους selectors και για τις συσκευές Tablet και τα αποτελέσματα φαίνονται παρακάτω.

| selectors | JQuery 2.0.3 | JQuery 1.10.2 | JQuery 1.9.1 | JQuery 1.8.3 | JQuery 1.7.2 | JQuery 1.6.1 | JQuery 1.5.2 | JQuery 1.4.2 | JQuery 1.3.2 |
|------------------------------|--------------|---------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| \$(":selected"); | 45 | 40 | 41 | 46 | 42 | 44 | 42 | 36 | 48 |
| \$("input:checkbox"); | 6 | 5 | 9 | 9 | 5 | 6 | 5 | 7 | 6 |
| \$(":visible"); | 17 | 20 | 28 | 22 | 20 | 24 | 23 | 47 | 16 |
| \$(":hidden"); | 15 | 13 | 12 | 11 | 10 | 13 | 13 | 27 | 15 |
| \$(":header"); | 13 | 14 | 12 | 17 | 18 | 12 | 16 | 12 | 17 |
| \$("p:has(code)"); | 12 | 8 | 8 | 6 | 9 | 6 | 6 | 12 | 9 |
| \$("p","div"); | 10 | 10 | 13 | 6 | 8 | 16 | 9 | 19 | 24 |

Πίνακας 4- Μετρήσεις απόδοσης Selectors από Tablet

Το ίδιο σκηνικό συμβαίνει και στις μετρήσεις από Tablet. Παρακάτω ακολουθεί το αντίστοιχο του γράφημα.



Γράφημα 13 - Μετρήσεις απόδοσης Selectors από Tablet

Παρατηρώντας τις μετρήσεις ή τα γραφήματα παρατηρούμε ότι οι συγκεκριμένοι selectors που επιλέξαμε εκτελούνται αρκετά πιο σύντομα από σταθερούς υπολογιστές, ύστερα από tablet και απαιτούν τον μεγαλύτερο χρόνο από τα smartphones.

Αξίζει να σχολιάσουμε ότι γενικά τα tablet είναι δεύτερες συσκευές όσο αφορά το χρόνο εκτέλεσης, η μέγιστη τιμή έπαιτε των σταθερών υπολογιστών όπως και η ελάχιστη. Άρα στους JFs selectors που επιλέξαμε βλέπουμε ότι οι χρόνοι ακολουθούν τα αρχικά μας γραφήματα, που περιγράφουν γενικά την εκτέλεση διάφορων Javascript Frameworks, όπου καλύτερους χρόνους απόδοσης έχουν οι σταθεροί υπολογιστές, έπειτα τα tablet και τέλος τα smartphones. Γεγονός που αποδεικνύει ότι χρειάζονται βελτίωση τα JQuery frameworks για να αποδίδουν εξίσου καλά και γρήγορα και στις «έξυπνες» κινητές συσκευές.

4.5 Επιπλέον Αποτελέσματα Έρευνας (2)

Στο προηγούμενο κεφάλαιο εκτελέσαμε τα slickspeed test για διαφορετικές εκδόσεις της jQuery βιβλιοθήκης όπως και για διάφορους selectors. Στο παρόν κεφάλαιο θα εκτελέσουμε το slickspeed test για τα Javascript Frameworks που εκτελέσαμε και στην ενότητα 4.1. Τα Selectors που επιλέξαμε τώρα είναι:

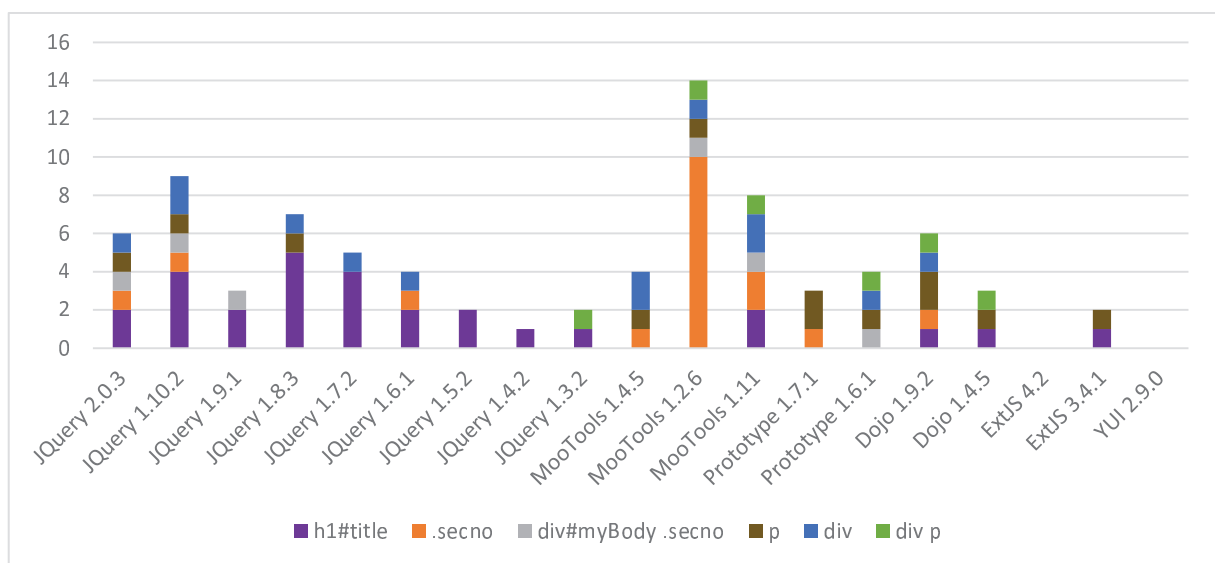
| selectors |
|--------------------------|
| h1#title |
| .secno |
| div#myBody .secno |
| p |
| div |
| div p |

4.5.1. Μετρήσεις από PC/Laptop

Οι αντίστοιχες μετρήσεις απεικονίζονται στον παρακάτω πίνακα και γράφημα:

| selectors | jQuery 2.0.3 | jQuery 1.10.2 | jQuery 1.9.1 | jQuery 1.8.3 | jQuery 1.7.2 | jQuery 1.6.1 | jQuery 1.5.2 | jQuery 1.4.2 | jQuery 1.3.2 | MooTools 1.4.5 | MooTools 1.2.6 | MooTools 1.11 | Prototype 1.7.1 | Prototype 1.6.1 | Dojo 1.9.2 | Dojo 1.4.5 | ExtJS 4.2 | ExtJS 3.4.1 | YUI 2.9.0 |
|--------------------------|--------------|---------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|----------------|----------------|---------------|-----------------|-----------------|------------|------------|-----------|-------------|-----------|
| h1#title | 2 | 4 | 2 | 5 | 4 | 2 | 2 | 1 | 1 | 0 | 0 | 2 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| .secno | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 10 | 2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| div#myBody .secno | | | | | | | | | | | | | | | | | | | |
| p | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| div | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 1 | 2 | 1 | 0 | 1 | 0 |
| div p | 1 | 2 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 2 | 1 | 2 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| div p | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |

Πίνακας 5 - Μετρήσεις απόδοσης Selectors από PC



Γράφημα 14 - Μετρήσεις απόδοσης Selectors από PC

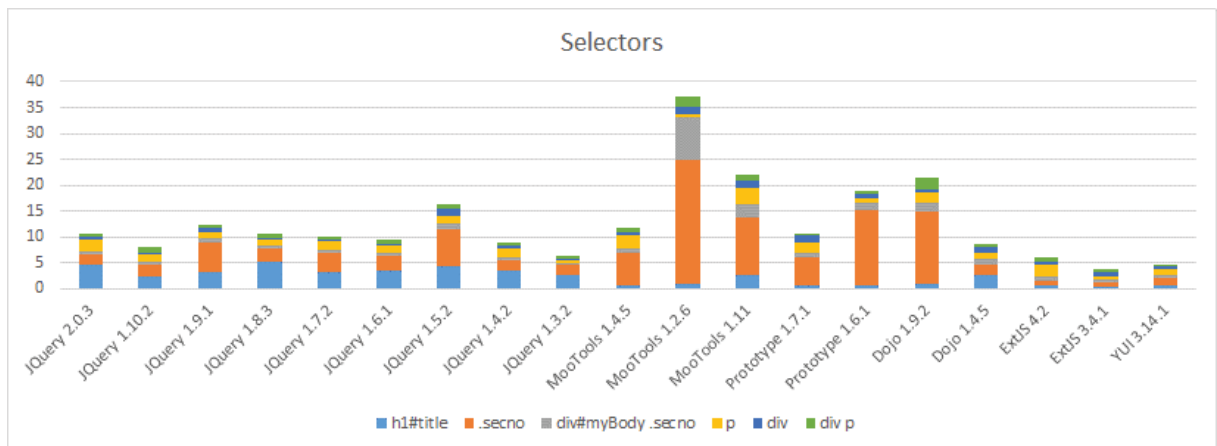
Από το γράφημα παρατηρούμε ότι ο selector «**h1#title**» από όλα σχεδόν τα πλαίσια εκτελείται. Ο υψηλότερος χρόνος προσπέλασης είναι στην έκδοση jQuery 1.9.1. Επίσης βλέπουμε ότι το στοιχείο αυτό δεν παρουσιάζει την αναμενόμενη γραμμή, δηλ μικρότεροι χρόνοι σε νέες εκδόσεις και μεγαλύτεροι σε παλαιότερες, αλλά κάνει κυματισμούς με αυξομειώσεις.

4.5.2 Μετρήσεις από Smartphone

Οι αντίστοιχες μετρήσεις απεικονίζονται στον παρακάτω πίνακα και γράφημα:

| selectors | jQuery 2.0.3 | jQuery 1.10.2 | jQuery 1.9.1 | jQuery 1.8.3 | jQuery 1.7.2 | jQuery 1.6.1 | jQuery 1.5.2 | jQuery 1.4.2 | jQuery 1.3.2 | MooTools s 1.4.5 | MooTools s 1.2.6 | MooTools s 1.11 | Prototype e 1.7.1 | Prototype e 1.6.1 | Dojo 1.9.2 | Dojo 1.4.5 | ExtJS 4.2 | ExtJS 3.4.1 | YUI 3.14.1 |
|-------------------|--------------|---------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|------------------|------------------|-----------------|-------------------|-------------------|------------|------------|-----------|-------------|------------|
| h1#title | 4,75 | 2,25 | 3,25 | 5,25 | 3,25 | 3,5 | 4,25 | 3,5 | 2,5 | 0,75 | 1 | 2,5 | 0,5 | 0,5 | 1 | 2,5 | 0,75 | 0,25 | 0,75 |
| .sceno | 2 | 2,5 | 5,75 | 2,5 | 3,75 | 2,75 | 7,25 | 2 | 2 | 6,25 | 23,75 | 11,25 | 5,5 | 14,75 | 14 | 2 | 0,75 | 1 | 1,25 |
| div#myBody .sceno | 0,5 | 0,5 | 0,75 | 0,5 | 0,5 | 0,75 | 1 | 0,5 | 0,5 | 0,75 | 8,25 | 2,5 | 1 | 1,25 | 1,5 | 1,25 | 0,75 | 0,5 | 0,75 |
| p | 2,25 | 1,5 | 1,25 | 1,25 | 1,75 | 1,25 | 1,5 | 1,75 | 0,5 | 2,5 | 0,75 | 3,25 | 2 | 1 | 2 | 1,25 | 2,5 | 0,5 | 1 |
| div | 0,5 | 0,25 | 0,75 | 0,25 | 0,25 | 0,5 | 1,5 | 0,5 | 0,25 | 0,75 | 1,5 | 1,25 | 1,25 | 0,75 | 0,75 | 1 | 0,5 | 1 | 0,5 |
| div p | 0,75 | 1 | 0,5 | 1 | 0,5 | 0,75 | 0,75 | 0,75 | 0,5 | 0,75 | 1,75 | 1,25 | 0,5 | 0,75 | 2,25 | 0,75 | 0,75 | 0,5 | 0,5 |

Πίνακας 6 - Μετρήσεις απόδοσης Selectors από Smartphone



Γράφημα 15 - Μετρήσεις απόδοσης Selectors από Smartphone

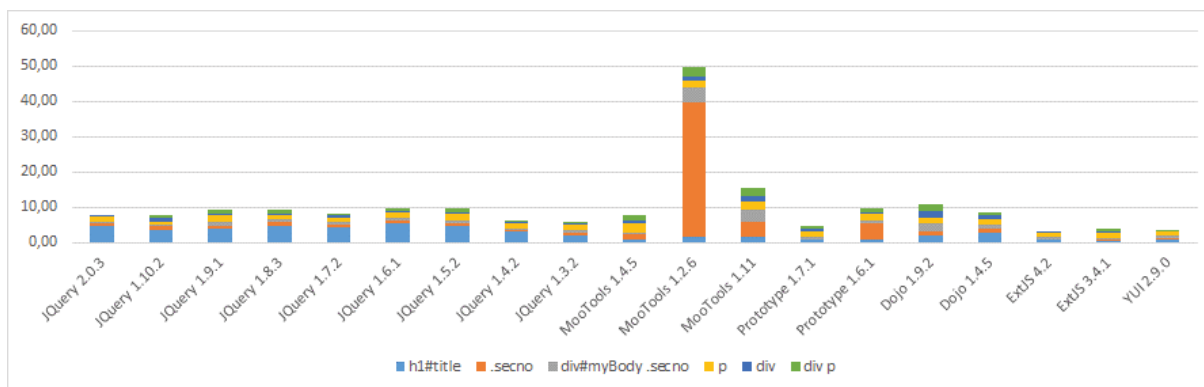
Από το γράφημα παρατηρούμε ότι ο selector «**h1#title**» από όλα σχεδόν τα πλαίσια εκτελείται. Ο υψηλότερος χρόνος προσπέλασης είναι στην έκδοση **jQuery 1.8.3**. Επίσης βλέπουμε ότι το στοιχείο αυτό δεν παρουσιάζει την αναμενόμενη γραμμή, δηλ μικρότεροι χρόνοι σε νέες εκδόσεις και μεγαλύτεροι σε παλαιότερες, αλλά κάνει κυματισμούς με αυξομειώσεις. Η μεγαλύτερη τιμή παρουσιάζεται στον selector “**.sceno**” και στο πλαίσιο **MooTools 1.2.6**. Αυτό που διακρίνουμε ξεκάθαρα είναι οι κατά πολύ μεγαλύτεροι χρόνοι εκτέλεσης των αντίστοιχων selectors από τις έξυπνες συσκευές.

4.5.3 Μετρήσεις από Tablet

Οι αντίστοιχες μετρήσεις απεικονίζονται στον παρακάτω πίνακα και γράφημα:

| selectors | jQuery 2.0.3 | jQuery 1.10.2 | jQuery 1.9.1 | jQuery 1.8.3 | jQuery 1.7.2 | jQuery 1.6.1 | jQuery 1.5.2 | jQuery 1.4.2 | jQuery 1.3.2 | MooTools s 1.4.5 | MooTools s 1.2.6 | MooTools s 1.11 | Prototype e 1.7.1 | Prototype e 1.6.1 | Dojo 1.9.2 | Dojo 1.4.5 | ExtJS 4.2 | ExtJS 3.4.1 | YUI 2.9.0 |
|-------------------|--------------|---------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|------------------|------------------|-----------------|-------------------|-------------------|------------|------------|-----------|-------------|-----------|
| h1#title | 4,67 | 3,67 | 4,00 | 4,67 | 4,33 | 5,33 | 4,67 | 3,00 | 2,00 | 0,67 | 1,67 | 1,67 | 0,67 | 1,00 | 2,00 | 2,67 | 0,67 | 0,33 | 0,67 |
| .sceno | 0,67 | 1,00 | 0,67 | 1,00 | 0,67 | 1,00 | 0,67 | 0,67 | 0,67 | 1,67 | 38,00 | 4,00 | 0,33 | 4,33 | 1,00 | 1,33 | 0,33 | 0,67 | 0,67 |
| div#myBody .sceno | 0,67 | 0,33 | 1,33 | 1,00 | 0,67 | 0,67 | 1,00 | 0,33 | 1,00 | 0,33 | 4,33 | 3,67 | 0,67 | 1,00 | 2,33 | 1,00 | 0,67 | 0,33 | 0,67 |
| p | 1,33 | 1,00 | 1,67 | 1,00 | 1,33 | 1,67 | 1,67 | 1,33 | 1,33 | 2,67 | 1,67 | 2,33 | 1,67 | 1,67 | 1,67 | 1,67 | 1,00 | 1,33 | 1,00 |
| div | 0,33 | 1,00 | 0,67 | 0,67 | 0,67 | 0,33 | 0,67 | 0,67 | 0,33 | 1,00 | 1,33 | 1,67 | 0,67 | 0,67 | 2,00 | 1,00 | 0,33 | 0,67 | 0,33 |
| div p | 0,00 | 0,67 | 1,00 | 1,00 | 0,67 | 0,67 | 1,00 | 0,33 | 0,33 | 1,33 | 2,67 | 2,00 | 0,67 | 1,00 | 1,67 | 1,00 | 0,00 | 0,67 | 0,33 |

Πίνακας 7 - Μετρήσεις απόδοσης Selectors από Tablet



Γράφημα 16 - Μετρήσεις απόδοσης Selectors από Tablet

Από το γράφημα παρατηρούμε ότι ο selector «**h1#title**» από όλα σχεδόν τα πλαίσια εκτελείται. Ο υψηλότερος χρόνος προσπέλασης είναι στην έκδοση **jQuery 1.6.1**. Η μεγαλύτερη τιμή παρουσιάζεται στον selector “**.sceno**” και στο πλαίσιο **MooTools 1.2.6**. Αυτό που διακρίνουμε ξεκάθαρα είναι οι ενδιάμεσες τιμές στους χρόνους εκτέλεσης των αντίστοιχων selectors από τους υπολογιστές και τις έξυπνες συσκευές.

Ακόμα και από τις μετρήσεις συγκεκριμένων selectors τα γραφήματά μας δείχναν την ίδια εικόνα: Οι μεγαλύτεροι χρόνοι είναι εκτέλεσης πραγματοποιήθηκαν από τα smartphone και οι μικρότεροι από τους σταθερούς υπολογιστές. Και οι συσκευές tablet είναι πάλι στις ενδιάμεσες τιμές.

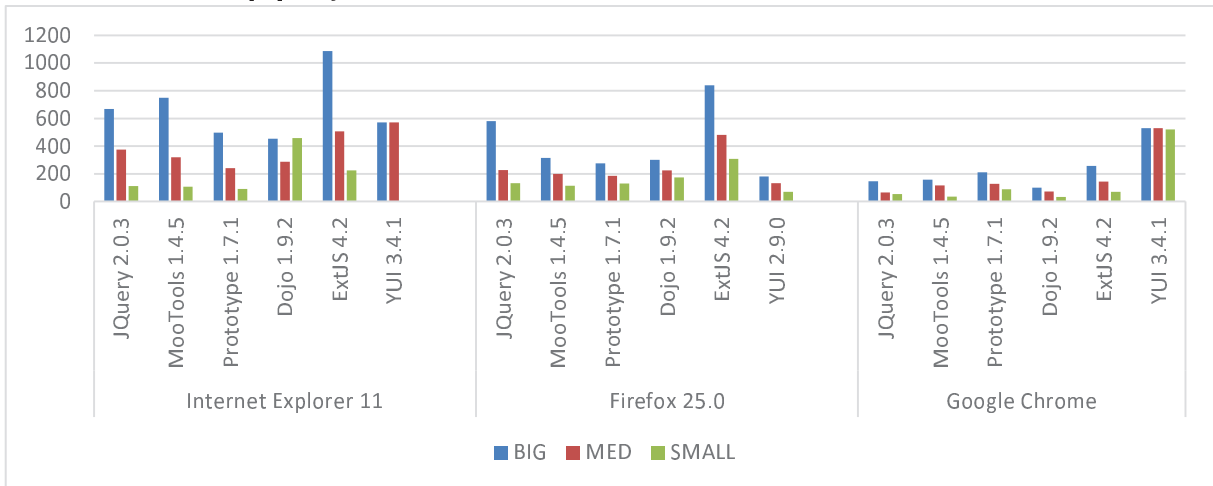
4.6 Επιπλέον Αποτελέσματα Έρευνας (3)

Για να μπορέσουμε να αναλύσουμε καλύτερα τα αποτελέσματα δημιουργήσαμε ένα καινούργιο υπολογιστικό βιβλίο όπου επιλέξαμε τις μετρήσεις από τα JFs με τις πιο πρόσφατες εκδόσεις, και έτσι έχουμε:

| Javascript Frameworks |
|------------------------|
| jQuery 2.0.3 |
| MooTools 1.4.5 |
| Prototype 1.7.1 |
| Dojo 1.9.2 |
| ExtJS 4.2 |
| YUI 2.9.0 |

Για τα συγκεκριμένα Frameworks πήραμε τις μετρήσεις και από τα τρία τεστ Big, Med & Small. Παρακάτω ακολουθούν τα γραφήματα ανά συσκευές.

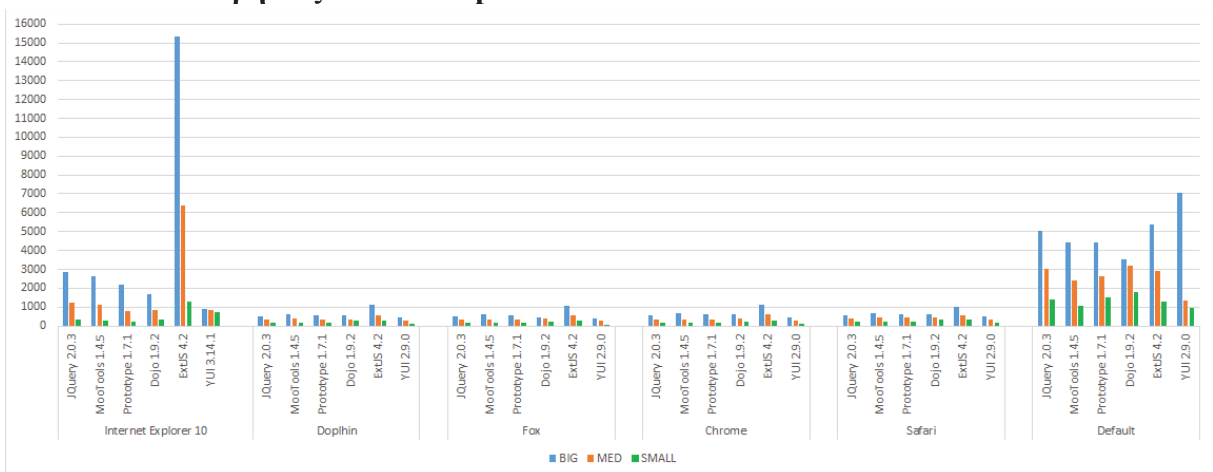
4.6.1 Μετρήσεις από PC/Laptop



Γράφημα 17 - Μετρήσεις νεότερων εκδόσεων JFs από PC/Laptop

Παρατηρούμε ότι οι μεγαλύτεροι χρόνοι ανήκουν στον Internet Explorer και οι μικρότερες στον Google Chrome. Κάτι το οποίο συνέβαινε και στις προηγούμενες μετρήσεις. Επίσης το πλαίσιο ExtJS παρουσιάζει τις υψηλότερες τιμές, εκτός από τον περιηγητή της Google.

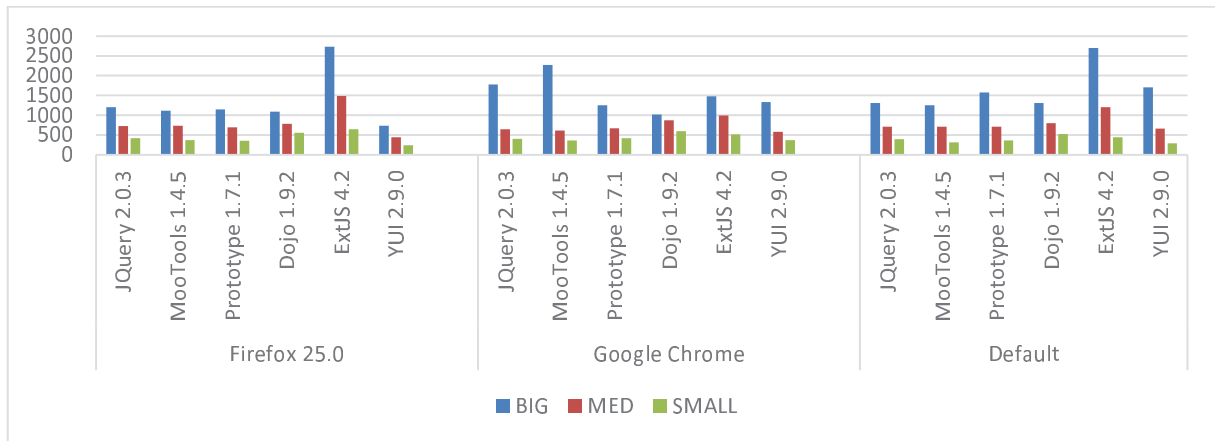
4.6.2 Μετρήσεις από Smartphone



Γράφημα 18 - Μετρήσεις νεότερων εκδόσεων JFs από Smartphone

Στις συσκευές smartphone παρατηρούμε ότι οι μεγαλύτεροι χρόνοι παρατηρούνται στον προεπιλεγμένο περιηγητή που έχει η κάθε συσκευή και έπειτα στον Internet Explorer. Ενώ αρκετά μικρότεροι είναι οι χρόνοι στους περιηγητές της συσκευής iPhone. Το πλαίσιο με τον μεγαλύτερο χρόνο είναι το ExtJS 4.2 στον περιηγητή Internet Explorer.

4.6.3 Μετρήσεις από Tablet



Γράφημα 19 - Μετρήσεις νεότερων εκδόσεων JFs από Tablet

Στις συσκευές tablet παρατηρούμε ότι το πλαίσιο με τον μεγαλύτερο χρόνο είναι το ExtJS 4.2 στον προκαθορισμένο περιηγητή της συσκευής και στον Mozilla Firefox 25.0. περιηγητή Internet Explorer.

Κεφάλαιο 5 – Συμπεράσματα

Μέσα από την πτυχιακή μας εργασία μελετήσαμε την ανάπτυξη της Java και κυρίως των Javascript Frameworks. Η κύρια ασχολία μας ήταν η ανάπτυξη μεθοδολογίας μέτρησης της απόδοσης των Javascript Frameworks, όπως και η συλλογή και αξιολόγηση αποτελεσμάτων. Μέσα από την έρευνά μας δεν μπορούμε να καταλήξουμε ότι το καλύτερο Javascript Framework είναι κάποιο συγκεκριμένο ή ότι κάποιο Javascript Framework αποδίδει καλύτερα από κάποιο άλλο, διότι αυτό εξαρτάται από τα χαρακτηριστικά της κάθε συσκευής, τους περιηγητές που εκτελούνται καθώς και από το τι άλλο μπορεί να εκτελείται στην συσκευή μας ταυτόχρονα, διότι οι μετρήσεις μας γίνανε στην ιδανική περίπτωση όπου όλες οι συσκευές εκτελούσαν μόνο το test που είχαμε δημιουργήσει.

Τα συμπεράσματά μας από αυτή την έρευνα θεωρούμε ότι είναι ένα καλό εργαλείο ή βάση για να χρησιμοποιηθεί από προγραμματιστές όσο αφορά την απόδοση των DOM selectors και των Javascript Frameworks. Επιπλέον με την εκτέλεση συγκεκριμένων δοκιμών από διάφορες συσκευές και προγράμματα περιήγησης, διακρίνονται τα μειονεκτήματα και τα πλεονεκτήματα χρήσης συγκεκριμένων πλαισίων. Επίσης μέσα από τα συμπεράσματα και τα γραφήματα οι προγραμματιστές έχουν ένα εργαλείο να εντοπίσουν τις καλές πρακτικές των JSFs.

Πιο συγκεκριμένα τα τελικά μας συμπεράσματα περιλαμβάνουν:

1. Οι νεότερες εκδόσεις Javascript Frameworks κατά μέσο όρο, αποδίδουν καλύτερα σε σχέση με παλαιότερες εκδόσεις.
2. Τα Javascript Frameworks αποδίδουν καλύτερα στους σταθερούς και φορητούς υπολογιστές, στη συνέχεια στις συσκευές tablet και τέλος στις έξυπνες συσκευές.
3. Μέσα από τις μετρήσεις του κάθε τεστ που υπάρχουν στα αρχεία Excel που συνοδεύουν την πτυχιακή μας εργασία, μπορούμε να δούμε τους selectors που αποδίδουν καλύτερα και αυτούς που δεν αποδίδουν. Έτσι μέσα από αυτές τις μετρήσεις μπορεί να επιλέξουν οι προγραμματιστές τα στοιχεία εκείνα που πρέπει να αναπτυχθούν καλύτερα για να βελτιωθούν.
4. Όταν οι προγραμματιστές δημιουργούν κάποιο πρόγραμμα για smartphone είναι καλύτερο να επιλέγουν την jQuery, διότι αποδίδει αρκετά καλύτερα σε σχέση με άλλα πλαίσια. Επίσης το πλαίσιο που χρήζει βελτίωσης για προγραμματισμό σε smartphone είναι το ExtJS 4.2., έτσι ώστε να μπορεί να γίνει πιο αποδοτικό σε σχέση με τα υπόλοιπα πλαίσια.
5. Αν μπορούσαμε να προτείνουμε τις πλατφόρμες περιήγησης στο διαδίκτυο, δεν θα προτείναμε τον Internet Explorer το οποίο δυστυχώς έχει τις χειρότερες αποδόσεις. Άρα η πρέπει να ξαναληφθεί υπόψη ο σχεδιασμός τους ή να βελτιωθούν τα Javascript Frameworks έτσι ώστε να αποδίδουν εξίσου καλά και στον Internet Explorer.
6. Αξιοσημείωτο είναι το γεγονός ότι όλα τα Javascript Frameworks αποδίδουν καλά αν εκτελούνται από συσκευές iPhone και από οποιονδήποτε περιηγητή τους.

Τέλος αυτό που θα προτείναμε για μελλοντική εργασία, έτσι ώστε οι προγραμματιστές να έχουν πάντα ένα εργαλείο αξιολόγησης απόδοσης των Javascript Frameworks είναι να υπάρχει μία online Βάση δεδομένων, όπου θα διατηρούνται τα αποτελέσματα από τις επιδόσεις των Javascript Frameworks αλλά και των DOM Selectors, έτσι ώστε να βελτιώνεται και η διαδικασία δημιουργίας προγραμμάτων και web applications. Αυτό το προτείνουμε διότι η διαδικασία εκτέλεσης κάθε test και συλλογής αποτελεσμάτων για περαιτέρω αξιολόγηση, υπήρξε αρκετά πολύπλοκη και χρονοβόρα. Δεδομένου ότι η χρήση εφαρμογών μέσα από όλες τις συσκευές που χρησιμοποιούν διαδίκτυο αυξάνεται όλο και περισσότερο, πρέπει να υπάρχει ο μηχανισμός εκείνος που θα μας δίνει πιο εύκολα τις αποδόσεις των συσκευών.

Βιβλιογραφία

1. <http://el.wikipedia.org/wiki/Java>
2. <http://el.wikipedia.org/wiki/JavaScript>
3. <http://www.htmlgoodies.com/beyond/javascript/article.php/3470971/Java-vs-JavaScript.htm>
4. <http://jquery.com/>
5. <http://en.wikipedia.org/wiki/JQuery>
6. <http://jquery.com/browser-support/>
7. <http://www.idesigner.gr/%CE%B5%CE%B9%CF%83%CE%B1%CE%B3%CF%89%CE%B3%CE%AE-%CF%83%CF%84%CE%B7%CE%BD-jquery/>
8. <http://www.ituts.gr/%CE%BE%CE%B5%CE%BA%CE%B9%CE%BD%CF%8E%CE%BD%CF%84%CE%B1%CF%82-%CE%BC%CE%B5-%CF%84%CE%B7%CE%BD-jquery>
9. <http://jquerymobile.com/demos/1.2.0/docs/about/platforms.html>
10. http://en.wikipedia.org/wiki/Dojo_Toolkit
11. http://dojotoolkit.org/documentation/tutorials/1.8/hello_dojo/
12. <http://livedocs.dojotoolkit.org/dojox/mobile/SimpleDialog>
13. http://en.wikipedia.org/wiki/Ext_JS
14. <http://www.sencha.com/products/extjs/>
15. <http://en.wikipedia.org/wiki/MooTools>
16. <http://mootools.net/>
17. http://en.wikipedia.org/wiki/Prototype_JavaScript_Framework
18. http://en.wikipedia.org/wiki/YUI_Library
19. <http://yuilibrary.com/>

Ευρετήριο Εικόνων

| | |
|---|----|
| Εικόνα 1 – Αναζήτηση πληροφοριών για βασικές Javascript βιβλιοθήκες | 28 |
| Εικόνα 2 - Δημοφιλή Javascript Frameworks..... | 29 |

Ευρετήριο Πινάκων

| | |
|---|----|
| Πίνακας 1 - Εκδόσεις JQuery..... | 14 |
| Πίνακας 2 - Συμβατότητα Frameworks με Browsers | 15 |
| Πίνακας 3 - Μετρήσεις απόδοσης Selectors από PC | 45 |
| Πίνακας 4- Μετρήσεις απόδοσης Selectors από Tablet..... | 47 |
| Πίνακας 5 - Μετρήσεις απόδοσης Selectors από PC | 49 |
| Πίνακας 6 - Μετρήσεις απόδοσης Selectors από Smartphone | 50 |
| Πίνακας 7 - Μετρήσεις απόδοσης Selectors από Tablet | 50 |

Ευρετήριο Γραφημάτων

| | |
|---|----|
| Γράφημα 1 - Μετρήσεις JFs από PC..... | 36 |
| Γράφημα 2 - Μετρήσεις JQuery εκδόσεων από PC | 37 |
| Γράφημα 3 - Μετρήσεις JFS από Smartphones | 38 |
| Γράφημα 4 - Μετρήσεις JQuery εκδόσεων από Smartphone | 39 |
| Γράφημα 5 - Μετρήσεις JSFs από Tablet | 41 |
| Γράφημα 6 - Μετρήσεις jQuery εκδόσεων από Tablet | 42 |
| Γράφημα 7 - Μετρήσεις από Laptop & PC..... | 43 |
| Γράφημα 8 - Μετρήσεις από Smartphone Android & iPhone | 44 |
| Γράφημα 9 - Μετρήσεις από tablet Android | 44 |
| Γράφημα 10 - Μετρήσεις απόδοσης Selectors από PC | 46 |
| Γράφημα 11 - Μετρήσεις απόδοσης Selectors από Smartphone | 46 |
| Γράφημα 12 - Μετρήσεις απόδοσης Selectors από Smartphone | 47 |
| Γράφημα 13 - Μετρήσεις απόδοσης Selectors από Tablet | 48 |
| Γράφημα 14 - Μετρήσεις απόδοσης Selectors από PC | 49 |
| Γράφημα 15 - Μετρήσεις απόδοσης Selectors από Smartphone | 50 |
| Γράφημα 16 - Μετρήσεις απόδοσης Selectors από Tablet | 51 |
| Γράφημα 17 - Μετρήσεις νεότερων εκδόσεων JFs από PC/Laptop | 52 |
| Γράφημα 18 - Μετρήσεις νεότερων εκδόσεων JFs από Smartphone | 52 |
| Γράφημα 19 - Μετρήσεις νεότερων εκδόσεων JFs από Tablet | 53 |