

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη εφαρμογής για την μετατροπή Αγγλικών χαρακτήρων σε  
Ελληνικούς

Παπαπέτρου Αθανάσης Π2017149

Επιβλέπων: Αναπληρωτής Καθηγητής Φοίβος Μυλωνάς

Απρίλιος 2022

## Επιβλέπων Καθηγητής

Αναπληρωτής Καθηγητής Φοίβος Μυλωνάς, ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

## Τριμελής Επιτροπή

Αναπληρωτής Καθηγητής Φοίβος Μυλωνάς, ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Επίκουρος Καθηγητής Θέμης Έξαρχος, ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Επίκουρος Καθηγητής Άρης Βραχάτης, ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



## ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας πτυχιακής εργασίας είναι η Ανάπτυξη εφαρμογής για την μετατροπή Αγγλικών χαρακτήρων σε Ελληνικούς. Για τον λόγο αυτόν, παρουσιάζονται εκτενώς στα επόμενα κεφάλαια της πτυχιακής εργασίας για το πως δημιούργησα ένα macro στο Word και ένα extension στο Chrome και στο Mozilla Firefox. Ο σκοπός του προγράμματος είναι να μετατρέπει τους Αγγλικούς χαρακτήρες σε Ελληνικά με το πάτημα ενός κουμπιού.

# Περιεχόμενα

|                                                                       |                           |
|-----------------------------------------------------------------------|---------------------------|
| ΕΞΩΦΥΛΛΟ                                                              | <a href="#"><u>1</u></a>  |
| ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ                                                     | <a href="#"><u>1</u></a>  |
| ΠΕΡΗΛΙΨΗ                                                              | <a href="#"><u>3</u></a>  |
| ΠΕΡΙΕΧΟΜΕΝΑ                                                           | <a href="#"><u>4</u></a>  |
| ΚΕΦΑΛΑΙΟ 1ο - Εισαγωγή                                                | <a href="#"><u>5</u></a>  |
| 1.1 Θεωρητικά στοιχεία για String Matching                            | <a href="#"><u>5</u></a>  |
| 1.2 Visual Basic                                                      | <a href="#"><u>7</u></a>  |
| 1.3 JSON                                                              | <a href="#"><u>8</u></a>  |
| 1.4 JavaScript                                                        | <a href="#"><u>9</u></a>  |
| 1.5 CSS                                                               | <a href="#"><u>10</u></a> |
| ΚΕΦΑΛΑΙΟ 2ο                                                           | <a href="#"><u>11</u></a> |
| 2.1 Εφαρμογή για Microsoft Word                                       | <a href="#"><u>11</u></a> |
| 2.2 Τι είναι τα macros στο Word                                       | <a href="#"><u>11</u></a> |
| 2.3 Πώς λειτουργούν οι μακροεντολές στο Word                          | <a href="#"><u>11</u></a> |
| 2.4 Πώς φτιάχνουμε ένα macro στο Word                                 | <a href="#"><u>12</u></a> |
| ΚΕΦΑΛΑΙΟ 3ο                                                           | <a href="#"><u>17</u></a> |
| 3.1 Εφαρμογή για browser extensions (Google Chrome / Mozilla Firefox) | <a href="#"><u>17</u></a> |
| ΚΕΦΑΛΑΙΟ 4                                                            |                           |
| 4.1 Δυσκολίες που αντιμετώπισα                                        | <a href="#"><u>20</u></a> |
| 4.2 Μελλοντικές επεκτάσεις                                            | <a href="#"><u>20</u></a> |
| 4.3 Συμπεράσματα                                                      | <a href="#"><u>20</u></a> |
| Παράρτημα I - κώδικας για word                                        | <a href="#"><u>21</u></a> |
| Παράρτημα II - κώδικας για extensions                                 | <a href="#"><u>24</u></a> |
| Βιβλιογραφία                                                          | <a href="#"><u>28</u></a> |

## ΚΕΦΑΛΑΙΟ 1ο - Εισαγωγή

### Τι είναι η String Matching.

Η String Matching είναι ένας αλγόριθμος αντιστοίχισης, είναι μια κατηγορία αλγορίθμων συμβολοσειρών που προσπαθούν να βρουν ένα μέρος όπου μια ή περισσότερες συμβολοσειρές βρίσκονται μέσα σε μια μεγαλύτερη συμβολοσειρά ή κείμενο. Είναι πολύ αποδοτική στον χρόνο που χρειάζεται για την εκτέλεση εργασιών σε πολλούς τομείς.

Όπως αναφέρετε στο [1] geeksforgeeks υπάρχουν δυο τύποι αλγορίθμων αντιστοιχίσεις, οι αλγόριθμοι ακριβούς αντιστοίχισης συμβολοσειρών και η κατά προσέγγιση αλγόριθμοι αντιστοίχισης συμβολοσειρών.

Οι αλγόριθμοι ακριβούς αντιστοίχισης συμβολοσειρών είναι ο εντοπισμός μιας, πολλών ή όλων των εμφανίσεων μιας καθορισμένης συμβολοσειράς σε μια μεγάλη συμβολοσειρά (κείμενο ή ακολουθίες) έτσι ώστε η κάθε αντιστοίχιση να είναι τέλεια και ακριβής. Όλα τα αλφάβητα των μοτίβων πρέπει να ταιριάζουν με την αντίστοιχη αντιστοιχισμένη υποακολουθία. Αυτά ταξινομούνται περαιτέρω σε τέσσερις κατηγορίες:

#### 1. Αλγόριθμοι που βασίζονται στη σύγκριση χαρακτήρων:

- Naïve αλγόριθμος: Περνά το μοτίβο πάνω από το κείμενο ένα προς ένα και ελέγχει για αντιστοιχία. Εάν βρεθεί ένα ταιριασμα, προχωρά ξανά κατά 1 για να ελέγξετε για τις επόμενες αντιστοιχίσεις. Χρησιμοποιείτε συνήθως για μικρά κείμενα και δεν χρειάζεται κάποια φάση προεπεξεργασίας.
- Αλγόριθμος KMP (Knuth Morris Pratt): Για κάθε φορά που εντοπίζεται κάποια αναντιστοιχία, γνωρίζουμε ήδη μερικούς από τους χαρακτήρες στο κείμενο του επόμενου παραθύρου. Με αυτόν τον τρόπο ο αλγόριθμος εκμεταλλεύεται αυτή την πληροφορία για να αποφύγει την αντιστοίχιση των χαρακτήρων που γνωρίζουμε ότι ούτως ή άλλως θα ταιριάζουν.
- Αλγόριθμος Boyer Moore: Ο συγκεκριμένος αλγόριθμος χρησιμοποιεί τα καλύτερα ευρετικά στοιχεία του αλγόριθμου Naïve και KMP. Ο αλγόριθμος ξεκινά να σαρώνει και να ταιριάζει από τα δεξιά προς τα αριστερά ξεκινώντας από το πιο δεξιό.

#### 2. Μέθοδος Deterministic Finite Automaton (DFA):

- Automaton Matcher Algorithm: Ξεκινά από την πρώτη κατάσταση του πίνακα και τον πρώτο χαρακτήρα του κειμένου. Σε κάθε βήμα, εξετάζει τον επόμενο χαρακτήρα του κειμένου και αναζητά την επόμενη κατάσταση του ενσωματωμένου πίνακα και μεταβαίνει σε μια νέα κατάσταση. Εξετάζει κάθε χαρακτήρα του κειμένου ακριβώς μία φορά και αναφέρει όλες τις έγκυρες μετατοπίσεις

#### 3. Αλγόριθμοι που βασίζονται σε Bit (μέθοδος παραλληλισμού):

- Αλγόριθμος Aho-Corasick: Βρίσκει όλες τις λέξεις σε χρόνο  $O(n + m + z)$  όπου  $n$  είναι το μήκος του κειμένου και  $m$  είναι ο συνολικός αριθμός χαρακτήρων σε όλες τις λέξεις και  $z$  είναι ο συνολικός αριθμός εμφανίσεων των λέξεων στο κείμενο. Αυτός ο αλγόριθμος αποτελεί τη βάση της αρχικής εντολής Unix fgrep.

#### 4. Αλγόριθμοι αντιστοίχισης συμβολοσειρών κατακερματισμού:

- Αλγόριθμος Rabin Karp: Αντιστοιχεί την τιμή κατακερματισμού του μοτίβου με την τιμή κατακερματισμού της τρέχουσας υποσυμβολοσειράς κειμένου και εάν οι τιμές κατακερματισμού ταιριάζουν, τότε μόνο αρχίζει να ταιριάζει με μεμονωμένους χαρακτήρες.

Οι αλγόριθμοι αντιστοίχισης συμβολοσειρών κατά προσέγγιση πραγματοποιούν αναζήτηση για υποσυμβολοσειρές της συμβολοσειράς εισόδου. Αυτός ο αλγόριθμος βρίσκει όλες τις εμφανίσεις μοτίβων στο κείμενο των οποίων η απόσταση είναι το πολύ κάποιος αριθμός  $k$ . Μερικές γνωστές αποστάσεις επεξεργασίας είναι η Απόσταση επεξεργασίας Levenshtein και απόσταση επεξεργασίας Hamming.

Αυτές οι δύο τεχνικές χρησιμοποιούνται όταν η ποιότητα του κειμένου είναι αρκετά χαμηλή, δηλαδή υπάρχουν λάθοι. Για παράδειγμα ορθογραφικά λάθοι στο μοτίβο ή το κείμενο, εύρεση υποακολουθιών DNA μετά από μετάλλαξη κ.λπ. Μερικοί κατά προσέγγιση αλγόριθμοι αντιστοίχισης συμβολοσειρών είναι:

- Naive Approach
- Αλγόριθμος Sellers (Δυναμικός Προγραμματισμός)
- Αλγόριθμος Shift (Αλγόριθμος Bitmap)

Κάποιες εφαρμογές που χρησιμοποιούν αλγόριθμο αντιστοίχισης συμβολοσειρών είναι οι εξής:

- Ανίχνευση λογοκλοπής: Τα έγγραφα που συγκρίνονται για λογοκλοπή αποσυνδέονται σε διακριτικά συμβολοσειρών και συγκρίνονται χρησιμοποιώντας αλγόριθμους αντιστοίχισης συμβολοσειρών. Έτσι, αυτοί οι αλγόριθμοι χρησιμοποιούνται για να εντοπίσουν ομοιότητες μεταξύ των δυο εγγράφων και να δηλώσουν εάν το κείμενο που δώσαμε είναι λογοκλοπή ή πρωτότυπο.
- Βιοπληροφορική και αλληλουχία DNA: Στην βιοπληροφορική περιλαμβάνετε η εφαρμογή της τεχνολογίας των πληροφοριών και της επιστήμης των υπολογιστών σε προβλήματα που αφορούν γενετικές αλληλουχίες για την εύρεση προτύπων DNA. Οι αλγόριθμοι αντιστοίχισης συμβολοσειρών και η ανάλυση DNA χρησιμοποιούνται για την εύρεση της εμφάνισης του συνόλου προτύπων.
- Digital Forensics: Οι αλγόριθμοι αντιστοίχισης συμβολοσειρών Digital Forensic χρησιμοποιούνται για τον εντοπισμό συγκεκριμένων συμβολοσειρών κειμένου που ενδιαφέρουν στο ψηφιακό εγκληματολογικό κείμενο, οι οποίες χρησιμεύουν για την έρευνα.
- Ορθογραφικός έλεγχος: Το Trie βασίζεται σε ένα καθορισμένο σύνολο μοτίβων που χρησιμοποιείται για αντιστοίχιση συμβολοσειρών. Το κείμενο κόβετε ανά γράμμα για να μπορεί ο αλγόριθμος να εντοπίσει τα λάθοι. Στην συνέχεια το κείμενο λαμβάνεται ως είσοδος και εάν εμφανιστεί ένα τέτοιο μοτίβο, εμφανίζεται με την σωστή κατάσταση αποδοχής.
- Φίλτρα ανεπιθύμητης αλληλογραφίας: Τα φίλτρα ανεπιθύμητης αλληλογραφίας χρησιμοποιούν αντιστοίχιση συμβολοσειρών για να απορρίπτουν αυτόματα ανεπιθύμητα μηνύματα. Για να κατηγοριοποιηθεί ένα μήνυμα ηλεκτρονικού ταχυδρομείου ως ανεπιθύμητο ή όχι, εντοπίζονται κάποιες ύποπτες λέξεις κλειδιά στο περιεχόμενο του ηλεκτρονικού μηνύματος με αλγόριθμους αντιστοίχισης συμβολοσειρών. Με αυτόν τον τρόπο το περιεχόμενο ταξινομείται ως ανεπιθύμητο ή όχι.
- Μηχανές αναζήτησης ή αναζήτηση περιεχομένου σε μεγάλες βάσεις δεδομένων: Χρησιμοποιείτε για την αποτελεσματική κατηγοριοποίηση και οργάνωση δεδομένων, χρησιμοποιούνται αλγόριθμοι αντιστοίχισης συμβολοσειρών. Η

κατηγοριοποίηση γίνεται με βάση τις λέξεις κλειδιά στην αναζήτηση. Έτσι, οι αλγόριθμοι αντιστοίχισης συμβολοσειρών διευκολύνουν κάποιον να βρει τις πληροφορίες που ψάχνει.

- Σύστημα ανίχνευσης εισβολής: Τα πακέτα δεδομένων που περιέχουν λέξεις κλειδιά που σχετίζονται με εισβολές εντοπίζονται με την εφαρμογή αλγορίθμων αντιστοίχισης συμβολοσειρών. Όλος ο κακόβουλος κώδικας που ανιχνευτικό αποθηκεύεται στη βάση δεδομένων και όλα τα εισερχόμενα δεδομένα συγκρίνονται με τα αποθηκευμένα δεδομένα. Εάν βρεθεί αντιστοιχία, τότε δημιουργείται ο συναγερμός. Βασίζεται σε αλγόριθμους ακριβούς αντιστοίχισης συμβολοσειρών, όπου πρέπει να ανιχνεύεται κάθε εισερχόμενο πακέτο.

## Visual Basic

Η Visual Basic (VB) έχει αναπτυχθεί από την Microsoft και είναι μια γλώσσα προγραμματισμού τρίτης γενιάς, έχει ολοκληρωμένο περιβάλλον ανάπτυξης από τη Microsoft για το μοντέλο προγραμματισμού COM. Η Visual Basic θεωρείται μία εύκολη γλώσσα προγραμματισμού στην εκμάθηση και τη χρήση της, λόγω των χαρακτηριστικών της, καθώς έχει Γραφικό Περιβάλλον για τον χρήστη και συγγένεια με την γλώσσα προγραμματισμού BASIC. Όπως αναφέρετε στο Wikipedia η Visual Basic προέρχεται από τη γλώσσα BASIC και επιτρέπει την γρήγορη ανάπτυξη εφαρμογών με Γραφικό Περιβάλλον, πρόσβαση σε βάσεις δεδομένων χρησιμοποιώντας αντικείμενα (Remote Data Objects, ActiveX Data Objects, Data Access Objects), και τη δημιουργία στοιχείων ελέγχου ActiveX και αντικειμένων. Όλες οι γλώσσες προγραμματισμού που είναι "scripting", όπως η VBA και VBScript συντακτικά είναι παρόμοιες με τη Visual Basic, αλλά έχουν διαφορετικές επιδόσεις.

Όπως αναφέρετε στο [2] wikipedia η Visual Basic ήταν ένα από τα πρώτα συστήματα που έκαναν πρακτική τη σύνταξη προγραμμάτων για το λειτουργικό σύστημα Windows. Αυτό ήταν δυνατό επειδή η VB περιλάμβανε εργαλεία λογισμικού για την αυτόματη δημιουργία του λεπτομερούς προγραμματισμού που απαιτείται από τα Windows. Αυτά τα εργαλεία λογισμικού όχι μόνο δημιουργούν προγράμματα των Windows, αλλά εκμεταλλεύονται επίσης πλήρως τον γραφικό τρόπο που λειτουργούν τα Windows αφήνοντας τους προγραμματιστές να ζωγραφίσουν τα συστήματά τους με ένα ποντίκι στον υπολογιστή. Αυτός είναι ο λόγος που ονομάζεται Visual Basic.

Η Visual Basic παρέχει επίσης μια μοναδική και ολοκληρωμένη αρχιτεκτονική λογισμικού. Αρχιτεκτονική είναι ο τρόπος με τον οποίο συνεργάζονται προγράμματα υπολογιστών, όπως τα προγράμματα Windows και VB. Ένας από τους κύριους λόγους για τους οποίους η Visual Basic ήταν τόσο επιτυχημένη είναι ότι περιλαμβάνει όλα όσα είναι απαραίτητα για τη σύνταξη προγραμμάτων για Windows.

Η VBA είναι μια έκδοση της Visual Basic 6.0 που χρησιμοποιείται ως εσωτερική γλώσσα προγραμματισμού σε πολλά άλλα συστήματα, όπως προγράμματα του Microsoft Office, όπως είναι το Word και το Excel. (Προηγούμενες εκδόσεις της Visual Basic χρησιμοποιήθηκαν με παλαιότερες εκδόσεις του Office.) Πολλές άλλες εταιρείες εκτός από τη Microsoft έχουν χρησιμοποιήσει το VBA για να προσθέσουν ικανότητα προγραμματισμού στα δικά τους συστήματα. Το VBA δίνει τη δυνατότητα σε ένα άλλο σύστημα, όπως το Excel, να εκτελεί ένα πρόγραμμα εσωτερικά και να παρέχει αυτό που είναι ουσιαστικά μια προσαρμοσμένη έκδοση του Excel για έναν συγκεκριμένο σκοπό.

Από το 1991, όταν πρωτοπαρουσιάστηκε από τη Microsoft, υπήρξαν εννέα εκδόσεις της Visual Basic μέχρι το VB.NET 2005, την τρέχουσα έκδοση. Οι πρώτες έξι εκδόσεις ονομάζονταν όλες Visual Basic. Το 2002, η Microsoft παρουσίασε τη Visual Basic .NET 1.0, μια πλήρως επανασχεδιασμένη και ξαναγραμμένη έκδοση που ήταν βασικό μέρος μιας πολύ μεγαλύτερης αρχιτεκτονικής

υπολογιστή. Οι πρώτες έξι εκδόσεις ήταν όλες "συμβατές προς τα πίσω". Αυτό σημαίνει ότι οι νεότερες εκδόσεις της VB θα μπορούσαν να χειριστούν προγράμματα γραμμένα με προηγούμενη έκδοση. Επειδή η αρχιτεκτονική .NET ήταν μια τόσο ριζική αλλαγή, οι προηγούμενες εκδόσεις της Visual Basic πρέπει να ξαναγραφτούν για να μπορέσουν να χρησιμοποιηθούν με το .NET. Πολλοί προγραμματιστές εξακολουθούν να προτιμούν τη Visual Basic 6.0 και μερικοί χρησιμοποιούν ακόμη παλαιότερες εκδόσεις. Μία από τις μεγάλες αλλαγές που εισήγαγε το .NET ήταν η πλήρης αντικειμενοστραφής αρχιτεκτονική. Η Visual Basic 6 ήταν κυρίως αντικειμενοστραφής αλλά δεν είχε μερικά χαρακτηριστικά όπως η κληρονομικότητα.

Μία από τις μεγάλες καινοτομίες που εισήγαγε η Visual Basic ήταν ένας τρόπος να χωριστεί ένα πρόγραμμα σε δύο μέρη. Ένα μέρος γράφεται από τον προγραμματιστή και κάνει ό,τι κάνει αυτό το πρόγραμμα μοναδικό, όπως την προσθήκη δύο συγκεκριμένων τιμών. Το άλλο μέρος κάνει όλη την επεξεργασία που μπορεί να χρειαστεί οποιοδήποτε πρόγραμμα, όπως ο προγραμματισμός για την προσθήκη οποιονδήποτε τιμών. Το δεύτερο μέρος ονομάζεται runtime στη Visual Basic 6 και στις παλαιότερες εκδόσεις, αποτελεί μέρος του συστήματος της Visual Basic. Το runtime είναι στην πραγματικότητα ένα συγκεκριμένο πρόγραμμα και κάθε έκδοση της Visual Basic έχει μια αντίστοιχη έκδοση του χρόνου εκτέλεσης. Στο VB 6, ο χρόνος εκτέλεσης ονομάζεται MSVBVM60. Πολλά άλλα αρχεία χρειάζονται επίσης για ένα πλήρες περιβάλλον runtime VB 6. Στο .NET, η ίδια έννοια εξακολουθεί να χρησιμοποιείται με πολύ γενικό τρόπο, αλλά δεν ονομάζεται πλέον runtime (είναι μέρος του .NET Framework) και κάνει πολλά περισσότερα.

Με τη Visual Basic είναι δυνατή η δημιουργία .exe αρχείων, αρχείων DLL ή στοιχείων ελέγχου ActiveX, αλλά χρησιμοποιείται κυρίως για την ανάπτυξη εφαρμογών για τα Windows και τη διασύνδεση συστημάτων βάσεων δεδομένων. Μπορούμε ακόμα να δημιουργήσουμε τα παράθυρα popup όπου περιέχουν μέσα κάποιο κείμενο και κάποια κουμπιά. Τα στοιχεία ελέγχου παρέχουν τις βασικές λειτουργίες της εφαρμογής, ενώ οι προγραμματιστές μπορούν να εισαγάγουν επιπλέον λογική μέσα στο κατάλληλο χειριστή γεγονότων. Για παράδειγμα, ένα πτυσσόμενο πλαίσιο θα εμφανίζει αυτόματα μια λίστα που θα επιτρέπει στο χρήστη να επιλέξει οποιοδήποτε στοιχείο. Ένας χειριστής γεγονότων καλείται όταν ένα αντικείμενο είναι επιλεγμένο, και στη συνέχεια μπορεί να εκτελεστεί πρόσθετος κώδικας που δημιουργείται από τον προγραμματιστή για να εκτελεστεί κάποια ενέργεια που βασίζεται στο στοιχείο που έχει επιλεγθεί.

## JSON

Η JavaScript Object Notation γνωστή ως JSON είναι μια τυπική μορφή που βασίζεται σε κείμενο για την αναπαράσταση δομημένων δεδομένων με βάση τη σύνταξη αντικειμένου JavaScript. Χρησιμοποιείται αρκετά συχνά συνήθως για τη μετάδοση δεδομένων σε εφαρμογές Ιστού όπως είναι η αποστολή ορισμένων δεδομένων από τον διακομιστή στον πελάτη, ώστε να μπορούν να εμφανιστούν σε μια ιστοσελίδα ή κάποιο αντίστροφο.

Η JSON υπάρχει ως συμβολοσειρά και είναι χρήσιμη όταν θέλετε να μεταδώσετε δεδομένα σε ένα δίκτυο. Πρέπει να μετατραπεί σε ένα native JavaScript object για να αποκτήσετε πρόσβαση στα δεδομένα. Η JavaScript παρέχει ένα καθολικό αντικείμενο JSON που έχει διαθέσιμες μεθόδους για τη μετατροπή μεταξύ των δύο.

Η JSON είναι μια μορφή δεδομένων ανεξάρτητη από τη γλώσσα. Όπως αναφέρετε στο [3] wikipedia προήλθε από την JavaScript, αλλά πολλές σύγχρονες γλώσσες προγραμματισμού περιλαμβάνουν κώδικα για τη δημιουργία και την ανάλυση δεδομένων σε μορφή JSON.

Οι τύποι δεδομένων που χρησιμοποιεί η JSON είναι:

- Αριθμοί: ένας υπογεγραμμένος δεκαδικός αριθμός που μπορεί να περιέχει ένα κλασματικό μέρος και μπορεί να χρησιμοποιεί εκθετικό συμβολισμό  $e$ , αλλά δεν μπορεί να περιλαμβάνει μη αριθμούς όπως τους NaN (άπειροι αριθμοί). Η μορφή δεν κάνει διάκριση μεταξύ ακέραιου και κινητής υποδιαστολής.
- Συμβολοσειρά: μια ακολουθία από μηδέν ή περισσότερους χαρακτήρες του Unicode. Οι συμβολοσειρές οριοθετούνται με διπλά εισαγωγικά
- Boolean: επιστρέφει true ή false
- Πίνακες: μια ταξινομημένη λίστα με μηδέν ή περισσότερα στοιχεία, που το καθένα από τα οποία μπορεί να είναι οποιουδήποτε τύπου. Οι πίνακες χρησιμοποιούν σημειογραφία σε αγκύλες με στοιχεία διαχωρισμένα με κόμμα.
- Αντικείμενα: μια συλλογή ζευγών ονόματος - τιμές όπου τα ονόματα (ονομάζονται επίσης κλειδιά) είναι συμβολοσειρές.
- null: μια κενή τιμή, χρησιμοποιούμε την λέξη null

## JavaScript

Η JavaScript είναι μια text based γλώσσα προγραμματισμού που χρησιμοποιείται τόσο από client side όσο και από server side και μας επιτρέπει να κάνουμε τις ιστοσελίδες διαδραστικές. Όπου η HTML και η CSS είναι γλώσσες που δίνουν δομή και στυλ σε ιστοσελίδες, η JavaScript δίνει στις ιστοσελίδες διαδραστικά στοιχεία που προσελκύουν έναν χρήστη. Ακόμα μπορεί να χρησιμοποιηθεί σε web applications, στο game development και σε πολλά άλλα.

Πολλά browsers χρησιμοποιούν JavaScript σαν scripting γλώσσα για να κάνουν δυναμικό προγραμματισμό στο web. Μπορεί να υποστηρίξει μαθηματικούς υπολογισμούς, μας επιτρέπει να προσθέσουμε δυναμικό περιεχόμενο HTML στο DOM, δημιουργεί δηλώσεις δυναμικού στυλ, ανακτά περιεχόμενο από άλλο website και πολλά άλλα.

Στην JavaScript, όπως αναφέρετε στο [4] developer.Mozilla τα δεδομένα πρέπει να είναι του ενός ή του άλλου τύπου. Η JavaScript πρέπει να το γνωρίζει αυτό, ώστε να ξέρει πώς να το χρησιμοποιεί με άλλα δεδομένα ή πώς να λειτουργεί σε τέτοια δεδομένα. Ακολουθούν οι τύποι δεδομένων που μπορεί να πάρει.

- Αριθμός: μπορεί να εφαρμοστεί αριθμητική πράξη
- Συμβολοσειρά: οτιδήποτε βρίσκεται ανάμεσα σε μονά εισαγωγικά ( '...'), διπλά εισαγωγικά ( "...") και backtick ( ...).
- Boolean: έχει μόνο δύο τιμές True ή False
- Πίνακας: μια ομάδα δεδομένων (που μπορεί να είναι οποιουδήποτε τύπου, συμπεριλαμβανομένων των πινάκων) που χωρίζονται με κόμμα. Η ευρετηρίαση ξεκινά από το 0.
- Object: μια ομάδα δεδομένων αλλά με τη μορφή key:value. Το key πρέπει να είναι μια συμβολοσειρά και η τιμή μπορεί να είναι οποιοσδήποτε τύπος συμπεριλαμβανομένου ενός άλλου αντικειμένου. Η πρόσβαση στο περιεχόμενο της ομάδας γίνεται με το κλειδί.
- Undefined: Αυτά τα δεδομένα μπορούν να αντιστοιχιστούν σε μια μεταβλητή ρητά ή σιωπηρά (από JavaScript) εάν μια μεταβλητή έχει δηλωθεί αλλά δεν έχει εκχωρηθεί τιμή.
- Null: σημαίνει ότι δεν υπάρχει τιμή. Έχει μια τιμή, αλλά όχι μια πραγματική τιμή, μάλλον μηδενική.
- Function - Συνάρτηση: Μια συνάρτηση είναι ένας τύπος δεδομένων που καλεί ένα μπλοκ κώδικα όταν καλείται. Η JavaScript επιτρέπει στους χρήστες να αλληλεπιδρούν με ιστοσελίδες.

Όπως αναφέρετε στο [5] hackreactor δεν υπάρχουν σχεδόν όρια στα πράγματα που μπορούμε να κάνουμε με την JavaScript σε μια ιστοσελίδα. Ακολουθούν μερικά παραδείγματα:

- Εμφάνιση ή απόκρυψη περισσότερων πληροφοριών με το πάτημα ενός κουμπιού.
- Να αλλάξει το χρώμα ενός κουμπιού όταν το ποντίκι περνά από πάνω του.
- Να αλλάζουν φωτογραφίες αυτόματα από ένα album που βρίσκετε στο site.
- Μεγέθυνση ή σμίκρυνση μιας εικόνας.
- Εμφάνιση χρονοδιακόπτη ή αντίστροφης μέτρησης σε έναν site.
- Αναπαραγωγή ήχου και βίντεο σε μια ιστοσελίδα.
- Εμφάνιση κινούμενων εικόνων.

## CSS

Όπως αναφέρετε στο [6] tutorialpoint η Cascading Style Sheets είναι μια style sheet γλώσσα που χρησιμοποιείτε για την εμφάνιση και την αίσθηση websites. Χρησιμοποιώντας την CSS, μπορούμε να ελέγξουμε το χρώμα του κειμένου, το στυλ των γραμματοσειρών, την απόσταση μεταξύ των παραγράφων, το μέγεθος και τη διάταξη των στηλών, ποιες εικόνες φόντου ή χρώματα χρησιμοποιούνται, σχέδια διάταξης, παραλλαγές στην οθόνη για διαφορετικές συσκευές και μεγέθη οθόνης καθώς και μια ποικιλία άλλων εφέ.

Κάποια πλεονεκτήματα της CSS είναι:

- Εξοικονόμηση χρόνου: Μπορούμε να γράψουμε CSS μία φορά και στη συνέχεια να επαναχρησιμοποιούμε το ίδιο φύλλο σε πολλές σελίδες HTML. Μπορούμε να ορίσουμε ένα στυλ για κάθε στοιχείο HTML και να το εφαρμόσουμε σε όλες ιστοσελίδες θέλουμε.
- Οι σελίδες φορτώνουν πιο γρήγορα: Εάν χρησιμοποιούμε CSS, δεν χρειάζεται να γράφουμε χαρακτηριστικά ετικετών HTML κάθε φορά. Απλώς γράφουμε έναν κανόνα CSS μιας ετικέτας και εφαρμόζετε σε όλες τις εμφανίσεις αυτής της ετικέτας. Έτσι λιγότερος κώδικας σημαίνει ταχύτερους χρόνους λήψης.
- Εύκολη συντήρηση: Αν θέλουμε να κάνουμε μια γενική αλλαγή, απλώς αλλάζουμε το στυλ και όλα τα στοιχεία σε όλες τις ιστοσελίδες θα ενημερωθούν αυτόματα.
- Ανώτερα στυλ σε σχέση με την HTML: Η CSS έχει πολύ ευρύτερο φάσμα χαρακτηριστικών από την HTML, ώστε να μπορεί να δώσει μια πολύ καλύτερη εμφάνιση στη σελίδα HTML σε σύγκριση με τα χαρακτηριστικά HTML.
- Συμβατότητα πολλαπλών συσκευών: Τα Style sheets επιτρέπουν τη βελτιστοποίηση του περιεχομένου για περισσότερους από έναν τύπους συσκευών. Χρησιμοποιώντας το ίδιο έγγραφο HTML, μπορούμε να παρουσιάσουμε διαφορετικές εκδόσεις ενός site για φορητές συσκευές όπως PDA και κινητά τηλέφωνα ή για εκτύπωση.
- Παγκόσμια πρότυπα ιστού - Τώρα τα χαρακτηριστικά HTML καταργούνται και συνιστάται η χρήση CSS. Επομένως, είναι καλή ιδέα να αρχίσετε να χρησιμοποιείτε CSS σε όλες τις σελίδες HTML για να τις κάνετε συμβατές με μελλοντικά προγράμματα περιήγησης

Η σχέση μεταξύ HTML και CSS είναι στενά συνδεδεμένη μεταξύ τους. Δεδομένου ότι η HTML είναι μια γλώσσα σήμανσης και η CSS δίνει έμφαση στο στυλ. Η CSS δεν είναι τεχνικά αναγκαία, αλλά αν δούμε ένα site με HTML χωρίς CSS θα φαινόταν εντελώς γυμνό.

## ΚΕΦΑΛΑΙΟ 2ο

### Word

Το πρώτο πρόγραμμα έγινε στο Word, στην καρτέλα developer πάτησα το macro, στην συνέχεια έδωσα ένα όνομα και πάτησα edit για να ξεκινήσω να γράφω τον κώδικα με την γλώσσα της Visual Basic. Ο χρήστης μπορεί να επιλέξει το κείμενο που έχει γράψει στα greeklish, να τρέξει το macro και να το μετατρέψει σε ελληνικά. Αρχικά έκανα δύο πίνακες όπου μέσα στον πρώτο έβαλα τους ελληνικούς χαρακτήρες με τα κεφαλαία και μετά τα μικρά. Στον δεύτερο πίνακα έβαλα μέσα τους αντίστοιχους Λατινικούς χαρακτήρες των ελληνικών. Στην συνέχεια έκανα ακόμα δυο πίνακες, όπου μέσα έβαλα τα γράμματα με τόνους σε ελληνικά και αγγλικά αντίστοιχα. Ένωσα τους αντίστοιχους πίνακες μαζί, δηλαδή αν το πρόγραμμα βρίσκει λατινικό χαρακτήρα να πηγαίνει στον πίνακα με τους ελληνικούς χαρακτήρες και αν βρίσκει λατινικό χαρακτήρα με τόνο να πηγαίνει στον πίνακα που είναι οι ελληνικοί χαρακτήρες με τόνο. Αφού είδα ότι δουλεύει σωστά προχώρησα στην δημιουργία νέων πινάκων όπου είναι για τα διαλυτικά και διαλυτικά με τόνους. Το πρόγραμμα ανιχνεύει τους χαρακτήρες που έχουν μπροστά ; για τόνο, : για διαλυτικά και W για τα διαλυτικά με τόνο. Με αυτό τον τρόπο το πρόγραμμα αντικαθιστά από τους σωστούς πίνακες και παίρνει και το επόμενο γράμμα μαζί για να ξέρει πιο γράμμα έχουμε γράψει.

### Τι είναι τα Macros στο Word.

Τα macros θεωρούνται τυπικό χαρακτηριστικό στον προγραμματισμό καθώς και στη χρήση προγραμμάτων υπολογιστών. Είναι βασικά υποπρογράμματα που μπορούν να δημιουργήσουν οι προγραμματιστές και οι χρήστες για να αποθηκεύσουν ακολουθίες εντολών ή οδηγιών λειτουργίας, οι οποίες στη συνέχεια μπορούν να ξεκινήσουν με μια απλή ενέργεια. Όταν αντιμετωπίζετε πολλές τακτικά επαναλαμβανόμενες εργασίες, το Microsoft Word είναι συνήθως μια εφαρμογή που επωφελείται από την επιλογή αυτοματοποίησης των βημάτων εργασίας.

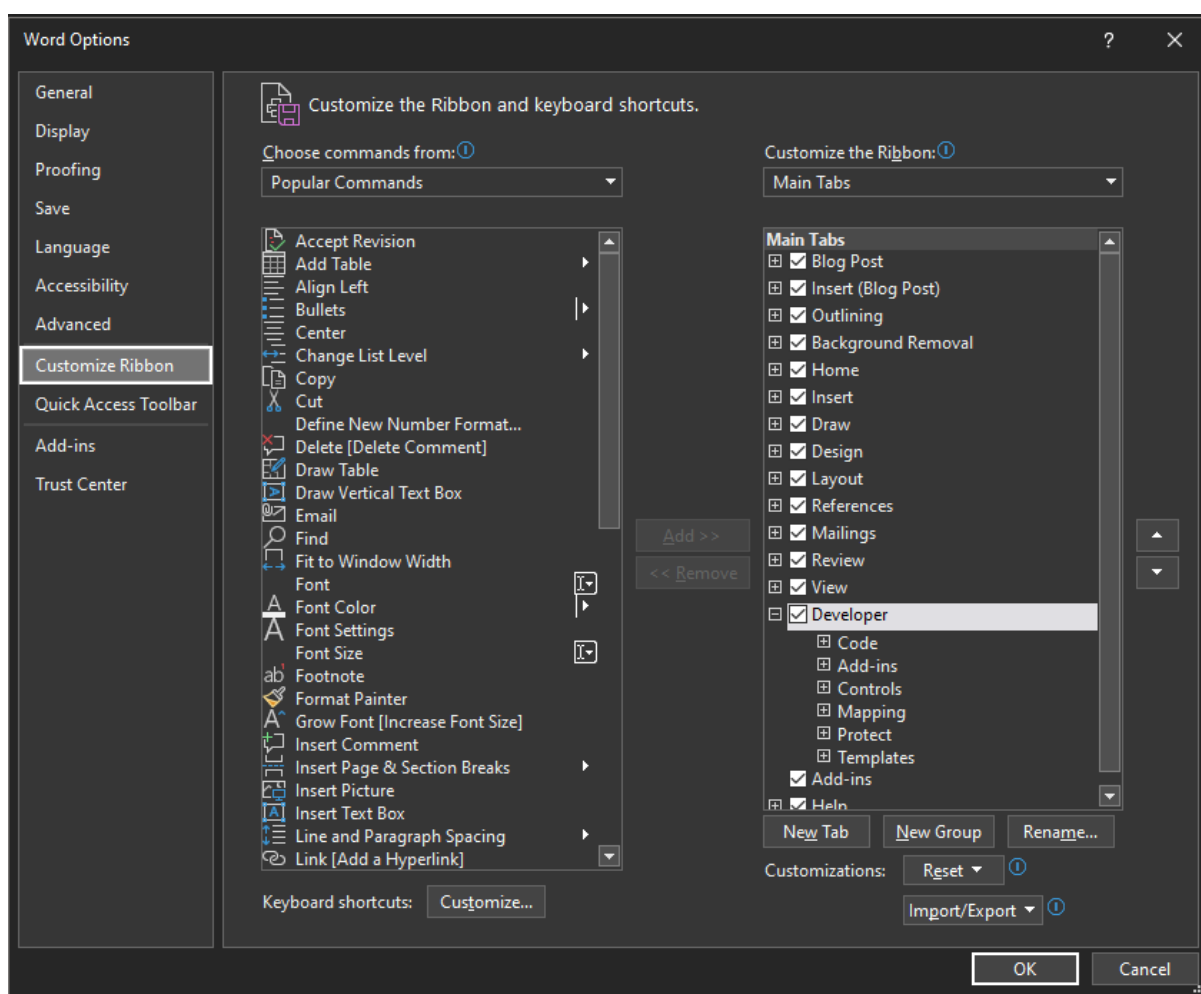
Για αυτόν τον λόγο, το πρόγραμμα επεξεργασίας (όπως είναι η Access, Excel ή το PowerPoint) περιλαμβάνουν ένα ενσωματωμένο εργαλείο για την εγγραφή και την εκτέλεση προσαρμοσμένων macro του Word. Να αναφέρουμε ότι όλα τα DocTools του Word περιέχουν macros. Ορισμένα macros του DocTools Word ενδέχεται να περιέχουν μόνο λίγες μακροεντολές, ενώ άλλα περιέχουν εκατοντάδες. Η γλώσσα που χρησιμοποιείτε είναι η Visual Basic for Applications (VBA), που αναπτύχθηκε επίσης από τη Microsoft και χρησιμεύει ως γλώσσα εγγραφής των macros.

### Πώς λειτουργούν οι μακροεντολές στο Word

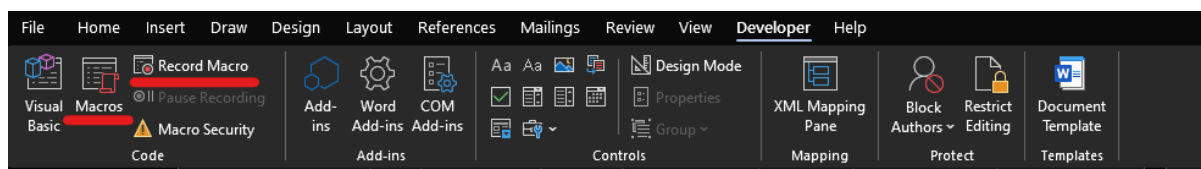
Η δημιουργία macro στο Word μπορεί αρχικά να ακούγεται σαν μια εργασία για έμπειρους προγραμματιστές. Εξάλλου, η σύνταξη νέου κώδικα προγράμματος σε μια ειδική γλώσσα προγραμματισμού δεν συγκαταλέγεται συχνά στις βασικές δεξιότητες του μέσου χρήστη του Word. Ωστόσο, το γεγονός ότι η δημιουργία και η χρήση μακροεντολών του Word είναι δυνατή χωρίς καμία απολύτως γνώση προγραμματισμού είναι χάρη στο ενσωματωμένο εργαλείο μακροεντολών. Γνωστό ως συσκευή εγγραφής μακροεντολών (macro recorder), αυτό το εργαλείο καθιστά τη σύνταξη κώδικα εντελώς περιττή. Για να δημιουργήσει μια νέα μακροεντολή, το μόνο που έχει να κάνει ο χρήστης είναι να ξεκινήσει την εγγραφή, να εκτελέσει τα επιθυμητά βήματα λειτουργίας και εντολές, στη συνέχεια να σταματήσει ξανά την εγγραφή όταν ολοκληρωθεί η ακολουθία. Ο κώδικας Visual Basic για μακροεντολές στο Word δημιουργείται αυτόματα και δεν απαιτείται περαιτέρω ενέργεια από την πλευρά του χρήστη. Ωστόσο όσοι χρήστες γνωρίζουν την γλώσσα της Visual Basic μπορούν να γράψουν τον δικό τους κώδικα χωρίς να χρειάζεται να κάνουν κάποιο record.

## Πώς φτιάχνουμε ένα macro στο Word

Εάν θέλουμε να δημιουργήσουμε τις δικές μας αυτοματοποιημένες ακολουθίες για τα έγγραφα μας στο Word, αλλά δεν έχουμε τις γνώσεις να το κάνουμε στην Visual Basic, θα πρέπει να χρησιμοποιήσουμε την επιλογή Create Macro με Recorder. Μπορούμε να βρούμε τα βασικά εργαλεία για τα macro στην καρτέλα Developer. Αν δεν μπορούμε να δούμε την καρτέλα Developer και θέλουμε να την ενεργοποιήσουμε, μπορούμε να πάμε στο File, να πατήσουμε Options, στην συνέχεια πάμε στην καρτέλα Customize the Ribbon και βρίσκουμε το Developer και πατάμε check στο κουτί. Με αυτόν τον τρόπο αν πάμε στην αρχική οθόνη της Word πάνω που είναι οι καρτέλες θα παρατηρήσουμε την καρτέλα Developer, αν πατήσουμε πάνω της θα δούμε στα αριστερά μας ένα κουμπί που γράφει Macro και δίπλα από αυτό το κουμπί Record Macro.



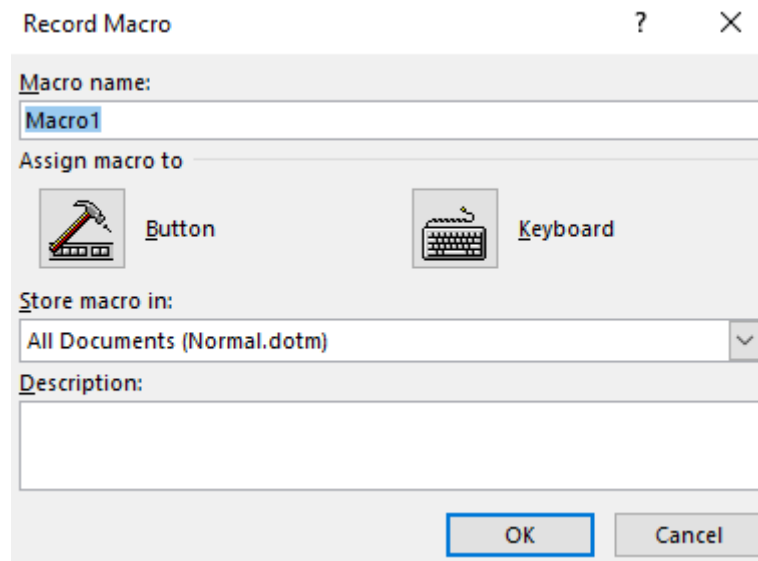
Φωτογραφία από το Word Option.



Φωτογραφία από την Καρτέλα Developer.

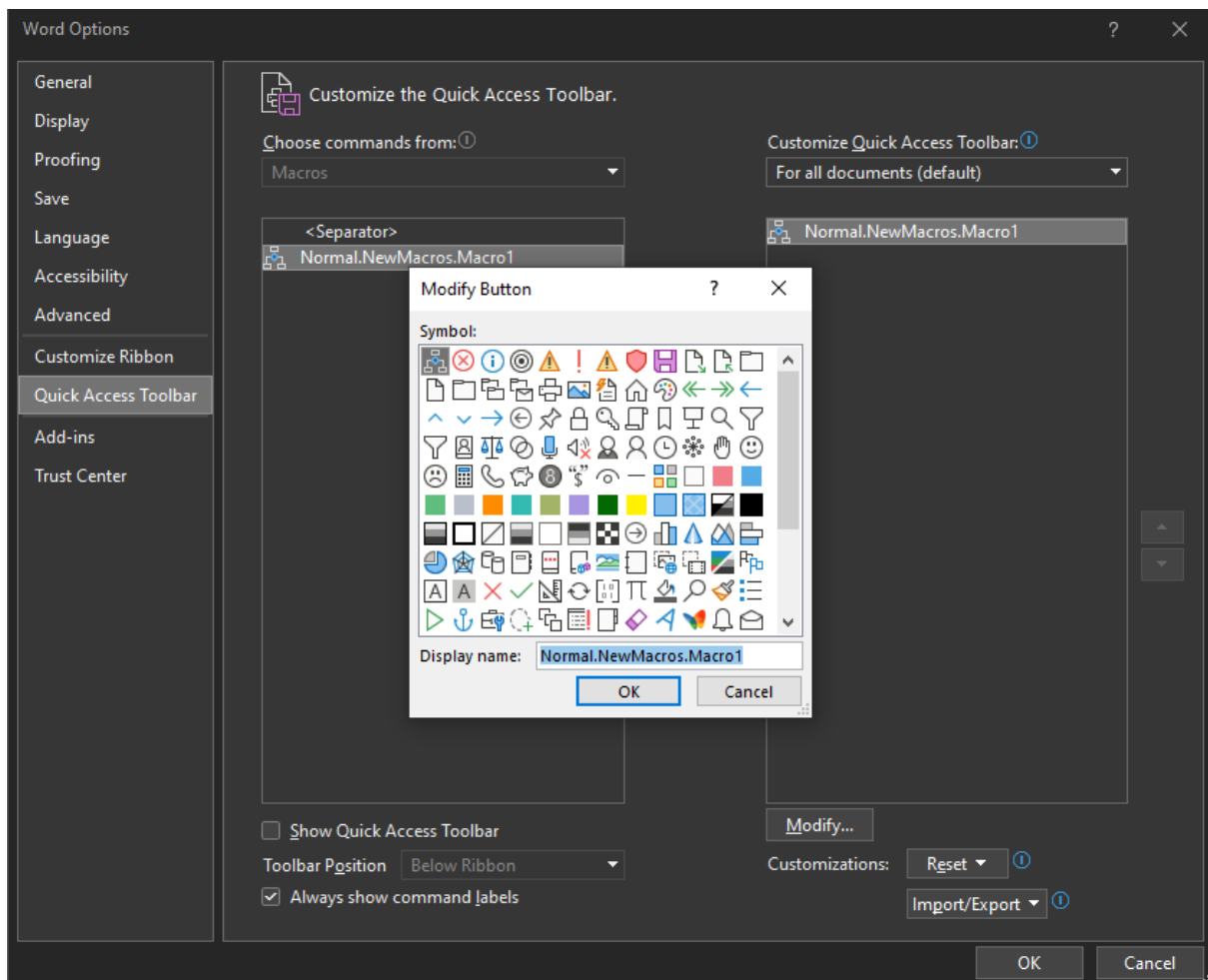
Αφού πατήσουμε το Record Macro μας βγάζει ένα παράθυρο όπου εκεί μπορούμε να δώσουμε ένα όνομα για το macro μας, πατάμε το κουμπί Button για να το κάνουμε να ενεργοποιείτε με κουμπί

στο μέλλον, στην συνέχεια επιλέγουμε σε πιο αρχείο θέλουμε να αποθηκευτεί για να δουλεύει και πατάμε OK.



Φωτογραφία από το Record Macro

Στο επόμενο παράθυρο που εμφανίζετε επιλέγουμε το macro από το αριστερό παράθυρο και πατάμε κλικ στο κουμπί ADD, καθιστώντας την επίσης επιλέξιμη στο δεξί παράθυρο. Ξαναπατώντας πάνω του μπορούμε να πατήσουμε το κουμπί Modify, στην συνέχεια να επιλέξουμε το εικονίδιο που θέλουμε και πατάμε OK.

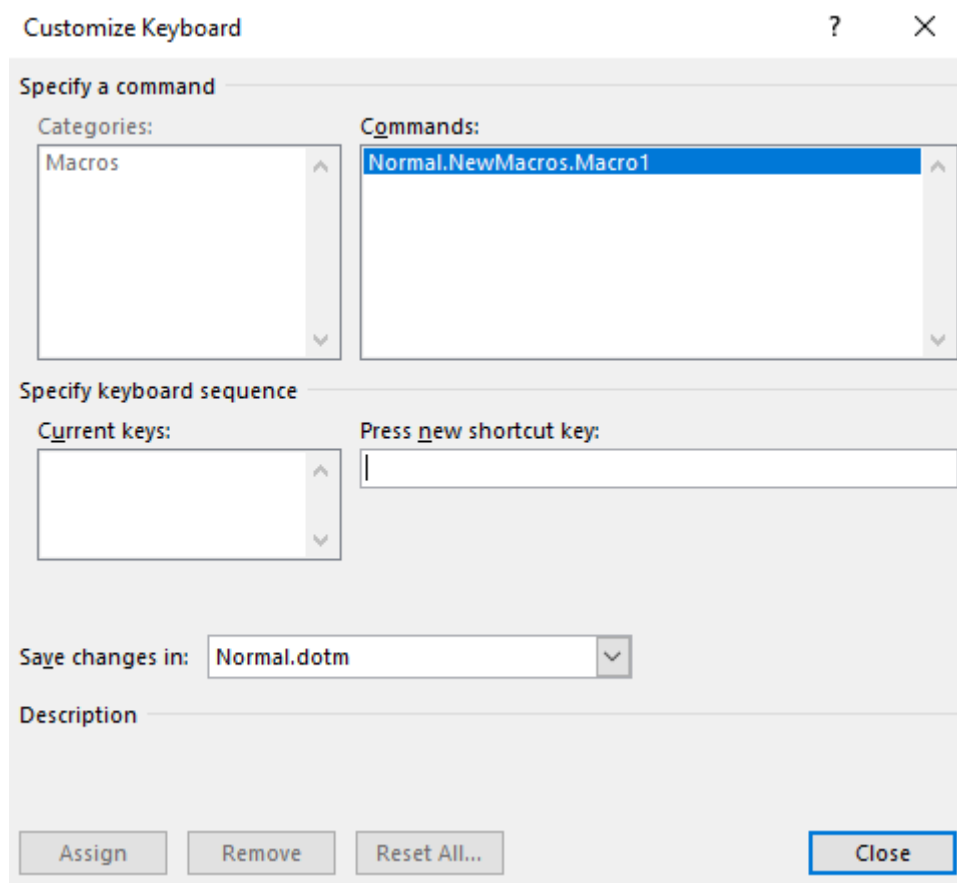


Φωτογραφία από το Modify.

Τώρα μπορούμε να εκτελέσουμε τις ενέργειες που θέλουμε για να τις αποθηκεύσουμε στο macro. Το Word θα καταγράψει τα κλικ του ποντικιού καθώς και τα κουμπιά του πληκτρολογίου που πατήσαμε. Ωστόσο, η συσκευή εγγραφής δεν αποθηκεύει κινήσεις ή επισημάνσεις με το ποντίκι. Έτσι, εάν θέλουμε να επιλέξουμε κάποιο κείμενο, για παράδειγμα, θα χρειαστεί να χρησιμοποιήσουμε το πληκτρολόγιο όπου πρέπει να κρατήσουμε πατημένο το πλήκτρο Shift και χρησιμοποιούμε τα βελάκια. Αφού ολοκληρώσουμε την ακολουθία για το macro, μπορούμε να διακόψουμε την εγγραφή πατώντας το Stop Recording.

Δημιουργία macro που εκτελείται μέσω συνδυασμό πλήκτρων.

Μπορούμε επίσης να δημιουργήσουμε ένα macro στο Word που μπορεί να ενεργοποιείτε από ένα συγκεκριμένο συνδυασμό πλήκτρων με τον ίδιο σχεδόν τρόπο. Ξεκινούμε την διαδικασία χρησιμοποιώντας το κουμπί Macro Recorder στην καρτέλα Developer, βάζουμε ένα όνομα και πατάμε στην συνέχεια το σύμβολο του keyboard για να ανοίξει το μενού να βάλουμε ένα καινούριο hotkey. Επιπλέον μπορούμε να επιλέξουμε σε πιο αρχείο θέλουμε να δουλεύει και πατάμε στην συνέχεια OK. Στο μενού Customize Keyboard που εμφανίζεται στη συνέχεια, επιλέγουμε το macro στην περιοχή Commands. Στη συνέχεια, κάνουμε κλικ στο πεδίο Press new shortcut key με το αριστερό κουμπί του ποντικιού και πατάμε τον συνδυασμό πλήκτρων που θέλουμε. Εδώ είναι σημαντικό να πατήσουμε αυτά τα πλήκτρα ταυτόχρονα.

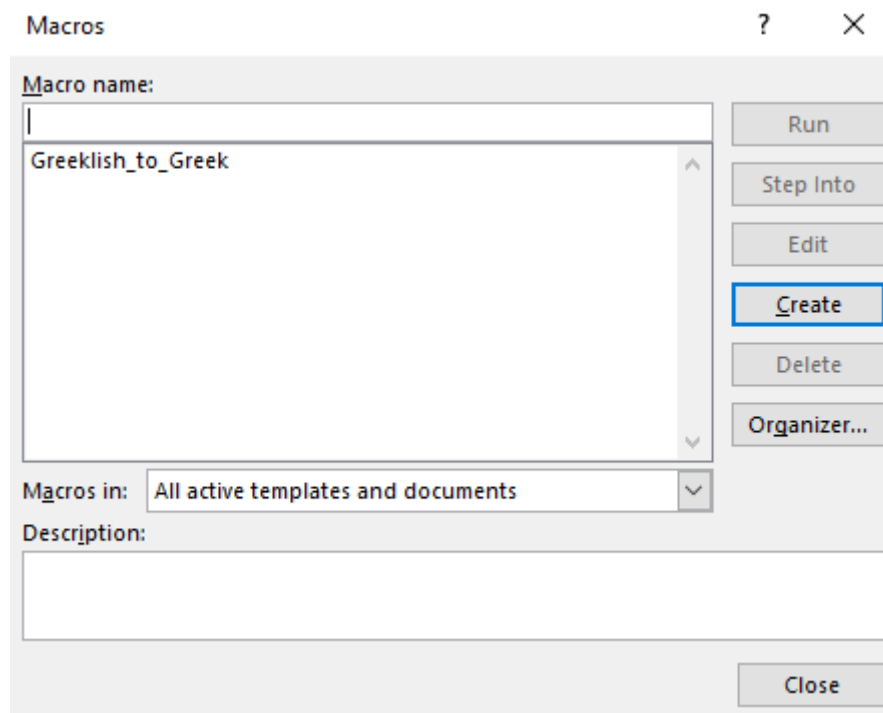


Φωτογραφία από το Customize keyboard.

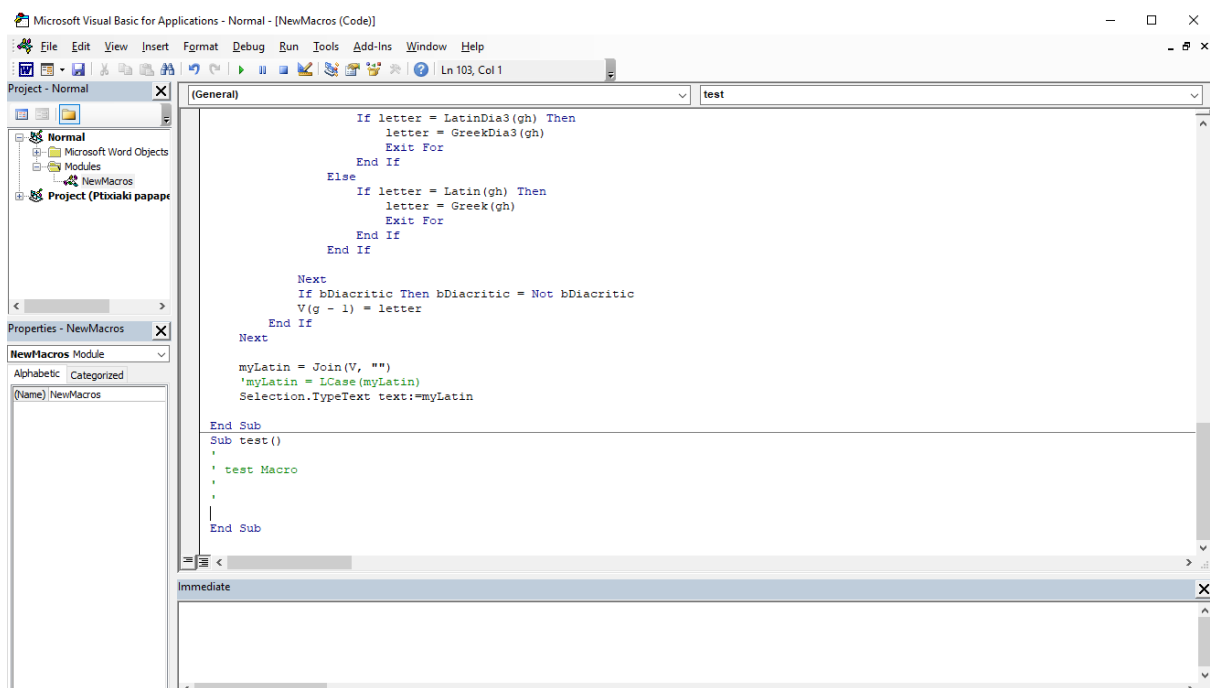
Στην περιοχή Save changes in, μπορούμε να ορίσουμε εάν ο συνδυασμός πλήκτρων θα πρέπει να ισχύει κανονικά ή μόνο για ένα συγκεκριμένο έγγραφο του Word, για να επιβεβαιώσουμε την επιλογή μας, κάνουμε κλικ στο κουμπί Assign. Μόλις κλείσουμε το μενού, θα ξεκινήσει η εγγραφή μακροεντολών.

Αφού δημιουργήσουμε το macro, μπορούμε να το ξεκινήσουμε ανά πάσα στιγμή χρησιμοποιώντας την καθορισμένη επιλογή εκτέλεσης. Χρειάζεται μόνο να χρησιμοποιήσουμε τον συνδυασμό πλήκτρων καθορισμού ή να κάνουμε κλικ στο αντίστοιχο κουμπί στο μενού quick-access πάνω αριστερά. Μπορούμε επίσης να εκτελέσουμε το macro του Word χρησιμοποιώντας τη λίστα των macro.

Ένας ακόμα τρόπος να κάνουμε macros εάν έχουμε εμπειρία με την Visual Basic είναι από την καρτέλα Developer να πατήσουμε το κουμπί Macros. Αρχικά πρέπει να δώσουμε ένα όνομα και στην συνέχεια πατάμε create. Έχοντας πατήσει το κουμπί create θα μας ανοίξει αυτόματα την Visual Basic και εκεί μπορούμε να γράψουμε τον κώδικα μας. Τελειώνοντας με τον κώδικα κάνουμε save και μπορούμε να τον βρούμε και να τον εκτελέσουμε από το κουμπί Macros.



Φωτογραφία από την καρτέλα Macros



Φωτογραφία από την Visual Basic

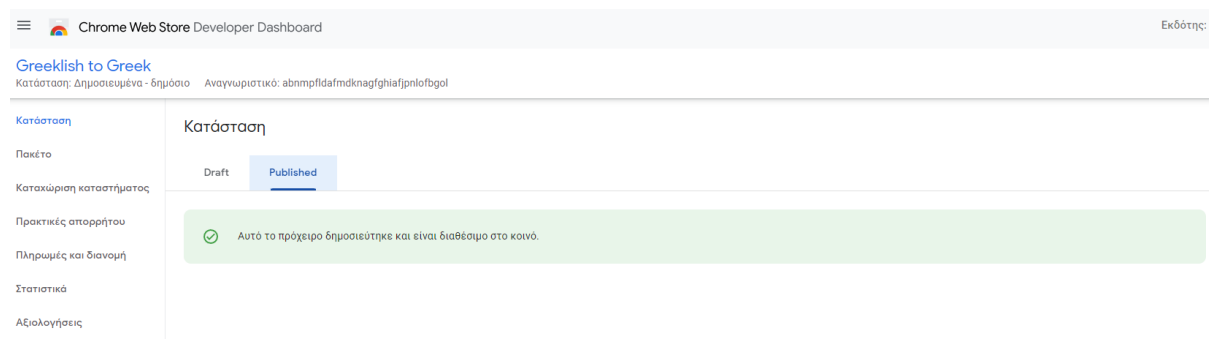
## ΚΕΦΑΛΑΙΟ 3ο

### Εφαρμογή για browser extensions (Google Chrome / Mozilla Firefox)

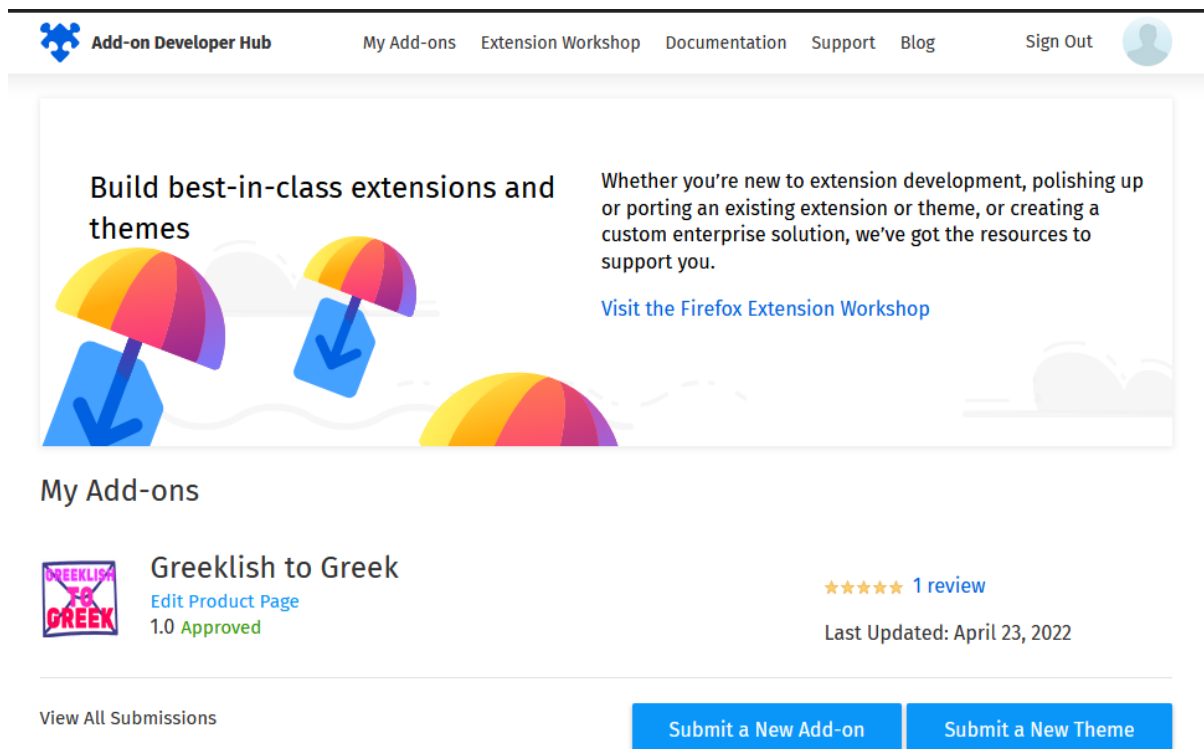
Για το δεύτερο πρόγραμμα έπρεπε να γίνει ένα extension αρχικά στο Google Chrome όπου ο χρήστης θα μπορεί να γράψει το greeklish κείμενο του και με το πάτημα ενός κουμπιού να λάβει το αντίστοιχο ελληνικό. Για να ξεκινήσω την δημιουργία του extension έπρεπε να κάνω ένα φάκελο και μέσα να βάλω το manifest.json όπου μέσα περιέχει τα στοιχεία του extension. Επίσης πρέπει να γίνουν δυο ακόμα φάκελοι, ο ένας είναι τα icons όπου περιέχει τα εικονίδια του extension και ο άλλος είναι ο popup που έχει τα scripts του προγράμματος. Αρχικά ξεκίνησα με το αρχείο style.css όπου απλά έκανα ένα μεγάλο κουτί για body και μέσα έβαλα δυο πιο μικρά κουτιά για το text, στην συνέχεια πρόσθεσα και ένα κουμπί όπου το πατάς για να ενεργοποιηθεί ο κώδικας για να σου μετατρέψει το greeklish κείμενο από το πρώτο κουτί σε ελληνικά στο δεύτερο. Ακολούθως ξεκίνησα να κάνω το πρόγραμμα στην JavaScript όπου χρησιμοποίησα την εντολή `new_text = new_text.replace(/x/g, "χ");` για να αντικαταστήσω όλα τα γράμματα. Έπειτα έκανα ένα αρχείο content.html όπου μέσα ένωσα τα scripts και έδωσα πιο πολλές πληροφορίες στο extension όπως μια ετικέτα πάνω από τα κουτιά κειμένου και ονόμασα το κουμπί Replace. Αφού σιγουρεύτηκα ότι το πρόγραμμα δουλεύει όπως πρέπει, πρόσθεσα ένα ακόμα hyperlink κάτω δεξιά όπου εάν πατήσουμε πάνω μας παραπέμπει στον ανοικτό κώδικα του προγράμματος στο GitHub. Στην συνέχεια συμπλήρωσα κάποια δεδομένα στο manifest.json αρχείο για παράδειγμα, ποιό είναι το version, όνομα, description κτλ. Έτσι τα έβαλα όλα σε ένα zip αρχείο και το δοκίμασα αν δουλεύει locally από το [7] `chrome://extensions`. Πάτησα το Developer Mode και στην συνέχεια Load unpacked, έτσι μπορούσα να δω αν δουλεύει σωστά μέσα στο Google Chrome. Έπειτα έκανα ένα developer account στο Google Chrome και το ανέβασα στο [8] Chrome WebStore.

Ακολούθως ανέβασα το ίδιο extension και στο Mozilla Firefox όπου και εκεί χρειάστηκε να κάνω ένα developer account και να το ανεβάσω στο [9] Mozilla Add-On Developer Hub.

Αφού έγινε Approve στο Google Chrome και στο Mozilla Firefox έλεγξα μέσα στο web store τους αν μπορώ να βρω το extension και εάν δουλεύει όπως πρέπει.

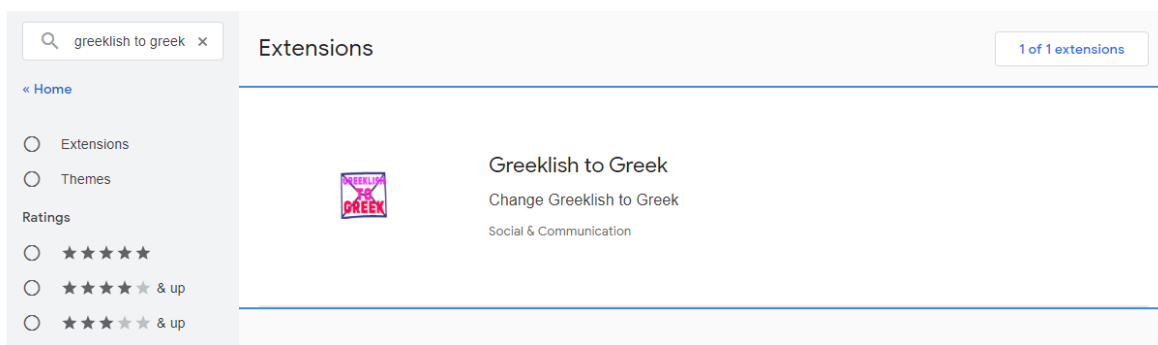


Φωτογραφία από το Chrome Web Store Developer Dashboard



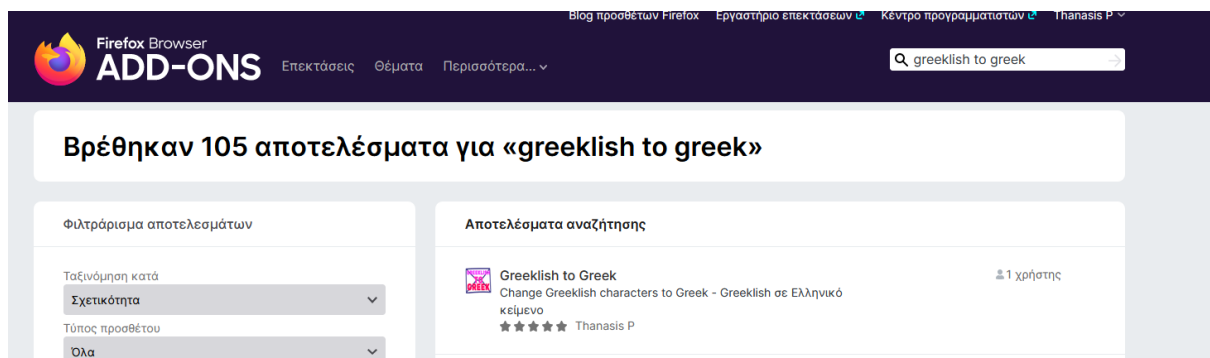
### Φωτογραφία από το DeveloperHub του Mozilla FireFox

Ο χρήστης μπορεί να το κατεβάσει το πρόγραμμα μπαίνοντας στο [10] Web Store του Google Chrome και αφού γράψει στο search bar Greeklisch to Greek θα του βγάλει το πρόγραμμα στα αποτελέσματα. Αφού έχει πατήσει πάνω, πατάει το κουμπί Add to Chrome και στο popup παράθυρο που βγάζει πατάει Add Extension. Στην συνέχεια αφού το πρόγραμμα εγκατασταθεί μπορούμε να το βρούμε στην καρτέλα Extension του Google Chrome.



### Φωτογραφία από τα αποτελέσματα που βγάζει το Web Store του Google Chrome

Για την αντίστοιχη εγκατάσταση στο Mozilla Firefox ο χρήστης πρέπει να μπει στην σελίδα [9] Addons και στην συνέχεια να γράψει στο search bar Greeklisch to Greek για να του εμφανίσει το πρόγραμμα στα αποτελέσματα. Αφού έχουμε πατήσει πάνω στο πρόγραμμα, στην επόμενη σελίδα πατάμε Add to FireFox, έπειτα στο popup παράθυρο πατάμε την επιλογή Add και μπορούμε να βρούμε το πρόγραμμα πάνω δεξιά όπου βρίσκονται τα extensions.



Φωτογραφία από τα αποτελέσματα που βγάζει το Firefox ADD-ONS

## ΚΕΦΑΛΑΙΟ 4ο

### 4.1 Δυσκολίες που αντιμετώπισα

Κάποιες δυσκολίες που αντιμετώπισα σχετικά με το πρόγραμμα είναι στην Word. Αφού είχα τελειώσει με τους πρώτους τέσσερις πίνακες, όταν δοκίμασα το πρόγραμμα αρχικά δεν μετέτρεπε κανένα λατινικό χαρακτήρα σε ελληνικό. Μετά από κάποια έρευνα που έκανα κατάλαβα ότι η Visual Basic δεν μπορεί να αναγνωρίσει τους ελληνικούς χαρακτήρες στον κώδικα. Στην συνέχεια όπως αναφέρετε στο [11] vbforums έπρεπε να αντικαταστήσω όλους τους ελληνικούς χαρακτήρες με τον αντίστοιχο δεκαεξαδικό αριθμό τους από το ASCII. Οπός αναφέρετε στο [12] docs.microsoft χρησιμοποίησα το ChrW και μετά σε παρένθεσή τον δεκαεξαδικό αριθμό για τον αντίστοιχο ελληνικό χαρακτήρα που ήθελα, σιγουρεύτηκα ότι δουλεύει σωστά και μετά αντικατέστησα όλους τους ελληνικούς χαρακτήρες με τους αριθμούς του ASCII τους οποίους βρήκα από το Character Map των Windows.

Ένα ακόμα πρόβλημα που αντιμετώπισα είναι ότι το Mozilla Firefox δεν έπαιρνε ακριβώς το ίδιο αρχείο με το Google Chrome, για αυτόν τον λόγο έπρεπε να αλλάξω το manifest.json για να μπορεί να το δεκτεί. Η μόνη διαφορά είναι ότι έπρεπε να ρίξω την manifest\_version από το τρία στο δυο γιατί το Mozilla δεν την υποστήριζε.

### 4.2 Μελλοντικές επεκτάσεις

Στο μέλλον το πρόγραμμα μπορεί να ανέβει και σε άλλα browsers όπως το Opera, Microsoft Edge, Brave και Safari. Επίσης ο κώδικας είναι open source, έτσι μπορούν και άλλοι developers να προσθέσουν τα δικά τους χαρακτηρίστηκα στο μέλλον. Ακόμα μπορούν να πάρουν τον JavaScript κώδικα και να τον αλλάξουν πολύ εύκολα σε άλλη γλώσσα επειδή είναι πάρα πολύ απλός.

### 4.3 Συμπεράσματα

Τα προγράμματα είναι πολύ εύκολά στην χρήση και έτσι μπορούν να τα χρησιμοποιήσουν χρήστες κάθε ηλικίας. Επίσης το πρώτο πρόγραμμα είναι συμβατό σε όλες τις εκδόσεις της Word και το δεύτερο μπορεί να βρεθεί πολύ εύκολα στα Web Store των αντίστοιχων browsers γράφοντας απλά [13] [14] Greeklish to Greek. Ακόμα μπορεί να χρησιμοποιηθεί καθημερινά εάν ο χρήστης είναι συνηθισμένος να γράφει greeklish.

## Παράρτημα Ι - Κώδικας για Word

Sub Greeklish\_to\_Greek()

Dim text As String

Dim pl As Integer, g As Integer, gh As Integer

Dim letter As String, myLatin As String

Dim V, Greek, GreekDia, GreekDia2, GreekDia3, Latin, LatinDia, LatinDia2, LatinDia3

Dim bDiacritic As Boolean, DiaArraySize As Long, DiaArrayType As String

text = Selection

Greek = VBA.Array(ChrW(&H391), ChrW(&H392), ChrW(&H393), ChrW(&H394), ChrW(&H395),  
ChrW(&H396), ChrW(&H397), ChrW(&H398), ChrW(&H399), ChrW(&H39A), ChrW(&H39B), \_

ChrW(&H39C), ChrW(&H39D), ChrW(&H39E), ChrW(&H39F), ChrW(&H3A0), ChrW(&H3A1),  
ChrW(&H3A3), ChrW(&H3A4), ChrW(&H3A5), ChrW(&H3A6), ChrW(&H3A7), ChrW(&H3A8),  
ChrW(&H3A9), \_

ChrW(&H3B1), ChrW(&H3B2), ChrW(&H3B3), ChrW(&H3B4), ChrW(&H3B5), ChrW(&H3B6),  
ChrW(&H3B7), ChrW(&H3B8), ChrW(&H3B9), ChrW(&H3BA), ChrW(&H3BB), \_

ChrW(&H3BC), ChrW(&H3BD), ChrW(&H3BE), ChrW(&H3BF), ChrW(&H3C0), ChrW(&H3C1),  
ChrW(&H3C3), ChrW(&H3C4), ChrW(&H3C5), ChrW(&H3C6), ChrW(&H3C7), ChrW(&H3C8),  
ChrW(&H3C9), ChrW(&H3C2))

GreekDia = VBA.Array(ChrW(&H386), ChrW(&H388), ChrW(&H389), ChrW(&H38A), ChrW(&H38C),  
ChrW(&H38E), ChrW(&H38F), \_

ChrW(&H3AC), ChrW(&H3AD), ChrW(&H3AE), ChrW(&H3AF), ChrW(&H3CC),  
ChrW(&H3CD), ChrW(&H3CE))

GreekDia2 = VBA.Array(ChrW(&H3AA), ChrW(&H3AB), ChrW(&H3CA), ChrW(&H3CB))

GreekDia3 = VBA.Array(ChrW(&H390), ChrW(&H3B0))

Latin = VBA.Array("A", "B", "G", "D", "E", "Z", "H", "U", "I", "K", "L", \_  
"M", "N", "J", "O", "P", "R", "S", "T", "Y", "F", "X", "C", "V", \_  
"a", "b", "g", "d", "e", "z", "h", "u", "i", "k", "l", \_  
"m", "n", "j", "o", "p", "r", "s", "t", "y", "f", "x", "c", "v", "w")

LatinDia = VBA.Array("A", "E", "H", "I", "O", "Y", "V", \_  
"a", "e", "h", "i", "o", "y", "v")

LatinDia2 = VBA.Array("I", "Y", "i", "y")

LatinDia3 = VBA.Array("i", "y")

'text = UCase(text)

```

pl = Len(text)

ReDim V(pl - 1)

For g = 1 To pl
    letter = Mid(text, g, 1)
    If letter = ";" Or letter = ":" Or letter = "W" Then
        If g = pl Then ' last char (can't be a diacritic sign)
            V(g - 1) = letter
        Else
            If Mid(text, g + 1, 1) = " " Then ' next char is a space (this is punctuation)
                V(g - 1) = letter
            Else
                bDiacritic = True
                If letter = ";" Then
                    DiaArraySize = UBound(GreekDia)
                    DiaArrayType = ";"
                ElseIf letter = ":" Then
                    DiaArraySize = UBound(GreekDia2)
                    DiaArrayType = ":"
                ElseIf letter = "W" Then
                    DiaArraySize = UBound(GreekDia3)
                    DiaArrayType = "W"
                End If
            End If
        End If
    Else
        For gh = 0 To If(bDiacritic, DiaArraySize, UBound(Greek))
            If bDiacritic And DiaArrayType = ";" Then
                If letter = LatinDia(gh) Then
                    letter = GreekDia(gh)
                Exit For
            End If
            ElseIf bDiacritic And DiaArrayType = ":" Then
                If letter = LatinDia2(gh) Then
                    letter = GreekDia2(gh)
                Exit For
            End If
            ElseIf bDiacritic And DiaArrayType = "W" Then
                If letter = LatinDia3(gh) Then
                    letter = GreekDia3(gh)
                Exit For
            End If
        Else
            If letter = Latin(gh) Then
                letter = Greek(gh)
            Exit For
        End If
    End If
End For

```

```
        End If

    Next
    If bDiacritic Then bDiacritic = Not bDiacritic
    V(g - 1) = letter
    End If
Next

myLatin = Join(V, "")
'myLatin = LCase(myLatin)
Selection.TypeText text:=myLatin

End Sub
```

## Παράρτημα II - κώδικας για extensions

Manifest.json

```
{  
  
  "manifest_version": 3,  
  "name": "Greeklish to Greek",  
  "version": "1.0",  
  
  "description": "Change Greeklish to Greek",  
  
  "icons": {  
    "128": "icons/icon128.png",  
    "48": "icons/icon48.png",  
    "16": "icons/icon16.png"  
  },  
  
  "action": {  
    "default_title": "Change Greeklish to Greek",  
    "default_icon": "icons/icon48.png",  
    "default_popup": "popup/content.html"  
  }  
}
```

Style.css

```
body {  
  width:500px;  
  margin:10px;  
}  
  
.popup_buttons {  
  text-decoration:none;  
  background-color:#4c8ebf;  
  color:#FFFFFF;  
  padding:5px;  
}  
  
textarea {  
  width:450px;  
  height:150px;  
}  
  
#down_right_link {  
  position:absolute;  
  bottom:10px;  
  right:10px;
```

```
}
```

Script.js

```
$(document).ready(function () {  
    $("#replace_btn").click(function () {  
        let new_text = $("#original_text").val();  
        new_text = new_text.replace(/:l/g, "İ");  
        new_text = new_text.replace(/:Y/g, "Ÿ");  
        new_text = new_text.replace(/:i/g, "ı");  
        new_text = new_text.replace(/:y/g, "ü");  
        new_text = new_text.replace(/:A/g, "Ä");  
        new_text = new_text.replace(/:a/g, "ä");  
        new_text = new_text.replace(/:E/g, "Ê");  
        new_text = new_text.replace(/:e/g, "ê");  
        new_text = new_text.replace(/:H/g, "Ĥ");  
        new_text = new_text.replace(/:h/g, "ĥ");  
        new_text = new_text.replace(/:l/g, "ł");  
        new_text = new_text.replace(/:i/g, "i");  
        new_text = new_text.replace(/:O/g, "Ō");  
        new_text = new_text.replace(/:o/g, "ó");  
        new_text = new_text.replace(/:Y/g, "Ÿ");  
        new_text = new_text.replace(/:y/g, "ý");  
        new_text = new_text.replace(/:V/g, "Ų");  
        new_text = new_text.replace(/:v/g, "ŵ");  
        new_text = new_text.replace(/Wi/g, "Ŵ");  
        new_text = new_text.replace(/Wy/g, "Ŷ");  
        new_text = new_text.replace(/:i/g, "i");  
        new_text = new_text.replace(/:A/g, "Ä");  
        new_text = new_text.replace(/Wy/g, "Ŷ");  
        new_text = new_text.replace(/:y/g, "ü");  
        new_text = new_text.replace(/w/g, "ç");  
        new_text = new_text.replace(/e/g, "ε");  
        new_text = new_text.replace(/r/g, "ρ");  
        new_text = new_text.replace(/t/g, "τ");  
        new_text = new_text.replace(/y/g, "υ");  
        new_text = new_text.replace(/u/g, "θ");  
        new_text = new_text.replace(/i/g, "ι");  
        new_text = new_text.replace(/o/g, "ο");  
        new_text = new_text.replace(/p/g, "π");  
        new_text = new_text.replace(/a/g, "α");  
        new_text = new_text.replace(/s/g, "σ");  
        new_text = new_text.replace(/d/g, "δ");  
        new_text = new_text.replace(/f/g, "φ");  
        new_text = new_text.replace(/g/g, "γ");  
        new_text = new_text.replace(/h/g, "η");  
        new_text = new_text.replace(/j/g, "ξ");  
        new_text = new_text.replace(/k/g, "κ");  
    });  
});
```

```

new_text = new_text.replace(/l/g, "λ");
new_text = new_text.replace(/z/g, "ζ");
new_text = new_text.replace(/x/g, "χ");
new_text = new_text.replace(/c/g, "ψ");
new_text = new_text.replace(/v/g, "ω");
new_text = new_text.replace(/b/g, "β");
new_text = new_text.replace(/n/g, "ν");
new_text = new_text.replace(/m/g, "μ");
new_text = new_text.replace(/E/g, "Ε");
new_text = new_text.replace(/R/g, "Ρ");
new_text = new_text.replace(/T/g, "Τ");
new_text = new_text.replace(/Y/g, "Υ");
new_text = new_text.replace(/U/g, "Θ");
new_text = new_text.replace(/I/g, "Ι");
new_text = new_text.replace(/O/g, "Ο");
new_text = new_text.replace(/P/g, "Π");
new_text = new_text.replace(/A/g, "Α");
new_text = new_text.replace(/S/g, "Σ");
new_text = new_text.replace(/D/g, "Δ");
new_text = new_text.replace(/F/g, "Φ");
new_text = new_text.replace(/G/g, "Γ");
new_text = new_text.replace(/H/g, "Η");
new_text = new_text.replace(/J/g, "Ξ");
new_text = new_text.replace(/K/g, "Κ");
new_text = new_text.replace(/L/g, "Λ");
new_text = new_text.replace(/Z/g, "Ζ");
new_text = new_text.replace(/X/g, "Χ");
new_text = new_text.replace(/C/g, "Ψ");
new_text = new_text.replace(/V/g, "Ω");
new_text = new_text.replace(/B/g, "Β");
new_text = new_text.replace(/N/g, "Ν");
new_text = new_text.replace(/M/g, "Μ");
$("#new_text").val(new_text);

return false;
});
});

```

Content.html

```

<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" type="text/css" href="style.css" />
  </head>

```

```
<body>
  <div id="wrapper">
    <label>Greeklish</label><br />
    <textarea id="original_text" ></textarea>
    <br /><br />
    <label>Greek - Ελληνικά</label><br />
    <textarea id="new_text" ></textarea>
    <br /><br />
    <a href="#" id="replace_btn" class="popup_buttons" >Replace</a>
    <a href="https://github.com/p17papa5/Greeklish-to-Greek" target="_blank"
id="down_right_link">Github</a>
  </div>
  <script src="jquery.min.js"></script>
  <script src="script.js"></script>
</body>
</html>
```

## Βιβλιογραφία

- [1] <https://www.geeksforgeeks.org/applications-of-string-matching-algorithms/>
- [2] [https://el.wikipedia.org/wiki/Visual\\_Basic](https://el.wikipedia.org/wiki/Visual_Basic)
- [3] <https://en.wikipedia.org/wiki/JSON>
- [4] <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>
- [5] <https://www.hackreactor.com/blog/what-is-javascript-used-for>
- [6] [https://www.tutorialspoint.com/css/what\\_is\\_css.htm](https://www.tutorialspoint.com/css/what_is_css.htm)
- [7] <chrome://extensions>
- [8] <https://developer.chrome.com/docs/extensions/mv3/getstarted/>
- [9] <https://addons.mozilla.org>
- [10] [Chrome Web Store - Extensions \(google.com\)](#)
- [11] <https://www.vbforums.com/showthread.php?325554-Inserting-Greek-Symbols-Into-Display-Labels>
- [12] <https://docs.microsoft.com/en-us/dotnet/api/microsoft.visualbasic.strings.chrw?view=net-6.0>
- [13] <https://chrome.google.com/webstore/search/greeklish%20to%20greek?hl=en-US>
- [14] <https://addons.mozilla.org/el/firefox/search/?q=greeklish%20to%20greek>  
<https://www.ionos.com/digitalguide/online-marketing/online-sales/creating-macros-in-word/>  
<https://wordaddins.com/support/what-is-an-add-in-and-a-macro/>