

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ



ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ &
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ &
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Το λογισμικό MkDocs και οι δυνατότητές του
για δημιουργία υποστηρικτικού υλικού
διδασκαλίας μαθημάτων προγραμματισμού

Ιωάννης Χ. Βούλγαρης

Επιβλέπων Καθηγητής: Χρήστος Γκόγκος

The MkDocs software and its capabilities for creating supporting material for
teaching programming courses.

Άρτα, Ιούνιος 2025

Περιεχόμενα

Εξώφυλλο	1
Ευχαριστίες.....	3
Υπεύθυνη δήλωση μη λογοκλοπής.....	4
Κατάλογος Εικόνων	5
Περίληψη	7
Abstract.....	7
Κεφάλαιο 1 - Εισαγωγή.....	8
Σκοποί και στόχοι της εργασίας.....	8
Η σημασία του υποστηρικτικού υλικού στη διδασκαλία προγραμματισμού.....	8
Κεφάλαιο 2 - Το λογισμικό MkDocs	10
Βασικά χαρακτηριστικά.....	10
Η χρήση του MkDocs στην συγγραφή περιεχομένου.....	11
Εναλλακτικές του MkDocs.....	11
Gitbook	11
Jekyll.....	12
Sphinx.....	13
Συμπεράσματα	14
Κεφάλαιο 3 - Δημιουργία εκπαιδευτικού υλικού με το MkDocs και το Material theme.....	16
Εφαρμογή του MkDocs σε πραγματικό σενάριο	16
Εγκατάσταση του λογισμικού	16
Visual Studio Code και ιεράρχηση των αρχείων	17
Η εντολή “mkdocs serve” για τοπική ανάπτυξη	20
Το Github και η χρήση του για hosting της σελίδας.	22
Χρήση του Github.....	23
Το αρχείο ci.yml.....	25
Ιδιαιτερότητες στην σύνταξη της Markdown	31
Κεφάλαιο 4 - Τεχνολογίες σύνταξης τεχνικών κειμένων και σχεδίασης σχημάτων	35
Markdown	35
Mermaid.....	36
Excalidraw.....	38
reStructuredText (reST).....	39
Επίλογος.....	41
Βιβλιογραφία	42
Παράρτημα	44

Ευχαριστίες

Θα ήθελα να εκφράσω τις ειλικρινείς και θερμές μου ευχαριστίες σε όσους συνέβαλαν καθοριστικά στην ολοκλήρωση της παρούσας πτυχιακής εργασίας.

Πρώτιστα, οφείλω να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ. Χρήστο Γκόγκο, για την πολύτιμη καθοδήγηση, την επιστημονική υποστήριξη και τις ουσιαστικές του παρατηρήσεις καθ' όλη τη διάρκεια της εργασίας. Η συμβολή του υπήρξε καθοριστική για την ολοκλήρωση της εργασίας αυτής.

Θα ήθελα επίσης να ευχαριστήσω τα μέλη της εξεταστικής επιτροπής, κ. Γιαννακέα Νικόλαο και κ. Τζάλλα Αλέξανδρο, για τον χρόνο που διέθεσαν για την αξιολόγηση της εργασίας.

Ευχαριστώ όλους τους καθηγητές της σχολής που συνέβαλαν στην απόκτηση των απαραίτητων γνώσεων για την επιτυχή φοίτησή μου. Μέσα από τα μαθήματα και τις συμβουλές τους, μου προσέφεραν υψηλού επιπέδου επιστημονική κατάρτιση, η οποία αποτέλεσε πολύτιμο εφόδιο για την πορεία μου.

Ιδιαίτερες ευχαριστίες οφείλω στην οικογένειά μου, για τη συνεχή ηθική και συναισθηματική τους υποστήριξη καθ' όλη τη διάρκεια των σπουδών μου. Η αμέριστη συμπαράσταση και η πίστη τους στις δυνατότητές μου υπήρξαν για εμένα πηγή έμπνευσης και δύναμης.

Σε όλους τους προαναφερθέντες εκφράζω την εκτίμηση και την ευγνωμοσύνη μου.

Υπεύθυνη δήλωση μη λογοκλοπής

«Δηλώνω υπεύθυνα ότι η παρούσα εργασία είναι αποτέλεσμα δικής μου πνευματικής εργασίας. Δεν έχει υποβληθεί στο παρελθόν, ούτε έχει αντιγραφεί, εν μέρει ή εξ ολοκλήρου, από άλλες πηγές χωρίς τη δέουσα αναγνώριση.

Έχω κάνει χρήση των απαραίτητων παραπομπών και βιβλιογραφικών αναφορών, σύμφωνα με τους κανόνες ακαδημαϊκής δεοντολογίας και συγγραφής.

Κατανοώ ότι οποιαδήποτε μορφή λογοκλοπής αποτελεί σοβαρό ακαδημαϊκό παράπτωμα και επισύρει τις προβλεπόμενες κυρώσεις από τον κανονισμό σπουδών του εκπαιδευτικού ιδρύματος.»

Ο δηλών



.....
Υπογραφή

Βούλγαρης Ιωάννης

Κατάλογος Εικόνων

Εικόνα 1. Το σφάλμα λόγω της ασφάλειας των Windows.....	17
Εικόνα 2. Τα αρχεία που δημιουργούνται μετά την εκτέλεση της εντολής “mkdocs new.”...	17
Εικόνα 3. Οι επεκτάσεις markdown που χρησιμοποιήθηκαν στο project.....	18
Εικόνα 4. Το αρχείο style.css.	19
Εικόνα 5. Τα αρχεία markdown, κάθε ένα για ένα διαφορετικό κεφάλαιο.	20
Εικόνα 6. Εκτέλεση της εντολής "mkdocs serve"	21
Εικόνα 7. Προβολή site μετά το τοπικό deploy	21
Εικόνα 8. Rebuild του site σε περίπτωση αλλαγής.....	21
Εικόνα 9. Διακοπή του τοπικού deploy	22
Εικόνα 10. Δημιουργία καινούργιου αποθετηρίου.....	24
Εικόνα 11. Commit σε προσθήκη νέας γραμμής.....	24
Εικόνα 12. Commit and Push σε staged αλλαγή.....	25
Εικόνα 13. Το αρχείο ci.yml.	26
Εικόνα 14. Log file κατά το deploy.....	27
Εικόνα 15. Τα credential που ορίστηκαν στο αρχείο.	28
Εικόνα 16. Η εγκατάσταση python στο runner.	28
Εικόνα 17. Η διαδικασία για την εβδομάδα του χρόνου.	29
Εικόνα 18. Ο μηχανισμός cache του Github.	30
Εικόνα 19. Η εγκατάσταση των πακέτων python.....	30
Εικόνα 20. Build και push στο gh-pages.	31

Εικόνα 21. Ιεραρχία επικεφαλίδων με την χρήση της #.....	32
Εικόνα 22. Ιεράρχηση κεφαλίδων στον ιστότοπο.	32
Εικόνα 23. Παράδειγμα χρήσης λιστών.	33
Εικόνα 24. Παράδειγμα χρήσης πίνακα.....	33
Εικόνα 25. Κώδικας HTML για την προσαρμογή και ευθυγράμμιση εικόνων.	34
Εικόνα 26. Αντικατάσταση HTML character.	34
Εικόνα 27. Διαφοροποίηση μεταξύ εικόνας και συνδέσμου.	36
Εικόνα 28. Παράδειγμα για την χρήση code block.	36
Εικόνα 29. Κώδικας Mermaid στο project.....	37
Εικόνα 30. Η οπτικοποίηση του κώδικα Mermaid.....	38
Εικόνα 31. Αρχική σελίδα του βιβλίου	44
Εικόνα 32. Παράδειγμα σελίδας με κώδικα (foldable)	45
Εικόνα 33. Αναζήτηση σε όλο το κείμενο του βιβλίου.....	46

Περίληψη

Στην παρούσα πτυχιακή εργασία εξετάζεται η έννοια της τεκμηρίωσης (documentation) και η αναγκαιότητα της χρήσης της σε ποικίλα πεδία της καθημερινής επαγγελματικής και ακαδημαϊκής πρακτικής. Ιδιαίτερη έμφαση δίνεται στην τεχνολογία MkDocs και ειδικότερα στο θέμα Material for MkDocs, το οποίο αξιοποιεί τη γλώσσα Markdown για τη σύνθεση τεχνικού περιεχομένου. Παράλληλα, παρουσιάζεται μια γενική επισκόπηση τεχνικών και εργαλείων τεκμηρίωσης.

Ως πρακτική εφαρμογή, πραγματοποιείται η μεταφορά του εκπαιδευτικού βιβλίου «Μια σύγχρονη προσέγγιση στη γλώσσα C» που διατίθεται δωρεάν από τις εκδόσεις ΚΑΛΛΙΠΟΣ, σε μορφή τεκμηρίωσης, με τη χρήση του MkDocs και του Material theme, συνοδευόμενη από αναλυτικό οδηγό υλοποίησης.

Τέλος, διεξάγεται συγκριτική αξιολόγηση διαφορετικών εργαλείων συγγραφής και οπτικοποίησης τεχνικού περιεχομένου, όπως το Mermaid, το Excalidraw και άλλες σύγχρονες πλατφόρμες.

Abstract

This thesis examines the concept of documentation and the necessity of its use in various fields of everyday professional and academic practice. Particular emphasis is placed on the MkDocs technology and specifically on the Material for MkDocs theme, which utilizes the Markdown language for composing technical content. At the same time, a general overview of documentation techniques and tools is presented.

As a practical application, the educational book “A modern approach to the C programming language”, which is freely available from KALLIPOS publications, is converted into a documentation format using MkDocs and the Material theme, accompanied by a detailed implementation guide.

Finally, a comparative evaluation is conducted on different tools for authoring and visualizing technical content, such as Mermaid, Excalidraw, and other modern platforms.

Κεφάλαιο 1 - Εισαγωγή

Με την εξέλιξη της τεχνολογίας επέρχεται και η εξέλιξη των εκπαιδευτικών μεθόδων για την διδασκαλία. Τόσο η ανάπτυξη όσο και η οργάνωση εκπαιδευτικού υλικού αποτελεί μία από τις πιο βασικές προκλήσεις στη σύγχρονη ψηφιακή εκπαίδευση. Ένα καλά δομημένο όσο και εύχρηστο εκπαιδευτικό υλικό θεωρείται ανάγκη για τους εκπαιδευτικούς, καθώς οι φοιτητές καλούνται να κατανοήσουν πολύπλοκες έννοιες και να αναπτύξουν πρακτικές δεξιότητες μέσα από σύγχρονες διδακτικές μεθόδους. Όσον αφορά την διδασκαλία μαθημάτων προγραμματισμού απαιτεί εκπαιδευτικά εργαλεία που μπορούν να προσαρμόζονται στις πολλαπλές ανάγκες μάθησης και να υποστηρίζουν την ομαλή μετάβαση από τη θεωρία στην πράξη.

Σκοποί και στόχοι της εργασίας

Στην παρούσα εργασία θα αναφερθούμε στο λογισμικό MkDocs [1] που αναδεικνύεται ως ένα εξαιρετικό εργαλείο για τη δημιουργία documentation και εκπαιδευτικών οδηγιών. Η δυνατότητά του για διαχείριση περιεχομένου σε μορφή Markdown όσο και η απλότητα στην χρήση του, το καθιστούν προσιτό ακόμη και σε χρήστες με περιορισμένη τεχνική εμπειρία. Η λειτουργικότητα του MkDocs με την προσθήκη του γραφικού θέματος Material for MkDocs, το καθιστούν τόσο λειτουργικό που όσοι χρήστες αποκτούν πρόσβαση το χαρακτηρίζουν ως ένα φιλικό προς τον χρήστη και επαγγελματικού επιπέδου εργαλείο για τη συγγραφή, οργάνωση και δημοσίευση εκπαιδευτικού υλικού. Το Material for MkDocs ενισχύει περαιτέρω την εμπειρία χρήστη, παρέχοντας πλούσιες δυνατότητες εξατομίκευσης, καθώς και προηγμένα χαρακτηριστικά όπως υποστήριξη αναζήτησης, αυτόματη δημιουργία πλοήγησης και υποστήριξη πολυγλωσσικού περιεχομένου.

Η σημασία του υποστηρικτικού υλικού στη διδασκαλία προγραμματισμού

Η παρούσα πτυχιακή εργασία εξετάζει πώς το MkDocs μπορεί να αξιοποιηθεί για τη δημιουργία υποστηρικτικού υλικού διδασκαλίας μαθημάτων προγραμματισμού, με ιδιαίτερη έμφαση στη διαδικασία δημοσίευσης ενός βιβλίου μέσω GitHub Pages. Παρουσιάζει τις δυνατότητες του λογισμικού, εξηγεί τη διαδικασία υλοποίησης και δημοσίευσης υλικού και αναδεικνύει πώς αυτό μπορεί να ενισχύσει την εκπαιδευτική εμπειρία τόσο για τους εκπαιδευτές όσο και για τους εκπαιδευόμενους. Ιδιαίτερη σημασία δίνεται στην προσβασιμότητα, καθώς και στη δυνατότητα συνεχούς ενημέρωσης και προσαρμογής του εκπαιδευτικού περιεχομένου στις εξελίξεις του αντικειμένου.

Ο κύριος σκοπός της εργασίας είναι να παρέχει:

- Μια αναλυτική εισαγωγή στο MkDocs και στις λειτουργίες του.
- Οδηγίες για τη δημιουργία και δημοσίευση εκπαιδευτικού περιεχομένου.
- Ένα πρακτικό παράδειγμα εφαρμογής δημοσίευσης εκπαιδευτικού βιβλίου.

Η εργασία οργανώνεται σε κεφάλαια που καλύπτουν τόσο θεωρητικές όσο και πρακτικές πτυχές. Αρχικά, παρέχεται μια επισκόπηση των βασικών χαρακτηριστικών του MkDocs. Στη συνέχεια, περιγράφεται αναλυτικά η διαδικασία δημιουργίας και διαχείρισης περιεχομένου, ενώ το τελευταίο μέρος επικεντρώνεται στη δημοσίευση ενός ολοκληρωμένου έργου. Επιπρόσθετα, η εργασία εξετάζει τις προκλήσεις που ενδέχεται να προκύψουν κατά τη διαδικασία υλοποίησης, καθώς και προτάσεις για την αντιμετώπισή τους. Με αυτόν τον τρόπο, η εργασία φιλοδοξεί να αποτελέσει έναν πρακτικό οδηγό για την αξιοποίηση του MkDocs ως εργαλείο δημιουργίας εκπαιδευτικού υλικού, δίνοντας έμφαση στη λειτουργικότητα και την πρακτική εφαρμογή.

Ελπίζουμε ότι η εργασία αυτή θα αποδειχθεί χρήσιμη για εκπαιδευτές και επαγγελματίες που αναζητούν αποτελεσματικές λύσεις για τη δημιουργία και τη διάθεση εκπαιδευτικού περιεχομένου. Μέσω της ανάλυσης και της εφαρμογής που παρουσιάζεται, ευελπιστούμε να συμβάλλουμε στη βελτίωση της ποιότητας και της προσβασιμότητας της εκπαίδευσης, ενθαρρύνοντας την υιοθέτηση τεχνολογιών που ενισχύουν την εμπειρία μάθησης.

Κεφάλαιο 2 - Το λογισμικό MkDocs

Βασικά χαρακτηριστικά

Το MkDocs είναι ένα εύχρηστο εργαλείο δημιουργίας documentation, το οποίο βασίζεται στη γλώσσα Markdown και χρησιμοποιείται για τη δημιουργία στατικών ιστοσελίδων. Πρόκειται για ένα εργαλείο που απευθύνεται κυρίως σε προγραμματιστές και ομάδες ανάπτυξης λογισμικού, προσφέροντας μια απλή αλλά ισχυρή λύση για τη διαχείριση και δημοσίευση υλικού. Ένα από τα μεγαλύτερα πλεονεκτήματά του είναι η ευκολία εγκατάστασης και χρήσης, καθώς επιτρέπει τη δημιουργία ενός ολοκληρωμένου ιστότοπου με λίγες μόνο εντολές. Με την υποστήριξη της Python, το MkDocs μπορεί να εγκατασταθεί μέσω του package manager `pip`, ενώ η διαμόρφωσή του γίνεται μέσα από ένα απλό αρχείο `YAML`. Παρέχει έναν ενσωματωμένο `server` ανάπτυξης, ο οποίος επιτρέπει την προεπισκόπηση των αλλαγών σε πραγματικό χρόνο, γεγονός που διευκολύνει τη συγγραφή και την αναθεώρηση του περιεχομένου. Υπάρχουν αρκετά `themes` για το MkDocs όπως το δημοφιλές θέμα `"ReadTheDocs"` που προσφέρει μια καθαρή και ευανάγνωστη εμφάνιση, η οποία είναι συμβατή με τις βέλτιστες πρακτικές τεκμηρίωσης λογισμικού. Επιπλέον, το MkDocs υποστηρίζει επεκτάσεις και πρόσθετα, επιτρέποντας την προσθήκη επιπλέον λειτουργιών, όπως υποστήριξη για διαγράμματα μέσω `MermaidJS` και κώδικα με επισήμανση σύνταξης. Παρόλο που το MkDocs είναι εξαιρετικά δημοφιλές λόγω της απλότητάς του, υπάρχουν αρκετές εναλλακτικές που προσφέρουν παρόμοιες ή και πιο προηγμένες δυνατότητες. Ένα από τα πιο γνωστά εργαλεία είναι το `Sphinx`, το οποίο είναι επίσης γραμμένο σε Python, αλλά απευθύνεται περισσότερο σε χρήστες που χρειάζονται υψηλή προσαρμοστικότητα και εκτεταμένες δυνατότητες μορφοποίησης. Το `Sphinx` χρησιμοποιεί `reStructuredText` αντί για Markdown, γεγονός που του επιτρέπει να υποστηρίζει πιο πολύπλοκες δομές, όπως αυτόματη δημιουργία ευρετηρίων, υποστήριξη για `LaTeX` εξαγωγή, και διασύνδεση με τον κώδικα Python μέσω `docstrings`. Επιπλέον, το `Sphinx` έχει μεγάλη υποστήριξη από την επιστημονική κοινότητα, καθώς χρησιμοποιείται συχνά για την τεκμηρίωση βιβλιοθηκών και ερευνητικών έργων.

Το MkDocs κυκλοφόρησε με την `version 0.2` στις 24 Ιανουαρίου του 2014 από μία ομάδα προγραμματιστών με επικεφαλής τον Tom Christie. Σχεδόν μετά από ένα χρόνο το `"ReadTheDocs"` το πρόσθεσε στην λίστα του με τις διαθέσιμες τεχνολογίες που υποστηρίζει. Γρήγορα ακολούθησε και η υποστήριξη για τα θέματα που είναι διαθέσιμα για εγκατάσταση. Το γνωστό θέμα `Material for MkDocs` ξεκίνησε να δημιουργείται και γρήγορα κέρδισε τον ανταγωνισμό. Καθώς όμως όλο και περισσότεροι το χρησιμοποιούσαν η ομάδα ανάπτυξης αποφάσισε να προσφέρει τα νέα χαρακτηριστικά του μέσω συνδρομής. Μέσα από την συνδρομή για το `Insiders` [2] ο χρήστης μπορεί να αποκτήσει αποκλειστικά νέα χαρακτηριστικά για την σελίδα του. Στην περίπτωση που η χρηματοδότηση καταφέρει τον στόχο της, τότε τα χαρακτηριστικά αυτά γίνονται διαθέσιμα για όλους.

Η χρήση του MkDocs στην συγγραφή περιεχομένου

Πολλές εταιρείες χρησιμοποιούν το MkDocs σε συνδυασμό με το Material για να έχουν το documentation αναρτημένο στο διαδίκτυο. Ένα σημαντικό πλεονέκτημα του MkDocs έναντι άλλων τεχνολογιών παραγωγής στατικών σελίδων είναι ότι όλα τα αρχεία που επεξεργάζεται ο χρήστης για το περιεχόμενο του documentation είναι markdown αρχεία (.md) [1] .

Η Google χρησιμοποιεί για το documentation των βιβλιοθηκών του Jetpack Compose, το οποίο είναι ένα εργαλείο για σχεδίαση γραφικού περιβάλλοντος, το mkdocs material το οποίο το ονομάζει “Accompanist”¹. Ένα documentation για την ομάδα ISE (Industry Solutions Engineering) έχει φτιάξει η Microsoft με όνομα “Engineering Fundamentals Playbook”², το οποίο περιέχει βασικές οδηγίες για να αυξήσουν την συνολική αποδοτικότητα, να μειώσουν κοινά λάθη στην ομάδα για να γίνουν καλύτεροι μηχανικοί. Επίσης η κοινότητα της OWASP (Open Web Application Security Project) έχει μια σελίδα με βασικές πληροφορίες σε συγκεκριμένα θέματα πάνω σε ασφάλεια εφαρμογών το οποίο το ονόμασαν OWASP “Cheat Sheet Series”³.

Τόσο στο εξωτερικό όσο και στην Ελλάδα υπάρχουν documentation σε MkDocs με ποιο γνωστό αυτό του Εθνικού Δικτύου Υποδομών Τεχνολογίας και Έρευνας για τον υπερυπολογιστή A.R.I.S (Advanced Research Information System) “ARIS DOCUMENTAION”⁴, με βασικές οδηγίες αλλά και διαχειριστικά μηνύματα προς τους χρήστες. Επίσης το “Aristotelis Docs”⁵ Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης έχει δικό του site στο οποίο εξηγεί την πρόσβαση στον υπερυπολογιστή του Πανεπιστημίου Aristotle, την υποδομή του, διαθέσιμους πόρους του συστήματος αλλά και βασικές διεργασίες που γίνονται τόσο σε μεταπτυχιακές αλλά και σε διδακτορικές εργασίες και αποτελέσματα.

Μία ακόμη χρήση είναι αυτή του “The Blue Book”⁶ του project lyz-code/blue-book στο github. Το οποίο η δομή του θυμίζει περισσότερο ένα διαδικτυακό ημερολόγιο με πολλές και διαφορετικές ενότητες από υγεία, τέχνες και γλώσσες μέχρι επιστήμες, τεχνητή νοημοσύνη αλλά και προγραμματισμό. Παρέχει μέσα από την ιστοσελίδα της ο χρήστης βλέπει καθημερινές ενέργειές της αλλά και διάφορες άλλες δραστηριότητες και ταξίδια που κάνει με αναφορές και εικόνες.

Εναλλακτικές του MkDocs

Gitbook

Το GitBook αποτελεί μια άλλη ενδιαφέρουσα εναλλακτική, η οποία προσφέρει έναν φιλικό προς τον χρήστη επεξεργαστή και δυνατότητες συνεργασίας σε πραγματικό χρόνο. Σε

¹ <https://google.github.io/accompanist/>

² <https://microsoft.github.io/code-with-engineering-playbook/>

³ <https://cheatsheetseries.owasp.org/index.html>

⁴ <https://doc.aris.grnet.gr/>

⁵ <https://hpc.it.auth.gr/>

⁶ <https://lyz-code.github.io/blue-book/>

αντίθεση με το MkDocs, το GitBook μπορεί να χρησιμοποιηθεί είτε τοπικά είτε ως cloud-based υπηρεσία, κάτι που διευκολύνει τη συνεργασία μεταξύ διαφορετικών μελών μιας ομάδας. Παρόλο που το GitBook διαθέτει μια εύχρηστη διεπαφή, η χρήση του μπορεί να περιορίζεται από το μοντέλο συνδρομής του, καθώς πολλές προηγμένες λειτουργίες απαιτούν πληρωμή. Επίσης, παρέχει δυνατότητες ενσωμάτωσης με υπηρεσίες όπως το GitHub, διευκολύνοντας έτσι τη συνεχή ενημέρωση της τεκμηρίωσης [10].

Το GitBook είναι ένα σύγχρονο εργαλείο που επικεντρώνεται στη δημιουργία και δημοσίευση βιβλίων, οδηγιών και documentation. Αρχικά κυκλοφόρησε το 2014 από τον Aaron O'Mullan και τον Samy Pessé, και σχεδιάστηκε με γνώμονα την απλότητα και την ευχρηστία. Το GitBook χρησιμοποιεί Markdown, μια δημοφιλή γλώσσα σήμανσης που είναι απλή και εύκολη στην εκμάθηση, καθιστώντας το προσιτό ακόμα και για όσους δεν έχουν τεχνικό υπόβαθρο.

Το μεγαλύτερο πλεονέκτημα του GitBook είναι η δυνατότητά του να ενσωματώνεται άφογα με το Git, το σύστημα διαχείρισης εκδόσεων που χρησιμοποιείται ευρέως από προγραμματιστές. Με αυτόν τον τρόπο, οι χρήστες μπορούν να αποθηκεύουν και να διαχειρίζονται τα έγγραφά τους, διασφαλίζοντας ότι κάθε αλλαγή είναι καταγεγραμμένη και μπορεί να αναιρεθεί αν χρειαστεί. Αυτή η προσέγγιση ενθαρρύνει τη συνεργατικότητα, καθώς πολλές ομάδες μπορούν να δουλεύουν πάνω στο ίδιο έργο ταυτόχρονα.

Ένα ακόμα πλεονέκτημα του GitBook είναι η δυνατότητα δημοσίευσης περιεχομένου στο διαδίκτυο. Το εργαλείο παρέχει ένα πλήρως λειτουργικό hosting service, όπου τα έγγραφα μπορούν να φιλοξενηθούν, να ανανεώνονται αυτόματα και να είναι διαθέσιμα σε πραγματικό χρόνο. Το GitBook υποστηρίζει επίσης πλούσια χαρακτηριστικά, όπως διαδραστικά περιεχόμενα, αναζητήσεις πλήρους κειμένου και ενσωματώσεις πολυμέσων, γεγονός που το καθιστά ιδανικό για τη δημιουργία εύχρηστων και επαγγελματικών εγχειριδίων.

Παρά τα προτερήματά του, το GitBook έχει υποστεί κάποιες αλλαγές τα τελευταία χρόνια, που το απομάκρυναν από την παραδοσιακή του μορφή ως εργαλείου ανοιχτού κώδικα. Οι νεότερες εκδόσεις του επικεντρώνονται περισσότερο στο μοντέλο SaaS (Software as a Service), το οποίο απαιτεί συνδρομή για την πλήρη πρόσβαση στα χαρακτηριστικά του. Παρόλα αυτά, παραμένει ένα από τα πιο δημοφιλή εργαλεία για ομάδες που επιθυμούν να δημιουργήσουν γρήγορα επαγγελματικό περιεχόμενο.

Jekyll

Το Jekyll είναι ένα εργαλείο δημιουργίας στατικών ιστότοπων που δημιουργήθηκε από τον Tom Preston-Werner, τον συνιδρυτή του GitHub, το 2008. Έχει σχεδιαστεί για να μετατρέπει αρχεία Markdown ή HTML σε πλήρως λειτουργικούς ιστότοπους χωρίς την ανάγκη για βάσεις δεδομένων ή σύνθετα CMS (Content Management Systems). Το Jekyll βασίζεται στη γλώσσα προγραμματισμού Ruby και είναι ενσωματωμένο απευθείας στο GitHub μέσω του GitHub Pages, καθιστώντας το ιδανικό για φιλοξενία μικρών ιστοσελίδων ή προσωπικών blogs.

Η αρχική του ρύθμιση μπορεί να είναι πιο περίπλοκη από το MkDocs. Ενώ το Jekyll παρέχει πλήρη ελευθερία στη σχεδίαση και προσαρμογή ιστοσελίδων, η περίοδος εκμάθησης μπορεί να είναι μεγάλη για αρχάριους χρήστες, ιδιαίτερα αν δεν έχουν εμπειρία με τη γλώσσα Ruby ή τα Liquid templates που χρησιμοποιεί για τη δυναμική δημιουργία περιεχομένου.

Το Jekyll, απευθύνεται κυρίως σε προγραμματιστές που έχουν ήδη εξοικείωση με τη Ruby και τη διαμόρφωση αρχείων Liquid, ενώ επιτρέπει τη δημιουργία προσαρμοσμένων στατικών ιστότοπων. Κάθε εργαλείο έχει τα δικά του πλεονεκτήματα και μειονεκτήματα, και η επιλογή εξαρτάται από τις ανάγκες του εκάστοτε έργου.

Ένα από τα βασικά χαρακτηριστικά του Jekyll είναι η απλότητα του. Οι χρήστες γράφουν το περιεχόμενό τους σε αρχεία Markdown, ενώ το Jekyll αναλαμβάνει τη δημιουργία των αντίστοιχων HTML σελίδων. Αυτό εξαλείφει την ανάγκη για πολύπλοκες βάσεις δεδομένων ή backend διεργασίες, κάνοντας το Jekyll ιδιαίτερα ελκυστικό για προγραμματιστές που επιθυμούν ταχύτητα και αξιοπιστία. Το Jekyll υποστηρίζει επίσης την προσαρμογή θεμάτων, δίνοντας στους χρήστες τη δυνατότητα να δημιουργήσουν εξατομικευμένα σχέδια για τις ιστοσελίδες τους.

Επιπλέον, λόγω της ενσωμάτωσής του με το GitHub Pages, οι χρήστες μπορούν να δημοσιεύουν τις ιστοσελίδες τους με ένα απλό “git push”, γεγονός που το καθιστά απίστευτα εύελικο και εύχρηστο. Παρά την ευελιξία του, το Jekyll έχει ορισμένους περιορισμούς. Καθώς πρόκειται για εργαλείο στατικών ιστότοπων, δεν είναι κατάλληλο για εφαρμογές που απαιτούν διαδραστική λειτουργικότητα ή δυναμικό περιεχόμενο. Επιπλέον, η εξοικείωση με τη Ruby και τη ρύθμιση του εργαλείου μπορεί να είναι αποθαρρυντική για όσους δεν έχουν εμπειρία σε προγραμματισμό.

Sphinx

Το Sphinx είναι ένα ισχυρό εργαλείο που αναπτύχθηκε αρχικά για να εξυπηρετήσει την ανάγκη τεκμηρίωσης της γλώσσας προγραμματισμού Python. Δημιουργήθηκε από τον Georg Brandl το 2008, με σκοπό την παραγωγή καθαρής, ευανάγνωστης και καλαίσθητης τεκμηρίωσης. Από τότε, το Sphinx έχει εξελιχθεί και υποστηρίζει πλέον μια μεγάλη γκάμα εφαρμογών πέρα από τον κόσμο της Python, όπως τεχνική τεκμηρίωση, οδηγούς χρήστη και αρχεία οδηγιών.

Το Sphinx βασίζεται στη γλώσσα reStructuredText (reST), ένα ελαφρύ και εύελικο markup language που επιτρέπει στους χρήστες να γράφουν περιεχόμενο εύκολα, ενώ το εργαλείο αναλαμβάνει τη μετατροπή του σε HTML, PDF ή ακόμα και ePub. Ένα από τα σημαντικά πλεονεκτήματα του Sphinx είναι η εκτεταμένη υποστήριξη επεκτάσεων (extensions), οι οποίες δίνουν στους χρήστες τη δυνατότητα να προσαρμόζουν τη λειτουργία του. Για παράδειγμα, η επέκταση autodoc επιτρέπει την αυτόματη εξαγωγή τεκμηρίωσης από τον κώδικα Python, μειώνοντας έτσι τον χρόνο και την προσπάθεια που απαιτούνται για την καταγραφή τεχνικών λεπτομερειών.

Ένα άλλο χαρακτηριστικό που κάνει το Sphinx ιδιαίτερα δημοφιλές είναι η δυνατότητά του να δημιουργεί διασυνδεδεμένα έγγραφα. Αυτό σημαίνει ότι οι χρήστες μπορούν να δημιουργούν συνδέσμους μεταξύ διαφορετικών τμημάτων της τεκμηρίωσής τους, καθιστώντας την πιο διαδραστική και εύχρηστη. Επιπλέον, το Sphinx υποστηρίζει την εφαρμογή θεμάτων (themes), επιτρέποντας στους δημιουργούς να προσαρμόζουν την αισθητική της τεκμηρίωσης σύμφωνα με τις ανάγκες τους. Δημοφιλή θέματα, όπως το Read the Docs [13], έχουν κάνει τη χρήση του Sphinx ακόμα πιο προσιτή.

Παρά τη μεγάλη ευελιξία του, το Sphinx έχει και κάποιες προκλήσεις. Για παράδειγμα, η εκμάθηση της reStructuredText και η κατανόηση της ρύθμισης του εργαλείου μπορεί να φανεί

περίπλοκη στους αρχάριους. Ωστόσο, η εκτεταμένη τεκμηρίωση και η ενεργή κοινότητα χρηστών του Sphinx το καθιστούν ένα από τα καλύτερα εργαλεία στον τομέα της τεκμηρίωσης.

Συμπεράσματα

Συμπερασματικά, το MkDocs αποτελεί μια εξαιρετική επιλογή για όσους θέλουν ένα γρήγορο, απλό και αποτελεσματικό εργαλείο τεκμηρίωσης που βασίζεται στο Markdown. Είναι ιδανικό για έργα ανοιχτού κώδικα και προγραμματιστές που χρειάζονται μια εύχρηστη λύση για τη διαχείριση του υλικολογισμικού τους και την δημοσίευσή τους.

Η βασική διαφορά του MkDocs από τα άλλα εργαλεία τεκμηρίωσης έγκειται στον τρόπο με τον οποίο διαχειρίζεται τη δημιουργία και την οργάνωση του περιεχομένου. Ενώ το MkDocs επικεντρώνεται στη χρήση Markdown και διατηρεί μια σχετικά απλή ιεραρχία φακέλων, το Sphinx προσφέρει μεγαλύτερη ευελιξία με τη χρήση διαφόρων μορφών τεκμηρίωσης και προηγμένες επιλογές αναζήτησης. Το GitBook ξεχωρίζει για τη δυνατότητα συνεργασίας σε πραγματικό χρόνο και τη φιλική προς τον χρήστη διεπαφή, καθιστώντας το ιδανικό για ομάδες που θέλουν μια cloud-based λύση. Το Sphinx, διαθέτει μια πιο εκτενή βιβλιοθήκη οδηγιών και υποστηρίζεται από μια μεγάλη βάση χρηστών, ιδιαίτερα στον χώρο της επιστημονικής και τεχνικής τεκμηρίωσης. Το GitBook, παρόλο που είναι εύκολο στη χρήση, απαιτεί συχνά συνδρομή για πλήρη πρόσβαση στις λειτουργίες του, κάτι που μπορεί να αποτελεί περιοριστικό παράγοντα για μικρές ομάδες ή μεμονωμένους προγραμματιστές. Το Jekyll είναι εξαιρετικά προσαρμόσιμο, αλλά η έλλειψη αυτοματοποιημένων δυνατοτήτων και η ανάγκη για εκτεταμένη διαμόρφωση μπορεί να το καθιστούν λιγότερο ελκυστικό για χρήστες που αναζητούν μια γρήγορη και απλή λύση [12].

Εκτός από τα τεχνικά χαρακτηριστικά, η επιλογή εργαλείου εξαρτάται και από την κοινότητα υποστήριξης και την τεκμηρίωση που διαθέτει. Το MkDocs έχει μια ενεργή κοινότητα χρηστών και προγραμματιστών, οι οποίοι συνεισφέρουν στην ανάπτυξη νέων λειτουργιών και στην επίλυση προβλημάτων μέσω του GitHub.

Ωστόσο, για πιο προηγμένες ανάγκες, το Sphinx προσφέρει μεγαλύτερη προσαρμοστικότητα, ενώ το GitBook διευκολύνει τη συνεργασία μέσω cloud. Το Jekyll αποτελεί μια καλή επιλογή για όσους θέλουν περισσότερη ελευθερία στη διαμόρφωση του ιστότοπού τους, αλλά μπορεί να απαιτεί περισσότερη τεχνική εξειδίκευση.

Επομένως, η επιλογή μεταξύ αυτών των εργαλείων εξαρτάται από τις απαιτήσεις του έργου, τη γνώση του χρήστη και τον τρόπο με τον οποίο θέλει να διαχειριστεί την τεκμηρίωσή του. Ανάλογα με την πολυπλοκότητα και την κλίμακα της τεκμηρίωσης, το κάθε εργαλείο μπορεί να προσφέρει διαφορετικά οφέλη και να καλύψει διαφορετικές ανάγκες στον χώρο της τεκμηρίωσης λογισμικού.

Πίνακας 1. Σύγκριση των βασικών τεχνολογιών documentation.

Χαρακτηριστικό	MkDocs	Jekyll	Sphinx	Gitbook
<i>Κύρια χρήση</i>	Τεκμηρίωση έργων	Blogs & γενικής χρήσης στατικές σελίδες	Τεκμηρίωση έργων σε Python	Τεκμηρίωση & εκπαιδευτικό περιεχόμενο
<i>Τεχνολογία</i>	Python	Ruby	Python	JavaScript
<i>Δυσκολία ρύθμισης</i>	Εύκολη	Μέτρια	Μέτρια έως υψηλή	Πολύ εύκολη
<i>Μορφή περιεχομένου</i>	Markdown	Markdown + Liquid	reStructuredText	Markdown
<i>Αναζήτηση</i>	Ενσωματωμένη	Μέσω plugins ή εξωτερικών εργαλείων	Ενσωματωμένη	Ενσωματωμένη
<i>Φιλοξενία (Hosting)</i>	GitHub Pages	GitHub Pages	ReadTheDocs, GitHub Pages	GitBook.com ή custom
<i>Αδύναμα σημεία</i>	Περιορισμένη ευελιξία εκτός τεκμηρίωσης	Περίπλοκο για καθαρά τεκμηριωτικά έργα	Απότομη καμπύλη μάθησης	Περιορισμένη παραμετροποίηση
<i>Θέματα (Themes)</i>	Απλά, παραμετροποιήσιμα	Πολύ παραμετροποίηση	Πολύ ισχυρά, αλλά πιο περίπλοκα	Περιορισμένα

Κεφάλαιο 3 - Δημιουργία εκπαιδευτικού υλικού με το MkDocs και το Material theme

Εφαρμογή του MkDocs σε πραγματικό σενάριο

Εγκατάσταση του λογισμικού

Στόχος της παρούσας εργασίας και εκεί που θα εστιάστεί περισσότερο είναι η μετατροπή του βιβλίου *"Μια σύγχρονη προσέγγιση στη γλώσσα C"* σε στατική ιστοσελίδα μέσω της χρήσης του MkDocs και την χρήση του θέματος Material. Για την δημιουργία της ιστοσελίδας βασισμένου στο MkDocs και το θέμα Material, ακολουθείται μια λεπτομερής διαδικασία που περιλαμβάνει την εγκατάσταση εργαλείων για τη διαμόρφωση του έργου όπως το Visual Studio Code της Microsoft, τη διαχείριση με το Git και την τελική ανάρτηση στο GitHub Pages. Το αρχικό κομμάτι ξεκινά με την εγκατάσταση της γλώσσας προγραμματισμού Python και συγκεκριμένα ή με την έκδοση από την 3.7 ή μια πιο νεότερη.

Για την διευκόλυνση και την σωστή λειτουργία του project θα χρησιμοποιήσουμε την τεχνολογία της Python με όνομα εικονικό περιβάλλον (virtual environment). Η τεχνολογία αυτή επιτρέπει την απομόνωση των εξαρτήσεων του έργου, το οποίο δημιουργείται μέσω της εντολής `"python -m venv venv"` στο powershell και όντας μέσα στο φάκελο του project [4]. Με την δημιουργία και ενεργοποίηση του εικονικού περιβάλλοντος, όλες οι επόμενες εγκαταστάσεις πακέτων πραγματοποιούνται τοπικά, χωρίς να επηρεάζονται άλλα projects στο σύστημα.

Στη συνέχεια, γίνεται εγκατάσταση του MkDocs και του Material for MkDocs μέσω του διαχειριστή πακέτων pip της Python, με την εντολή `"pip install mkdocs-material"`. Η επιτυχής εγκατάσταση αυτών των εργαλείων παρέχει όλα τα απαραίτητα μέσα για τη δημιουργία και την ανάπτυξη του documentation. Η εντολή στα Windows για την ενεργοποίηση του virtual environment είναι `"venv\Scripts\Activate.ps1"`, λόγω όμως της ασφάλειας των Windows θα λάβουμε error σχετικά με τα execution policies. Για να ξεπεραστεί αυτό το πρόβλημα με την ασφάλεια τρέχουμε την εξής εντολή `"Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope Process"`. Με την τιμή του Process στο γνώρισμα του Scope στο PowerShell, μπορούμε να εξασφαλίσουμε ότι η διαδικασία θα εκτελείται μόνο όσο είναι ανοιχτή η κονσόλα του PowerShell. Αν κλείσουμε το παράθυρο του PowerShell, τότε και η διαδικασία θα τερματιστεί αυτόματα, αφού το scope είναι συνδεδεμένο με τη διάρκεια ζωής του PowerShell περιβάλλοντος. Αυτός ο μηχανισμός είναι χρήσιμος όταν θέλουμε να διασφαλίσουμε ότι μια διαδικασία εκτελείται προσωρινά και δεν παραμένει ενεργή μετά το κλείσιμο της συνεδρίας PowerShell. Αν τρέξουμε πάλι την εντολή για ενεργοποίηση του εικονικού περιβάλλοντος θα δούμε ότι στην αρχή της γραμμής θα εμφανιστεί με πράσινα γράμματα μέσα σε παρενθέσεις το όνομα του venv που έχουμε ορίσει, στην προκειμένη περίπτωση venv.


```

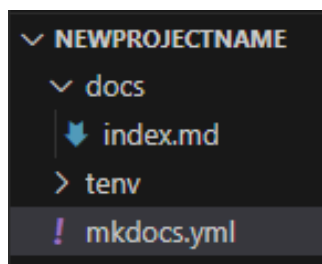
PS C:\Users\John\Documents\PT_Hia_Kee\kallipos-C>
PS C:\Users\John\Documents\PT_Hia_Kee\kallipos-C> ..\venv\Scripts\Activate.ps1
..\venv\Scripts\Activate.ps1 : File C:\Users\John\Documents\PT_Hia_Kee\venv\Scripts\Activate.ps1 cannot be loaded because running scripts is disabled on this system. For more information, see about_Execution_Policies at https:/go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ ..\venv\Scripts\Activate.ps1
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
PS C:\Users\John\Documents\PT_Hia_Kee\kallipos-C>
PS C:\Users\John\Documents\PT_Hia_Kee\kallipos-C>
PS C:\Users\John\Documents\PT_Hia_Kee\kallipos-C> Set-ExecutionPolicy RemoteSigned -Scope Process
PS C:\Users\John\Documents\PT_Hia_Kee\kallipos-C>
PS C:\Users\John\Documents\PT_Hia_Kee\kallipos-C>
PS C:\Users\John\Documents\PT_Hia_Kee\kallipos-C> ..\venv\Scripts\Activate.ps1
(venv) PS C:\Users\John\Documents\PT_Hia_Kee\kallipos-C>
(venv) PS C:\Users\John\Documents\PT_Hia_Kee\kallipos-C>
(venv) PS C:\Users\John\Documents\PT_Hia_Kee\kallipos-C>

```

Εικόνα 1. Το σφάλμα λόγω της ασφάλειας των Windows.

Visual Studio Code και ιεράρχηση των αρχείων

Το βασικό εργαλείο με το οποίο θα τροποποιούμε τα αρχεία θα είναι το Visual Studio Code. Αφού έχουμε ενεργοποιημένο το `venv` (εικονικό περιβάλλον) θα ανοίξουμε το φάκελο μέσα από το ήδη ανοιχτό powershell command με την εντολή “code .”. Αυτό ανοίγει το Visual Studio Code στο φάκελο που έχουμε δημιουργήσει για το project, στην προκειμένη περίπτωση είναι ο φάκελος `kallipos-C`. Για να ξεκινήσει η δημιουργία της βασικής δομής του project είναι απαραίτητη η εντολή “`mkdocs new .`”, με αυτήν την εντολή δημιουργούνται αυτόματα τα αρχεία “`mkdocs.yml`” και ο φάκελος `docs` που μέσα περιέχει το αρχείο “`index.md`”.



Εικόνα 2. Τα αρχεία που δημιουργούνται μετά την εκτέλεση της εντολής “`mkdocs new .`”.

Το αρχείο `mkdocs.yml` αποτελεί βασικό παράγοντα για το MkDocs καθώς η παρουσία του και η σωστή διαμόρφωσή του είναι ουσιώδης για το project. Η δομή, η εμφάνιση, οι λειτουργικές επεκτάσεις καθώς και η συμπεριφορά του τελικού site καθορίζονται από το αρχείο αυτό. Η κατάληξη του αρχείου (.yml) δηλώνει ότι είναι γραμμένο στην γλώσσα YAML η οποία έχει σχεδιαστεί και χρησιμοποιείται ευρέως για ρυθμίσεις εφαρμογών. Σε ένα project MkDocs, το αρχείο αυτό βρίσκεται στον root φάκελο του project και διαβάζεται αυτόματα κατά την εκτέλεση της εντολής “`mkdocs serve`”.

Εσωτερικά το αρχείο περιλαμβάνει κάποια πεδία και το καθένα εξυπηρετεί συγκεκριμένες λειτουργίες για το project. Από το όνομα του site που καθορίζεται από το πεδίο “`site_name`”, το οποίο φαίνεται στον τίτλο της σελίδας αλλά και στην γραμμή πλοήγησης. Το πεδίο “`nav`” είναι υπεύθυνο για την πλοήγηση στο site.

Πρόκειται για την καταγραφή και ιεράρχηση των αρχείων με κατάληξη .md που περιλαμβάνονται στο project και επιτρέπει την ομαδοποίηση τους σε ενότητες και υποενότητες σύμφωνα με τον χρήστη. Βασικό στοιχείο στο εκάστοτε project είναι το θέμα εμφάνισης του site, το οποίο καθορίζεται από το πεδίο “theme”. Το συγκεκριμένο πεδίο επηρεάζει τη γραφιστική απεικόνιση, τις γραμματοσειρές αλλά και την υποστήριξη της εναλλαγής του θέματος σε dark theme. Για το συγκεκριμένο project το θέμα που έχει επιλεγθεί είναι το Material for MkDocs.

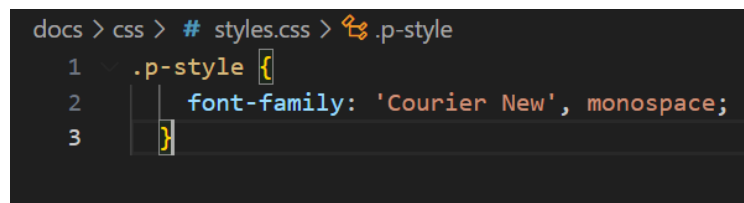
Η τεχνολογία MkDocs μας επιτρέπει να χρησιμοποιούμε και επεκτάσεις στο project τα οποία επιτρέπουν την προηγμένη δυνατότητα μορφοποίησης όπως σύνθετες σημειώσεις, αυτόματο πίνακα περιεχομένων, λίστες και πίνακες αλλά και την εισαγωγή κάποιου JavaScript tool. Ένα από αυτά είναι και το Mermaid το οποίο με τη χρήση κάποιου κειμένου σε Markdown μπορεί να αποδώσει προσαρμοσμένα διαγράμματα, γραφήματα και οπτικοποιήσεις. Οι παραπάνω παραμετροποιήσεις μπορούν να γίνουν με επεξεργασία του πεδίου “markdown_extensions” [15]. Η παρακάτω εικόνα απεικονίζει μερικές από τις επεκτάσεις που αξιοποιήθηκαν στο συγκεκριμένο project.

```
markdown_extensions:
  - abbr
  - admonition
  - attr_list
  - def_list
  - pymdownx.caret
  - pymdownx.mark
  - pymdownx.tilde
  - footnotes
  - toc:
    | title: Πίνακας περιεχομένων
  - markdown_grid_tables
  - md_in_html
  - pymdownx.mark
  - tables
  - pymdownx.superfences:
    | custom_fences:
    |   - name: mermaid
    |     class: mermaid
    |     format: !!python/name:pymdownx.superfences.fence_code_format
  - pymdownx.tabbed:
    | alternate_style: true
  - pymdownx.details
  - pymdownx.inlinehilite
  - pymdownx.arithmatex:
    | generic: true
  - pymdownx.highlight
  - pymdownx.tasklist:
    | custom_checkbox: true
```

Εικόνα 3. Οι επεκτάσεις markdown που χρησιμοποιήθηκαν στο project.

Με την προσθήκη του πεδίου “plugins” μπορούμε να καθορίσουμε τα πρόσθετα που θέλουμε να εισαχθούν στο project. Η χρήση αυτού του πεδίου επεκτείνει τη λειτουργικότητα του MkDocs προσθέτοντας δυνατότητες όπως την μπάρα αναζήτησης, τη χρήση του awesome-pages αλλά και το plugin για την εναλλαγή μεταξύ του light και dark theme.

Η ενσωμάτωση προσαρμοσμένου κώδικα CSS ή JavaScript στο site καθίσταται μέσω του πεδίου extra_css. Στο περιεχόμενο του βιβλίου χρησιμοποιούνται αρκετές λέξεις με διαφορετικό φόντο από τις υπόλοιπες για να τονιστεί η χρήση τους σε προγραμματιστικό πλαίσιο. Για τις ανάγκες του site, κρίθηκε απαραίτητο να προσαρμοστεί η εμφάνιση αυτών των λέξεων, χρησιμοποιώντας μία πιο χαρακτηριστική γραμματοσειρά, όπως η Courier New. Η προσαρμογή αυτή υλοποιήθηκε με τη δημιουργία ενός αρχείου CSS με όνομα “style.css”, στο οποίο ορίζεται μια ειδική κλάση με όνομα “.p-style”. Η κλάση αυτή εφαρμόζει τη γραμματοσειρά και το επιθυμητό φόντο στις λέξεις που την χρησιμοποιούν. Η χρήση της στο περιεχόμενο γίνεται με την εισαγωγή της λέξης μέσα σε μία HTML ετικέτας π.χ. “λέξη” [16]. Με αυτόν τον τρόπο, επιτυγχάνεται σαφής διάκριση των όρων προγραμματιστικού χαρακτήρα μέσα στο κείμενο.

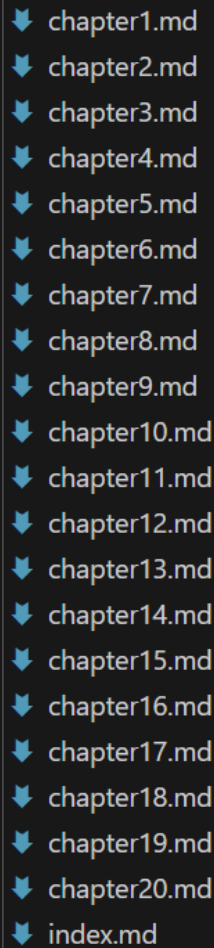


```
docs > css > # style.css > .p-style
1  .p-style {
2      font-family: 'Courier New', monospace;
3  }
```

Εικόνα 4. Το αρχείο style.css.

Το περιεχόμενο του site μπορεί να τμηματοποιηθεί σε ξεχωριστές σελίδες διευκολύνοντας έτσι την διαχείρισή του από τον χρήστη. Αυτή η θεματική κατάτμηση επιτυγχάνεται μέσω της χρήσης διαφορετικών αρχείων Markdown, τα οποία αναγνωρίζονται από την κατάληξή τους “.md”. Κατά την δημιουργία ενός καινούργιου project ένα από τα αρχεία που παράγεται είναι και το “index.md” μέσα στον φάκελο “docs”.

Κατά την διαδικασία δημιουργίας του site το αρχείο μετατρέπεται στη σελίδα που αντιστοιχεί στη ρίζα, δηλαδή το root “/”. Το “index.md” δηλώνεται αυτόματα από το MkDocs ως το αρχείο της βασικής πλοήγησης του documentation, εκτός αν οριστεί διαφορετικά μέσω του πεδίου “nav” στο αρχείο “mkdocs.yml”. Το περιεχόμενό του παρέχει στο χρήστη μια εισαγωγή στο δομημένο documentation. Συνήθως περιλαμβάνει διάφορα σημαντικά πεδία για το site όπως τον τίτλο, μια γενική εισαγωγή, κάποια χαρακτηριστικά για την πλοήγηση ή ακόμη και αυτούσιο το περιεχόμενο προς έκδοση.



- chapter1.md
- chapter2.md
- chapter3.md
- chapter4.md
- chapter5.md
- chapter6.md
- chapter7.md
- chapter8.md
- chapter9.md
- chapter10.md
- chapter11.md
- chapter12.md
- chapter13.md
- chapter14.md
- chapter15.md
- chapter16.md
- chapter17.md
- chapter18.md
- chapter19.md
- chapter20.md
- index.md

Εικόνα 5. Τα αρχεία markdown, κάθε ένα για ένα διαφορετικό κεφάλαιο.

Η εντολή “mkdocs serve” για τοπική ανάπτυξη

Μία δυνατότητα που μας δίνει το MkDocs είναι και η τοπική ανάπτυξη του site χωρίς να γίνει hosting σε κάποιο domain. Αυτή η διαδικασία εξυπηρετεί την άμεση μετατροπή των αρχείων με κατάληξη “.md” σε html αρχεία για την σύνθεση του site. Με αυτόν τον τρόπο μπορούμε να δούμε πως θα φαίνεται η τεκμηρίωση πριν την ανεβάσουμε σε κάποιον server ή σε πλατφόρμες όπως είναι αυτή του Github pages. Η εντολή “mkdocs serve” χρησιμοποιείται για να ξεκινήσει ένας τοπικός διακομιστής, ο οποίος επιτρέπει την άμεση προεπισκόπηση των αλλαγών που κάνουμε στα αρχεία Markdown και στο configuration αρχείο “mkdocs.yml”. Για να τρέξουμε την εντολή “mkdocs serve” ανοίγουμε ένα τερματικό και πηγαίνουμε στον φάκελο του project. Όταν θα έχουμε φτάσει στον φάκελο που περιέχει το αρχείο “mkdocs.yml” τρέχουμε την εντολή “mkdocs serve”. Το MkDocs ξεκινά ένα local server και εμφανίζει στο τερματικό μια διεύθυνση URL, που συνήθως είναι η loopback ip `http://127.0.0.1:8000/`. Στην Εικόνα 10 βλέπουμε την εκτέλεση αυτής της εντολής και την επιστροφή.

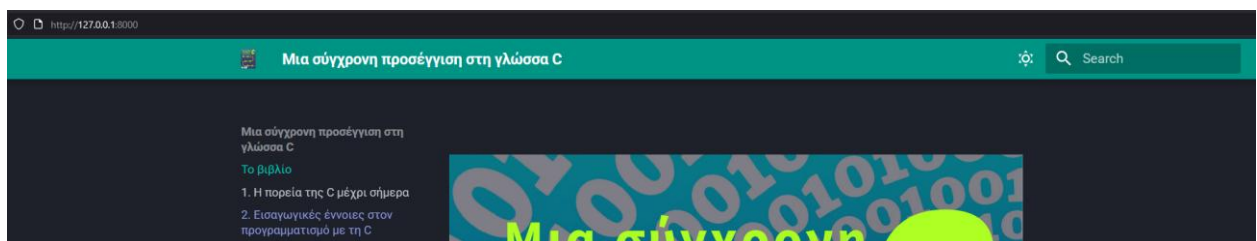
```

PS C:\Users\John\Documents\PT_Hia_Kee\kallipos-C> ..\venv\Scripts\Activate.ps1
(venv) PS C:\Users\John\Documents\PT_Hia_Kee\kallipos-C> mkdocs serve
INFO      - Building documentation...
INFO      - Cleaning site directory
INFO      - Documentation built in 17.69 seconds
INFO      - [23:48:32] Watching paths for changes: 'docs', 'mkdocs.yml'
INFO      - [23:48:32] Serving on http://127.0.0.1:8000/
INFO      - [23:48:57] Browser connected: http://127.0.0.1:8000/

```

Εικόνα 6. Εκτέλεση της εντολής "mkdocs serve"

Για την προβολή της ιστοσελίδας αρκεί να ανοίξουμε ένα φυλλομετρητή (browser) και να βάλουμε την διεύθυνση που παίρνουμε από την έξοδο της εντολής. Σε περίπτωση αλλαγής σε κάποιο αρχείο του project αν έχουμε ενεργοποιήσει στον επεξεργαστή την επιλογή για αυτόματη αποθήκευση τότε όποιες αλλαγές γίνονται στα αρχεία Markdown, εμφανίζονται αυτόματα με live reload, χωρίς να χρειάζεται να επανεκκινήσουμε τον server. Τα πλεονεκτήματα αυτού είναι η άμεση προεπισκόπηση των αλλαγών αλλά και ποια θα είναι η επίπτωση στο site και η ευκολία για τοπικό έλεγχο και διόρθωση πριν το τελικό push στο Github. Στην Εικόνα 7 Εικόνα 10 υπάρχει ο ιστότοπος όπως εμφανίζεται μετά το τοπικό deploy.



Εικόνα 7. Προβολή site μετά το τοπικό deploy

Μετά την προβολή του site στην περίπτωση που αλλάξουμε κάτι στα βασικά αρχεία βλέπουμε ότι ο server το αντιλαμβάνεται και ξεκινάει την διαδικασία του rebuild στο site.

```

INFO      - [23:48:57] Browser connected: http://127.0.0.1:8000/
INFO      - [00:57:36] Detected file changes
INFO      - Building documentation...
INFO      - Documentation built in 25.31 seconds
INFO      - [00:58:04] Reloading browsers
INFO      - [00:58:05] Browser connected: http://127.0.0.1:8000/

```

Εικόνα 8. Rebuild του site σε περίπτωση αλλαγής

Στην περίπτωση που θέλουμε να σταματήσουμε φτάνει απλά να πατήσουμε "Ctrl+c" και βλέπουμε το παρακάτω μήνυμα.

```
INFO - Documentation built in 25.31 seconds
INFO - [00:58:04] Reloading browsers
INFO - [00:58:05] Browser connected: http://127.0.0.1:8000/
INFO - Shutting down...
(venv) PS C:\Users\John\Documents\PT_Hia_Kee\kallipos-C>
```

Εικόνα 9. Διακοπή του τοπικού deploy

Το Github και η χρήση του για hosting της σελίδας.

Ένα ευρέως γνωστό εργαλείο στο χώρο του προγραμματισμού είναι και το Github. Το Github είναι μια πλατφόρμα που επιτρέπει στους προγραμματιστές να αναπτύσσουν, να μοιράζονται και να διαχειρίζονται τον κώδικά τους. Η δημιουργία του το 2008 από τους Tom Preston-Werner, Chris Wanstrath, PJ Hyett και Scott Chacon προέκυψε από την ανάγκη για ένα καλύτερο και πιο φιλικό προς το χρήστη τρόπο για τη διαχείριση του κώδικα που θα δημιουργείται μέσω του Git [7]. Αναπτυγμένο από το δημιουργό του linux, Linus Torvalds, το Git είναι ένα σύστημα ελέγχου των εκδόσεων του κώδικα. Με την ανάπτυξη του Github βελτιώθηκαν περισσότερο οι οργανωμένες διαδικασίες για την παρακολούθηση και τη διαχείριση των αλλαγών του κώδικα που απαιτούνταν στην ανάπτυξη λογισμικού. Καθώς το Git παρέχει ένα κατακευματισμένο τρόπο διαχείρισης ιστορικού εκδόσεων επιτρέπει σε ομάδες να συνεργάζονται με αποτέλεσμα να παρακολουθούν τις αλλαγές σε όλα τα στάδια ανάπτυξης του λογισμικού. Το Github βασίζεται στο Git, αλλά προσφέρει ένα σύνολο υπηρεσιών που καθιστούν την ανάπτυξη και τη συντήρηση λογισμικού πιο προσβάσιμη και παραγωγική. Η χρήση του Github βασίζεται στα αποθετήρια (repositories) στα οποία αποθηκεύεται ο κώδικας, συνήθως είναι ένα αποθετήριο για κάθε project. Στα αποθετήρια υπάρχει ο εξής διαχωρισμός τον οποίο πρέπει να αποφασίσει ο ιδιοκτήτης, αν το αποθετήριο έχει την σημασία για δημόσιο, το οποίο δίνει την δυνατότητα για να γίνει χρήση και να εξερευνηθεί, ή ιδιωτικό για έναν πιθανό project το οποίο δεν έχει ολοκληρωθεί ακόμη.

Ένα από τα πιο ισχυρά χαρακτηριστικά του GitHub είναι η δυνατότητα δημιουργίας παρακλαδιών (branches), που επιτρέπουν την παράλληλη ανάπτυξη χαρακτηριστικών χωρίς να διαταράσσεται η κύρια έκδοση (main) του λογισμικού. Με την χρήση των pull request οι χρήστες μπορούν να διορθώσουν, να κάνουν επισημάνσεις πάνω στον κώδικα ή να συγχωνεύσουν (merge) τις αλλαγές του κώδικα με την κύρια έκδοση. Επιπροσθέτως ένα από τα χαρακτηριστικά του το οποίο το κάνει να ξεχωρίζει είναι και η καταγραφή σφαλμάτων (bugs), τεκμηρίωση του project και αυτοματοποιημένες ενέργειες όπως έλεγχος του κώδικα ή και deployment μέσω του GitHub actions [19]. Ένα βασικό στοιχείο είναι και η υποστήριξη για την χρήση web hosting μέσω του GitHub pages, κάτι που παρατηρούμε και στο project που μελετάμε στην παρούσα εργασία.

Το Github έχει ένα ευρύ φάσμα χρηστών αλλά και υποστηριζόμενων περιβαλλόντων. Χρήστες που δημιουργούν ατομικά project και χρειάζονται ένα αποθετήριο για το project τους το οποίο είναι αποθηκευμένο στο Github. Πανεπιστημιακά ομαδικά project τα οποία μπορούν να παρακολουθούνται από τους καθηγητές, για συνεχή ενημέρωση του κώδικα σε πραγματικό χρόνο από όλους τους φοιτητές που συμμετέχουν. Παρατηρείται ότι στον ακαδημαϊκό χώρο, το GitHub χρησιμοποιείται όλο και συχνότερα για την υλοποίηση και παρουσίαση φοιτητικών εργασιών, ερευνητικών πειραμάτων αλλά και μαθημάτων με προσέγγιση τις ομαδικές εργασίες. Καθώς και εταιρικά περιβάλλοντα μιας και η χρήση του εξατομικευμένου issuing σε

κομμάτια κώδικα στον εκάστοτε υπάλληλο είναι κάτι το οποίο εξυπηρετεί μεγάλες υποδομές. Ταυτόχρονα η χρήση του για project ανοιχτού κώδικα είναι δημοφιλής και αποτελεί κύριο τρόπο συμμετοχής και συνδιαλλαγής σε τέτοια project. Εταιρείες όπως η Facebook ή η Google κρατάνε κάποιες τεχνολογίες τους στο Github καθώς επιτρέπουν την χρήση αυτών σε άλλα projects ανοιχτού κώδικα.

Χρήση του Github

Το Github πέρα από την ιδιότητά του για αποθήκευση του κώδικα στο cloud, διαθέτει ένα πλήρες πακέτο εργαλείων που μπορούν να χρησιμοποιηθούν καθ' όλη τα στάδια δημιουργίας ενός project. Ξεκινώντας από την αρχική συγγραφή και το documentation του project μέχρι και τις εκάστοτε δοκιμές που γίνονται αλλά και την συνεργασία και την τελική έκδοση του κώδικα. Όπως προαναφέραμε το Github βασίζεται πάνω στην τεχνολογία του Linus Torvalds δημιουργό του linux το Git. Το Git είναι ένα σύστημα εκδόσεων που ελέγχει και παρακολουθεί τις αλλαγές που γίνονται στον κώδικα, επιτρέποντας την επιστροφή σε παλαιότερη έκδοση, την δημιουργία καινούργιου branch, με σκοπό την σταθερότητα της βασικής ροής γνωστή ως main branch. Οι βασικές διαδικασίες που μπορεί να χρησιμοποιήσει κάποιος στο Github είναι οι εξής:

1) Δημιουργία αποθετηρίου

Ξεκινώντας ένα αποθετήριο (repository) στο Github με τα κατάλληλα labels από το τι γλώσσες θα χρησιμοποιηθούν μέχρι το αν θα είναι ιδιωτικό ή δημόσιο. Με ένα καινούργιο αποθετήριο δηλώνουμε ότι μέσα εκεί υπάρχει ένα ξεχωριστό project το οποίο θα λειτουργεί ως κεντρικός αποθηκευτικός χώρος για τον κώδικα.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * giannisvlg / Repository name * example
 example is available.

Great repository names are short and memorable. Need inspiration? How about [refactored-happiness](#) ?

Description (optional)
 This is an example.

☒ **Public**
 Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
 You choose who can see and commit to this repository.

Initialize this repository with:
☐ Add a README file
 This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore
 .gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license
 License: GNU General Public License v3.0

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

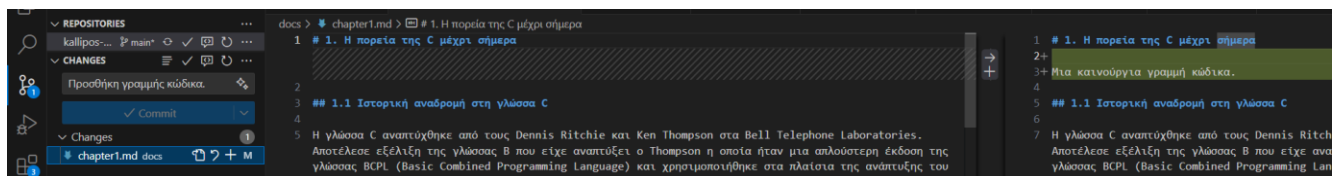
i You are creating a public repository in your personal account.

[Create repository](#)

Εικόνα 10. Δημιουργία καινούργιου αποθετηρίου.

2) Commit τις αλλαγές

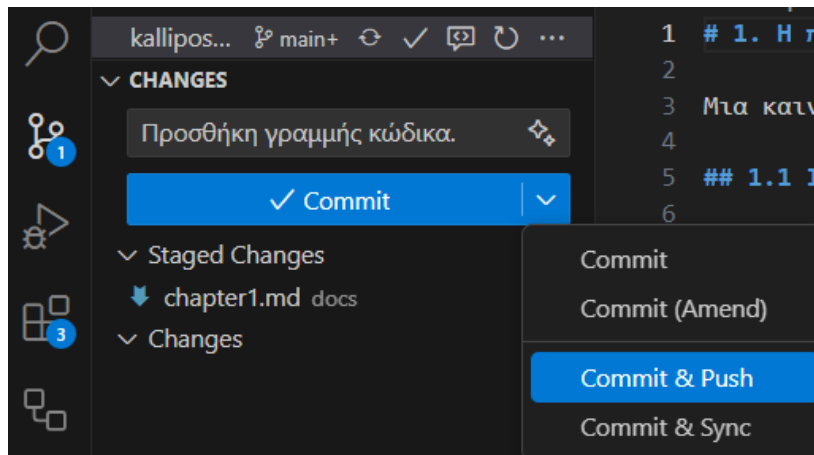
Κατά την διαδικασία της δημιουργίας του κώδικα θα υπάρχουν και αλλαγές οι οποίες αποθηκεύονται μέσω της εντολής “commit” και συνδέεται συνήθως με ένα σύντομο μήνυμα για την αλλαγή.



Εικόνα 11. Commit σε προσθήκη νέας γραμμής.

3) Push των αλλαγών που έχουν γίνει commit

Οι αλλαγές που έχουν γίνει commit αποθηκεύονται τοπικά και γίνεται μία ομαδοποίηση ώστε στην εντολή “push” όλες οι αλλαγές που έχουν γίνει commit να σταλούν στον server του Github.



Εικόνα 12. Commit and Push σε staged αλλαγή.

4) Αίτημα Pull

Μέσω του pull request μπορεί να ελεγχθεί να σχολιαστεί και να γίνει συγχωνεύσει με το main branch.

Το αρχείο ci.yml

Το MkDocs είναι ένα στατικό εργαλείο δημιουργίας τεκμηρίωσης το οποίο επιτρέπει τη συγγραφή περιεχομένου σε Markdown και την απόδοσή του σε μορφή webpage. Με την χρήση του Github actions αυτοματοποιείται η διαδικασία για την εξάλειψη πρόσθετου χρόνου. Το κεντρικό αρχείο που καθορίζει αυτή τη διαδικασία είναι το αρχείο ci.yml, το οποίο ορίζεται στον φάκελο .github/workflows/.

Το ci.yml είναι ένα αρχείο το οποίο περιγράφει την διαδικασία η οποία εκτελείται αυτόματα στον server του Github όταν πληρούνται συγκεκριμένες προϋποθέσεις όπως push του κώδικα. Το συγκεκριμένο αρχείο χρησιμοποιείται για την εγκατάσταση των απαραίτητων πακέτων python αλλά και MkDocs, δημιουργεί την τεκμηρίωση και θα αναπτύξει (deploy) την τεκμηρίωση στο GitHub pages [14]. Μέσα στο αρχείο γίνονται συγκεκριμένες δηλώσεις που χρειάζονται για να γίνει η σωστή και πλήρης ανάπτυξη του project σε webpage. Παρακάτω βλέπουμε το αρχείο ci.yml που χρησιμοποιήθηκε σε αυτό το project.

```

.github > workflows > ci.yml
1  name: ci
2  on:
3    push:
4      branches:
5        - master
6        - main
7  permissions:
8    contents: write
9  jobs:
10   deploy:
11     runs-on: ubuntu-latest
12     steps:
13       - uses: actions/checkout@v4
14       - name: Configure Git Credentials
15         run: |
16           git config user.name github-actions[bot]
17           git config user.email 41898282+github-actions[bot]@users.noreply.github.com
18       - uses: actions/setup-python@v5
19       with:
20         python-version: 3.x
21       - run: echo "cache_id=$(date --utc '+%V')" >> $GITHUB_ENV
22       - uses: actions/cache@v4
23       with:
24         key: mkdocs-material-${{ env.cache_id }}
25         path: .cache
26         restore-keys: |
27           mkdocs-material-
28       - run: pip install mkdocs-material
29       - run: pip install mkdocs-awesome-pages-plugin
30       - run: pip install markdown-grid-tables
31       - run: pip install mkdocs-glightbox
32       - run: mkdocs gh-deploy --force

```

Εικόνα 13. Το αρχείο ci.yml.

Η μεταβλητή name: ci ορίζει το όνομα της ροής εργασίας (workflow) σε ci (Continuous Integration). Ακριβώς κάτω από αυτό έχουμε το έναυσμα (trigger) δηλαδή το πότε θα εκτελείται η ροή. Στο συγκεκριμένο βλέπουμε ότι ξεκινάει όταν γίνεται push ένα από τα branch με όνομα master ή main. Αυτό καλύπτει περιπτώσεις όπου ένα έργο μπορεί να έχει οποιοδήποτε από τους δύο κεντρικούς κλάδους. Στην συνέχεια για να έχει την άδεια να γράφει στο repository καθώς είναι απαραίτητο για να εκτελεσθούν οι εντολές “gh-deploy” η οποία ενημερώνει το branch gh-pages. Το όρισμα ενός job με όνομα deploy το οποίο εκτελείται σε μια εικονική υποδομή με linux λειτουργικό και πιο συγκεκριμένα ubuntu-latest είναι βασικό για ένα project που έχει python builds. Αφού έχουμε καθορίσει τις βασικές επιλογές θα συμπληρώσουμε τα βήματα που θέλουμε να εκτελέσει το συγκεκριμένο workflow. Η αρχική εντολή κατεβάζει τον πηγαίο κώδικα του repository στο runner. Με τον όρο runner το Github ορίζει ένα εικονικό μηχάνημα το οποίο θα τρέξει τα βήματα που έχουμε ορίσει στο αρχείο ci.yml. Για να μπορέσει όμως ο runner να δουλέψει σωστά με τα αρχεία του project θα πρέπει πρώτα να έχει πρόσβαση σε αυτά. Αυτό γίνεται με την εντολή “- uses: actions/checkout@v4”. Στο βήμα αυτό χρησιμοποιείται το Github actions με το οποίο κάνει clone το repository στον τοπικό φάκελο του runner και μεταφέρει όλα τα αρχεία του branch.

```
deploy
succeeded 3 weeks ago in 19s

▼ ✓ Set up job

1 Current runner version: '2.324.0'
2 ▼ Operating System
3 Ubuntu
4 24.04.2
5 LTS
6 ▼ Runner Image
7 Image: ubuntu-24.04
8 Version: 20250511.1.0
9 Included Software: https://github.com/actions/runner-images/blob/ubuntu24/20250511.1/images/ubuntu/Ubuntu2404-Read
10 Image Release: https://github.com/actions/runner-images/releases/tag/ubuntu24%2F20250511.1
11 ▶ Runner Image Provisioner
13 ▶ GITHUB_TOKEN Permissions
16 Secret source: Actions
17 Prepare workflow directory
18 Prepare all required actions
19 Getting action download info
20 ▶ Download immutable action package 'actions/checkout@v4'
24 ▼ Download immutable action package 'actions/setup-python@v5'
25 Version: 5.6.0
26 Digest: sha256:0b35a0c11c97499e4e0576589036d450b9f5f9da74b7774225b3614b57324404
27 Source commit SHA: a26af69be951a213d495a4c3e4e4022e16d87065
28 ▶ Download immutable action package 'actions/cache@v4'
32 Complete job name: deploy

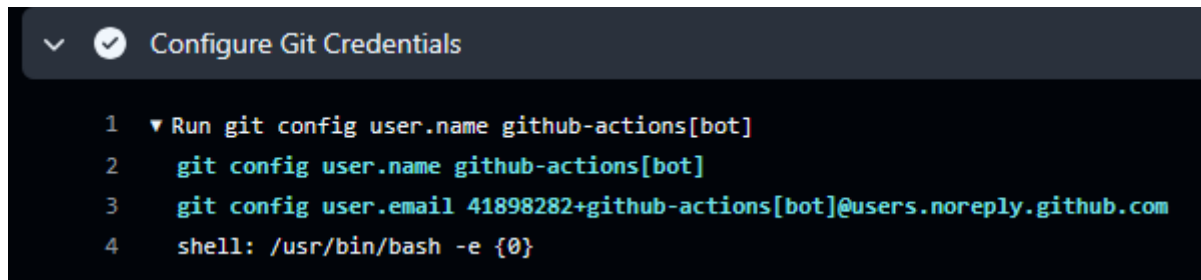
▼ ✓ Run actions/checkout@v4

1 ▶ Run actions/checkout@v4
16 Syncing repository: giannisvlg/kallipos-C
17 ▶ Getting Git version info
21 Temporarily overriding HOME='/home/runner/work/_temp/c92428e5-bce4-466d-821f-1e270c45098e' before making global git
22 Adding repository directory to the temporary git global config as a safe directory
23 /usr/bin/git config --global --add safe.directory /home/runner/work/kallipos-C/kallipos-C
24 Deleting the contents of '/home/runner/work/kallipos-C/kallipos-C'
25 ▶ Initializing the repository
39 ▶ Disabling automatic garbage collection
41 ▶ Setting up auth
47 ▶ Fetching the repository
51 ▶ Determining the checkout info
52 /usr/bin/git sparse-checkout disable
53 /usr/bin/git config --local --unset-all extensions.worktreeConfig
54 ▶ Checking out the ref
58 /usr/bin/git log -1 --format=%H
59 6701f41dd645ebbe6e21c0784978399ab935d576
```

Εικόνα 14. Log file κατά το deploy.

Το παρακάτω code block ρυθμίζει τις πληροφορίες χρήστη του Git στον runner που εκτελεί το workflow. Αυτό είναι απαραίτητο ώστε να μπορεί να εκτελέσει Git εντολές που απαιτούν αναγνώριση, όπως το “git commit” και το “git push” [18]. Με την εντολή “git config user.name github-actions[bot]” ορίζουμε το όνομα χρήστη του Git. Αυτό είναι το προεπιλεγμένο όνομα για το bot που χρησιμοποιεί το GitHub actions για τις αυτόματες ενέργειες. Με αυτή την αλλαγή όποτε δημιουργείται ένα commit από το workflow, θα φαίνεται σαν να έχει δημιουργηθεί από το github-actions[bot]. Σε ίδιο μοτίβο γίνεται και το όρισμα για το email “git config user.email 41898282+github-actions[bot]@users.noreply.github.com”, η εντολή ορίζει το email που θα συνοδεύει το commit. Το συγκεκριμένο email είναι η προεπιλεγμένη “ανώνυμη” διεύθυνση που παρέχει το GitHub για τα actions bots. Με την τρέχουσα διεύθυνση

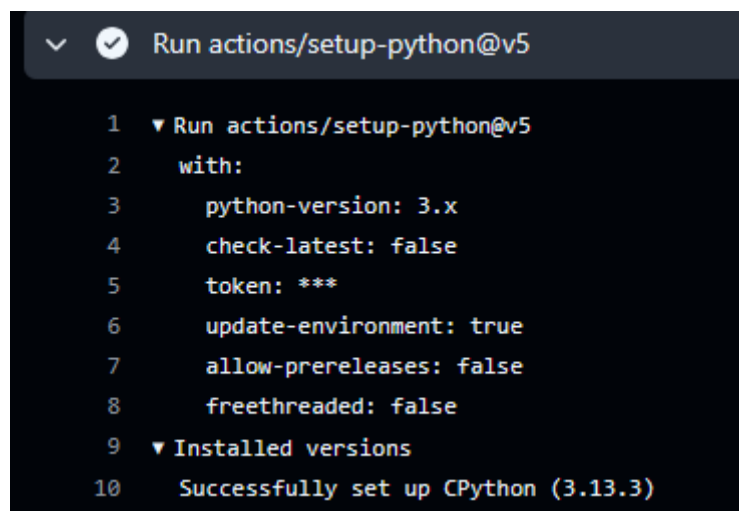
email εξυπηρετούμε δύο βασικούς σκοπούς. Πρώτον επιτρέπει στο commit να αναγνωριστεί από το GitHub ως έγκυρο και υπογεγραμμένο από trusted bot. Δεύτερον αποφεύγουμε την προβολή του πραγματικού email του χρήστη για λόγους ασφάλειας.



```
1 ▼ Run git config user.name github-actions[bot]
2   git config user.name github-actions[bot]
3   git config user.email 41898282+github-actions[bot]@users.noreply.github.com
4   shell: /usr/bin/bash -e {0}
```

Εικόνα 15. Τα credential που ορίστηκαν στο αρχείο.

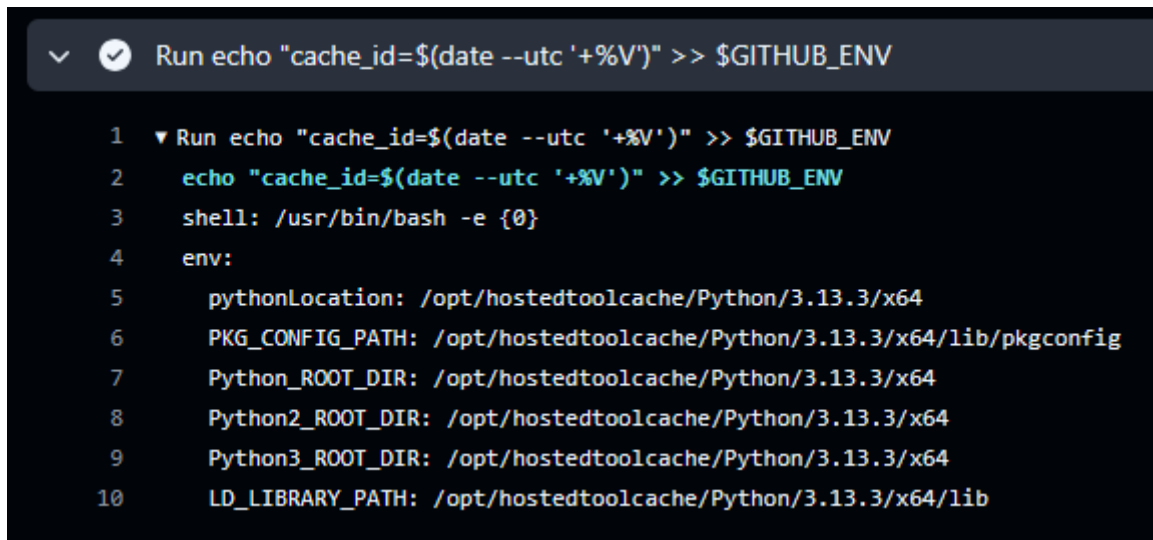
Βασική εντολή στο αρχείο ci.yml αφορά την εγκατάσταση της γλώσσας προγραμματισμού Python στο περιβάλλον εκτέλεσης του GitHub actions runner. Με την εντολή “- uses: actions/setup-python@v5” και τον ορισμό της έκδοσης Python με την εντολή “python-version: 3.x” ορίζουμε ότι στο συγκεκριμένο project θα υπάρχει εγκατεστημένη python 3.x και συνήθως επιλέγει την τελευταία έκδοση. Είναι απολύτως απαραίτητο γιατί το MkDocs και όλα τα plugins του είναι Python εφαρμογές, και συνεπώς απαιτούν το κατάλληλο περιβάλλον. Χωρίς αυτό το βήμα οποιαδήποτε μεταγενέστερη εγκατάσταση θα αποτύγχανε καθώς δεν θα έβρισκε το εργαλείο της Python pip (package manager) [20].



```
1 ▼ Run actions/setup-python@v5
2   with:
3     python-version: 3.x
4     check-latest: false
5     token: ***
6     update-environment: true
7     allow-prereleases: false
8     freethreaded: false
9   ▼ Installed versions
10  Successfully set up CPython (3.13.3)
```

Εικόνα 16. Η εγκατάσταση python στο runner.

Σε αυτό το βήμα χρησιμοποιείται η εντολή “echo” για να οριστεί μία μεταβλητή περιβάλλοντος με το όνομα cache_id όπως φαίνεται και από την εντολή μέσα στο αρχείο “- run: echo "cache_id=\$(date --utc '+%V')>> \$GITHUB_ENV”. Η τιμή της είναι ο αριθμός της τρέχουσας εβδομάδας του έτους, όπως υπολογίζεται με την εντολή “date --utc '+%V’”. Η μεταβλητή αποθηκεύεται στο ειδικό αρχείο \$GITHUB_ENV, το οποίο διαβάζεται από το GitHub actions ως ο κατάλογος των δηλωθέντων μεταβλητών και είναι διαθέσιμες στα επόμενα βήματα της αυτοματοποιημένης διαδικασίας. Αυτό επιτρέπει τη δημιουργία cache ανά εβδομάδα, επιτρέποντας τα builds να γίνονται πιο γρήγορα χρησιμοποιώντας ήδη εγκατεστημένες εξαρτήσεις, χωρίς όμως να παραμένουν στάσιμα λόγω παλιάς cache.



```
1 ▼ Run echo "cache_id=$(date --utc '+%V')" >> $GITHUB_ENV
2   echo "cache_id=$(date --utc '+%V')" >> $GITHUB_ENV
3   shell: /usr/bin/bash -e {0}
4   env:
5     pythonLocation: /opt/hostedtoolcache/Python/3.13.3/x64
6     PKG_CONFIG_PATH: /opt/hostedtoolcache/Python/3.13.3/x64/lib/pkgconfig
7     Python_ROOT_DIR: /opt/hostedtoolcache/Python/3.13.3/x64
8     Python2_ROOT_DIR: /opt/hostedtoolcache/Python/3.13.3/x64
9     Python3_ROOT_DIR: /opt/hostedtoolcache/Python/3.13.3/x64
10    LD_LIBRARY_PATH: /opt/hostedtoolcache/Python/3.13.3/x64/lib
```

Εικόνα 17. Η διαδικασία για την εβδομάδα του χρόνου.

Η αξιοποίηση του μηχανισμού cache που προσφέρει το Github actions επιτυγχάνεται με την εντολή “ - uses: actions/cache@v4”. Με την χρήση της Github action “actions/cache”, αποθηκεύεται το περιεχόμενο του φακέλου .cache το οποίο συνήθως περιέχει δεδομένα από προηγούμενες εγκαταστάσεις πακέτων ή plugins metadata, και άλλα βοηθητικά αρχεία. Η χρήση cache μειώνει σημαντικά τον χρόνο κατασκευής (build) των τεκμηριώσεων και περιορίζει τα αιτήματα προς τους package servers. Το cache key περιλαμβάνει το `cache\_id` της εβδομάδας, και υπάρχει η δυνατότητα rollback σε cache προηγούμενων εβδομάδων μέσω της παραμέτρου restore-keys, προσφέροντας ευελιξία.

```
Run actions/cache@v4

1  Run actions/cache@v4
2  with:
3    key: mkdocs-material-21
4    path: .cache
5    restore-keys: mkdocs-material-
6
7    enableCrossOsArchive: false
8    fail-on-cache-miss: false
9    lookup-only: false
10   save-always: false
11  env:
12    pythonLocation: /opt/hostedtoolcache/Python/3.13.3/x64
13    PKG_CONFIG_PATH: /opt/hostedtoolcache/Python/3.13.3/x64/lib/pkgconfig
14    Python_ROOT_DIR: /opt/hostedtoolcache/Python/3.13.3/x64
15    Python2_ROOT_DIR: /opt/hostedtoolcache/Python/3.13.3/x64
16    Python3_ROOT_DIR: /opt/hostedtoolcache/Python/3.13.3/x64
17    LD_LIBRARY_PATH: /opt/hostedtoolcache/Python/3.13.3/x64/lib
18    cache_id: 21
19  Cache not found for input keys: mkdocs-material-21, mkdocs-material-
```

Εικόνα 18. Ο μηχανισμός cache του Github.

Ακολουθούν τέσσερις διαδοχικές εντολές οι οποίες εγκαθιστούν μέσω pip τα απαραίτητα πακέτα Python. Συγκεκριμένα, γίνεται εγκατάσταση του mkdocs-material που αποτελεί το βασικό θέμα του site, και των plugins mkdocs-awesome-pages-plugin, markdown-grid-tables και mkdocs-glightbox, τα οποία προσθέτουν λειτουργίες όπως προσαρμοσμένη πλοήγηση, υποστήριξη πινάκων σε markdown, και εμφάνιση εικόνων με lightbox. Αυτά τα βήματα εξασφαλίζουν την κατάλληλη μορφοποίηση και λειτουργικότητα της τελικής τεκμηρίωσης.

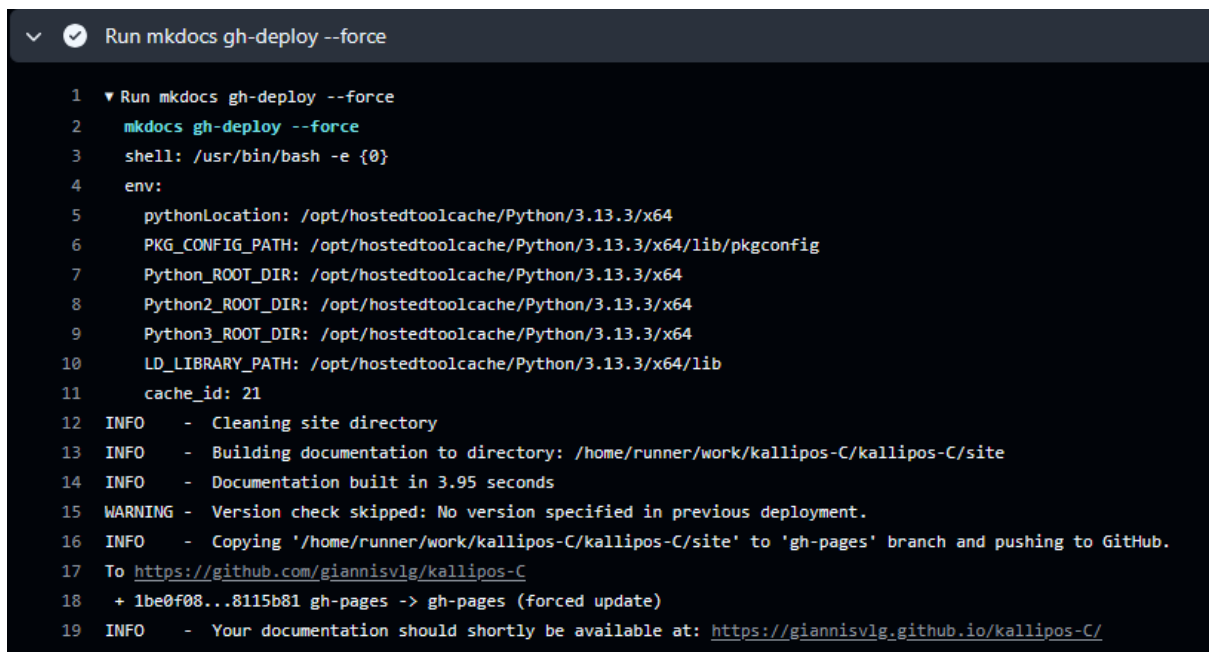
```
> Run pip install mkdocs-material
> Run pip install mkdocs-awesome-pages-plugin
> Run pip install markdown-grid-tables
> Run pip install mkdocs-glightbox
```

Εικόνα 19. Η εγκατάσταση των πακέτων python.

Στην εικόνα 15 βλέπουμε τα τέσσερα βασικά πακέτα python τα οποία απαραίτητα για την σωστή απεικόνιση του project με την μορφή webpage. Το πρώτο πακέτο είναι το βασικό το οποίο είναι το MkDocs μαζί με το θέμα material και τα ενσωματωμένα χαρακτηριστικά του.

Η εγκατάσταση συνεχίζει με το plugin που επιτρέπει προσαρμοσμένη ταξινόμηση και εμφάνιση της πλοήγησης στο sidebar με όνομα awesome-pages-plugin. Για την υποστήριξη εμφάνισης πλέγματος ή και πίνακα τύπου reStructuredText εντός Markdown χρειάζεται και η εγκατάσταση του markdown-grid-tables. Τελευταίο plugin είναι το glightbox το οποίο επιτρέπει την μεγέθυνση και προβολή μεμονωμένων φωτογραφιών στο site το οποίο έρχεται με διάφορα χαρακτηριστικά τόσο λειτουργικά όσο και εμφανισιακά.

Στο αρχείο ci.yml τελευταία εντολή είναι η “- run: mkdocs gh-deploy --force”. Η εντολή αυτή δημιουργεί το τελικό build του documentation, μετατρέποντας τα Markdown αρχεία σε HTML, και κάνει αυτόματο push του περιεχομένου στο branch gh-pages του repository. Με αυτόν τον τρόπο, γίνεται άμεσα διαθέσιμο στο διαδίκτυο. Εκτελώντας την εντολή “mkdocs gh-deploy”, η οποία κάνει τρεις βασικές διαδικασίες. Ξεκινάει κάνοντας build την τεκμηρίωση μετατρέποντας τα .md σε .html. Δημιουργεί ένα commit με το site στον φάκελο site/. Κάνει push στο branch gh-pages, το οποίο είναι το προεπιλεγμένο για το GitHub Pages hosting. Η επιλογή --force εξασφαλίζει ότι, ακόμη και αν υπάρχει ήδη κάποιο προηγούμενο deployment, θα αντικατασταθεί χωρίς επιβεβαίωση. Με αυτή την εντολή ολόκληρο το documentation γίνεται άμεσα διαθέσιμο online, μέσω GitHub Pages, χωρίς να απαιτείται χειροκίνητη παρέμβαση.

A screenshot of a terminal window with a dark background. At the top, there's a title bar that says "Run mkdocs gh-deploy --force" with a dropdown arrow and a checkmark icon. Below the title bar, the terminal shows the execution of the command. It starts with a prompt "1 ▼ Run mkdocs gh-deploy --force", followed by the command "2 mkdocs gh-deploy --force". Then it shows the shell environment: "3 shell: /usr/bin/bash -e {0}" and "4 env:". The environment variables listed are: "5 pythonLocation: /opt/hostedtoolcache/Python/3.13.3/x64", "6 PKG_CONFIG_PATH: /opt/hostedtoolcache/Python/3.13.3/x64/lib/pkgconfig", "7 Python_ROOT_DIR: /opt/hostedtoolcache/Python/3.13.3/x64", "8 Python2_ROOT_DIR: /opt/hostedtoolcache/Python/3.13.3/x64", "9 Python3_ROOT_DIR: /opt/hostedtoolcache/Python/3.13.3/x64", "10 LD_LIBRARY_PATH: /opt/hostedtoolcache/Python/3.13.3/x64/lib", and "11 cache_id: 21". The output of the command follows: "12 INFO - Cleaning site directory", "13 INFO - Building documentation to directory: /home/runner/work/kallipos-C/kallipos-C/site", "14 INFO - Documentation built in 3.95 seconds", "15 WARNING - Version check skipped: No version specified in previous deployment.", "16 INFO - Copying '/home/runner/work/kallipos-C/kallipos-C/site' to 'gh-pages' branch and pushing to GitHub.", "17 To https://github.com/giannisvlg/kallipos-C", "18 + 1be0f08...8115b81 gh-pages -> gh-pages (forced update)", and "19 INFO - Your documentation should shortly be available at: https://giannisvlg.github.io/kallipos-C/".

Εικόνα 20. Build και push στο gh-pages.

Ιδιαιτερότητες στην σύνταξη της Markdown

Κατά την διάρκεια της μετατροπής του βιβλίου σε MkDocs project υπήρξαν κάποιες ιδιαιτερότητες, καθώς στην εφαρμογή της η Markdown έχει κάποιους κανόνες που πρέπει να ληφθούν υπόψη. Τα χαρακτηριστικά της γνωρίσματα είναι τόσο η απλότητα όσο και η αναγνωσιμότητά της. Με τον στόχο αυτό υπάρχουν συντακτικές τεχνικές για την μορφοποίηση του κειμένου.

- 1) Καθώς κάθε αρχείο με κατάληξη .md που προστίθεται είναι και ένα διαφορετικό κεφάλαιο στην αρχή του εκάστοτε αρχείου δηλώνεται ο τίτλος του. Αυτό επιτυγχάνεται με τον χαρακτήρα “#” που έχει καθιερωθεί για τις επικεφαλίδες. Όπως και σε ένα απλό αρχείο word έτσι και εδώ υπάρχουν ιεραρχίες στις επικεφαλίδες, με την χρήση μονής δίδεσης (#) έχουμε την έμφαση της μεγάλης επικεφαλίδας. Κάθε επιπλέον εισαγωγή που γίνεται υποδηλώνει και μικρότερη ιεραρχία στην επικεφαλίδα. Παρακάτω βλέπουμε και ένα παράδειγμα του project.

```
1 # 1. Η πορεία της C μέχρι σήμερα
22 ## 1.4 Υλοποιήσεις της γλώσσας C
25
26 ### 1.4.1 GCC
27
```

Εικόνα 21. Ιεραρχία επικεφαλίδων με την χρήση της #.

Πως εμφανίζεται όμως στο site ;

Πίνακας περιεχομένων
1.1 Ιστορική αναδρομή στη γλώσσα C
1.2 Η φιλοσοφία της C
1.3 Προτυποποίηση της γλώσσας C
1.4 Υλοποιήσεις της γλώσσας C
1.4.1 GCC

Εικόνα 22. Ιεράρχηση κεφαλίδων στον ιστότοπο.

Φαίνεται ότι κεφάλαιο είναι το πρώτο υπάρχουν τα υποκεφάλαια ένα έως τέσσερα και με μία εισαγωγή της επικεφαλίδας λίγα εκατοστά πιο δεξιά εμφανίζεται ένα υποκεφάλαιο το οποίο ανήκει κάτω από το 1.4.

- 2) Οι λίστες είναι μία βασική επιλογή και συχνά γίνεται η χρήση αυτής. Υπάρχει ο βασικός διαχωρισμός αυτών σε διατεταγμένες και μη. Η διαφοροποίηση τους στην πράξη επιτυγχάνεται με τη χρήση ειδικών χαρακτήρων. Για μία διατεταγμένη λίστα χρησιμοποιούνται αριθμοί, ενώ οι εσοχές στο κείμενο υποδηλώνουν την ύπαρξη εμφωλευμένων λιστών. Στην άλλη περίπτωση των μη διατεταγμένων λιστών γίνεται με την χρήση των χαρακτήρων “-” και “*” για εμφωλευμένο κείμενο ή λίστα.


```

11 ## 16.2 Στατική ανάλυση κώδικα
12
13 Η στατική ανάλυση εξετάζει τον κώδικα χωρίς να τον εκτελεί. Ο κύριος σκοπός της είναι η ανίχνευση σφαλμάτων,
    βελτίωση της αξιοπιστίας και διευκόλυνση τη συντήρηση του κώδικα. Ειδικότερα, η στατική ανάλυση κώδικα
14
15 1. Ανίχνευση σφαλμάτων: Εντοπίζει συνηθισμένα προγραμματιστικά λάθη, όπως αποαναφορές (memory leaks) και άλλα.
16
17 2. Ποιότητα κώδικα: Επιβάλλει στάνταρντ κώδικα, οδηγίες στυλ και βέλτιστες πρακτικές. Εντοπίζει προβλήματα
    ονομάτων σε μεταβλητές και συναρτήσεις, και άλλα.
18
19 3. Βελτιστοποίηση απόδοσης: Εντοπίζει σημεία κώδικα που δημιουργούν προβλήματα ταχύτητας εκτέλεσης.
20
21
22 Μερικά από τα πλέον διαδεδομένα λογισμικά στατικής ανάλυσης κώδικα στη C είναι τα:
23
24 * Cppcheck: Το Cppcheck είναι ένα ευρέως χρησιμοποιούμενο εργαλείο στατικής ανάλυσης κώδικα για τη C και C++.
    Αναζητά σφάλματα που δεν χρησιμοποιούνται και άλλα. Το Cppcheck μπορεί να χρησιμοποιηθεί από τη γραμμή εντολών
    Cppcheck μέσω ενός γραφικού περιβάλλοντος.
25
26 * Clang-Tidy: Το Clang-Tidy είναι ένα εργαλείο στατικής ανάλυσης κώδικα που αποτελεί τμήμα της υποδομής
    Clang.
27
28 * Splint: Το Splint είναι ένα εργαλείο στατικής ανάλυσης, ανοικτού κώδικα, που έχει αναπτυχθεί συγκεκριμένα
    για την ανίχνευση σφαλμάτων χωρίς ελέγχους. Το Splint εμφανίζει αναλυτικές προειδοποιήσεις (warnings) για προβλήματα
    προγραμμάτων.
29
30 * Coverity: Το Coverity είναι ένα εμπορικό εργαλείο στατικής ανάλυσης που υποστηρίζει τη C και τη C++.
    Αναζητά σφάλματα που περιλαμβάνει ελέγχους ευπαθειών ασφάλειας, ελέγχους διαρροών πόρων και άλλους ελέγχους.
31
32 * PVS-Studio: Το PVS-Studio είναι εμπορικό λογισμικό στατικής ανάλυσης κώδικα για τη C και τη C++.
    Αναζητά αρχικοποιημένες μεταβλητές, μη-αποδοτικό κώδικα και άλλα. Εμφανίζει αναλυτικές αναφορές για τα προβλήματα.
33
34 Στη συνέχεια θα περιγραφούν ορισμένες βασικές δυνατότητες του Cppcheck. Ωστόσο, συχνά είναι αποτελεσματικότερη
    πληρέστερη ανάλυση και βελτίωση της ποιότητας του κώδικα.
35

```

Εικόνα 23. Παράδειγμα χρήσης λιστών.

- 3) Μέσα στα χαρακτηριστικά σημεία του MkDocs είναι και η χρήση πινάκων αλλά με περιορισμένη σύνταξη και δυσκολία στην ευθυγράμμιση. Ο πίνακας μπορεί να τοποθετηθεί οπουδήποτε μέσα στο κείμενο και μπορεί να περιέχει από κείμενο μέχρι κομμάτι κώδικα. Για την χρήση του απαιτείται ο ειδικός χαρακτήρας “|” το οποίο καθορίζει τον τίτλο της πρώτης στήλης και μετά για την εισαγωγή άλλης στήλης πρέπει να χρησιμοποιηθεί πάλι ο χαρακτήρας “|” και να υπάρχει και στο τέλος. Στην δεύτερη γραμμή εκδηλώνουμε στο MkDocs τη στοίχιση στηλών που θέλουμε. Από αριστερή στοίχιση με την χρήση της άνω κάτω τελείας αριστερά των παυλών, αντίστοιχα για την δεξιά αντιστοίχιση δεξιά από τις παύλες. Στο MkDocs για την σωστή ανταπόδοση του πίνακα θα πρέπει κάθε γραμμή της εκάστοτε στήλης να περιέχει τον ίδιο αριθμό χαρακτήρων. Για την επίτευξη αυτού στην δεύτερη γραμμή εισάγουμε όσες παύλες χρειαζόμαστε. Ακολουθεί παράδειγμα χρήσης από το project.

Πίνακας 1.1: *Επιλογή προτύπου της γλώσσας με τον διακόπτη μεταγλώττισης -std.*

Πρότυπο (standard)	Τιμή για τον διακόπτη μεταγλώττισης -std
:-----: :-----:	
`ISO C 1990`	c89 ή c90
`ISO C 1999`	c99
`ISO C 2011`	c11
`ISO C 2017`	c17 ή c18
`Working Draft for ISO C2x`	c2x

Εικόνα 24. Παράδειγμα χρήσης πίνακα.

- 4) Η ενσωμάτωση αυτούσιου κώδικα HTML μέσα στο κείμενο είναι ένα μοναδικό χαρακτηριστικό της Markdown. Αυτό επιτρέπει στους χρήστες να ξεπεράσουν κάποιους περιορισμούς της, αλλά ταυτόχρονα δημιουργεί εξάρτηση από τις γνώσεις HTML και ενδέχεται να επηρεάσει τη φορητότητα ή τη συμβατότητα του αρχείου. Στο project υπάρχει αυτή η ενσωμάτωση ειδικότερα σε σημεία που η χρήση της είναι απαραίτητη όπως:

α) για την προσαρμογή τριών εικόνων στην ίδια γραμμή

```
326 <div class="flex">
327   <div>
328     
329     <b> Σχήμα 6.1: </b><i>Περιεχόμενα μνήμης μετά την εκτέλεση της γραμμής 15.</i>
330   </div>
331   <div>
332     
333     <b> Σχήμα 6.2: </b><i>Περιεχόμενα μνήμης μετά την εκτέλεση της γραμμής 16.</i>
334   </div>
335   <div>
336     
337     <b> Σχήμα 6.3: </b><i>Περιεχόμενα μνήμης μετά την εκτέλεση της γραμμής 17.</i>
338   </div>
339 </div>
340
```

Εικόνα 25. Κώδικας HTML για την προσαρμογή και ευθυγράμμιση εικόνων.

β) για την εμφάνιση ειδικού χαρακτήρα διότι η χρήση του επηρέαζε την συμπεριφορά άλλου κώδικα στο οποίο βρισκόταν.

Χρήση του < αντί του < και αντίστοιχα του > για το >

```
120 5. <span class="p-style">break &lt;file:line#&gt;</span> - Ορισμός ενός breakpoint στη γραμμή #.
121 6. <span class="p-style">break &lt;file:line#&gt; if &lt;expression&gt;</span> - Ορισμός ενός breakpoint υπό συνθήκη στη γραμμή #.
```

Εικόνα 26. Αντικατάσταση HTML character.

Η Markdown διακρίνεται για την ευχρηστία και την απλότητά της, ωστόσο η σύνταξή της έχει συγκεκριμένες ιδιαιτερότητες που μπορεί να επηρεάσουν την ορθή παρουσίαση του περιεχομένου. Αποτελεί ιδανική επιλογή για έργα με διάφορους βαθμούς πολυπλοκότητας. Για προχωρημένες λειτουργίες, συχνά απαιτείται η υποστήριξη επεκτάσεων ή η χρήση HTML.

Κεφάλαιο 4 - Τεχνολογίες σύνταξης τεχνικών κειμένων και σχεδίασης σχημάτων

Η τεκμηρίωση αποτελεί κρίσιμο συστατικό σε κάθε έργο λογισμικού, καθώς διευκολύνει τη μετάδοση γνώσης, την κατανόηση συστημάτων και την ορθή χρήση των εφαρμογών. Για τον σκοπό αυτό, έχει αναπτυχθεί μια πληθώρα εργαλείων και γλωσσών που εξυπηρετούν τη συγγραφή τεχνικών κειμένων αλλά και την ενσωμάτωση διαγραμμάτων. Στο παρόν κεφάλαιο παρουσιάζονται τέσσερις ευρέως χρησιμοποιούμενες τεχνολογίες: Markdown, Mermaid, Excalidraw και reStructuredText (reST), μαζί με σύντομα παραδείγματα χρήσης τους.

Markdown

Η γλώσσα Markdown είναι μια εύκολη γλώσσα η οποία επιτρέπει τη σύνταξη κειμένων. Ύστερα μπορούν εύκολα να μετατραπούν σε κώδικα HTML και να προβληθούν σαν ιστοσελίδες ή documentation pages. Έχει ευρεία εφαρμογή σε αποθετήρια όπως το GitHub, σε static site generators όπως το MkDocs και σε editors όπως το Visual Studio Code και Jupyter [11]. Η βασική της φιλοσοφία είναι η ευχρηστία και η αναγνωσιμότητα. Με ελάχιστους κανόνες σύνταξης, επιτρέπει στον χρήστη να διαμορφώσει επικεφαλίδες, λίστες, πίνακες, έντονο υπογραμμισμένο και πλάγιο κείμενο, καθώς και να ενσωματώσει εικόνες ή και συνδέσμους. Η εκμάθησή της είναι εξαιρετικά εύκολη, ενώ η εφαρμογή της προσφέρει σημαντικά πλεονεκτήματα στην παραγωγή τεκμηρίωσης.

Η δημιουργία ενός Markdown αρχείο είναι αρκετά απλή σαν διαδικασία καθώς πέρα από μερικούς κανόνες σύνταξης δεν απαιτεί εξειδικευμένες γνώσεις προγραμματισμού. Η χρήση ενός απλού επεξεργαστή κειμένου είναι αρκετή για να δημιουργηθεί ένα τέτοιο αρχείο. Με την χρήση ενός τέτοιου προγράμματος επεξεργασίας κειμένου όπως το Visual Studio Code, το Notepad++ ή ακόμη και το σημειωματάριο των Windows μπορεί να δημιουργηθεί ένα αρχείο με κατάληξη “.md”. Με την κατάληξη αυτή δηλώνουμε ότι το αρχείο είναι τύπου Markdown. Αφού δημιουργηθεί το αρχείο είναι πλέον εφικτή η εισαγωγή περιεχομένου σύμφωνα με τους κανόνες σύνταξης του Markdown. Σε κάθε αρχείο οι τίτλοι καθορίζονται με τον ειδικό χαρακτήρα “#”. Με την προσθήκη επιπλέον χαρακτήρων δίεσης κατεβαίνει και το επίπεδο της κεφαλίδας καθώς με μόνο ένα χαρακτήρα θέτουμε το βασικό κεφάλαιο και με περισσότερα συνήθως δηλώνονται υποκεφάλαια. Μία ακόμη χρήση στο Markdown είναι και αυτή για την επισήμανση λέξεων ή ακόμη και φράσεων με την χρήση αστερίσκων ή κάτω παύλας. Για έντονη γραφή, περιβάλλουμε το κείμενο με διπλά αστεράκια “**”, ενώ για πλάγια γραφή χρησιμοποιούμε ένα αστεράκι “*” [3]. Αντίστοιχα η δημιουργία λιστών ξεκινώντας από τις αταξινομήτες λίστες που δημιουργούνται με τον χαρακτήρα “-” και για εμφολευμένη χρησιμοποιώντας “*” και οι ταξινομημένες δημιουργούνται ξεκινώντας κάθε στοιχείο με αριθμό και τελεία, όπως 1. ,2. κ.λπ .

Βασικό στοιχείο που μπορεί να ενταχθεί σε ένα Markdown αρχείο είναι εικόνες και σύνδεσμοι. Καθώς και τα δύο έχουν παρόμοια σύνταξη φαίνεται ότι οι εικόνες ξεκινούν με θαυμαστικό και ύστερα μέσα σε αγκύλες το όνομα της εικόνας και ακολουθείται ακριβώς μετά με την διεύθυνση ή την διαδρομή (path) του αρχείου. Στους συνδέσμους παρακολουθούμε το ίδιο μοτίβο με μία διαφορά, την αφαίρεση του αρχικού θαυμαστικού.

```

326 ![Εικόνα example.png](src/images/example.png)
327 [Το site που φιλοξενείται η εικόνα.](www.example.com)
328

```

Εικόνα 27. Διαφοροποίηση μεταξύ εικόνας και συνδέσμου.

Οι τρεις παύλες “---” σε ένα αρχείο Markdown υποδηλώνουν ότι υπάρχει γραμμή διαχωρισμού του κειμένου. Ένα βασικό χαρακτηριστικό σε τεχνολογίες documentation είναι και η υποστήριξη για κομμάτια κώδικα. Στο Markdown αυτό έχει επιτευχθεί είτε ενσωματωμένα σε γραμμή με τη χρήση του χαρακτήρα “`”, είτε σε ξεχωριστό μπλοκ με την χρήση τριπλού backtick “```” [8]. Παρακάτω βλέπουμε δύο παραδείγματα και τα δύο με την χρήση του τριπλού backtick. Στο πρώτο γίνεται εισαγωγή κώδικα c που δηλώνεται στην αρχή (.c) και το οποίο βρίσκεται σε άλλο αρχείο (ch6_p17.c). Αλλά και στο ίδιο αρχείο βλέπουμε ότι υπάρχει η δυνατότητα να εμφανίζεται ένα περιεχόμενο με την μορφή code block.

```

290 ```{.c title="Κώδικας 6.17: ch6_p17.c - παράδειγμα βελτιωμένης χρήσης ένωσης." linenums="1"}
291 --8<-- "src/ch6_p17.c"
292 ```
293
294 Μια πιθανή είσοδος και έξοδος του προγράμματος είναι η ακόλουθη:
295 ```
296 Input an int: 42
297 Integer: 42
298 Input a double: 3.14
299 Double: 3.140000
300 ```

```

Εικόνα 28. Παράδειγμα για την χρήση code block.

Σημαντικό επίσης είναι και η υποστήριξη για την δημιουργία πινάκων με την κατακόρυφη γραμμή “|” για τον διαχωρισμό των στηλών.

Μετά την αποθήκευση του αρχείου το Markdown μπορεί να προβληθεί με τον μορφοποιημένο τρόπο μέσω εξειδικευμένων προγραμμάτων προβολής ή μέσα από πλατφόρμες όπως το GitHub, οι οποίες ερμηνεύουν και αποδίδουν σωστά τη σύνταξη.

Εν κατακλείδι τα βήματα για την δημιουργία ενός αρχείου Markdown συνοψίζονται στα εξής:

- 1) άνοιγμα ενός επεξεργαστή κειμένου
- 2) κατάλληλη κατάληξη του αρχείου (.md)
- 3) σωστή σύνταξη σύμφωνα με τους κανόνες του Markdown.

Mermaid

Η Mermaid είναι μια βιβλιοθήκη της JavaScript που επιτρέπει τη δημιουργία διαγραμμάτων μέσα από κείμενο, χρησιμοποιώντας σύνταξη που θυμίζει Markdown. Μέσω αυτής της τεχνολογίας είναι δυνατή η απεικόνιση διαγραμμάτων ροής, Gantt charts, sequence diagrams και άλλων τύπων, χωρίς την ανάγκη χρήσης εξωτερικών σχεδιαστικών εργαλείων [17].

Χρησιμοποιείται κυρίως μέσα σε Markdown αρχεία και υποστηρίζεται από πολλές σύγχρονες πλατφόρμες, όπως το GitHub, το Obsidian και το Notion. Η ενσωμάτωσή της σε ένα έργο τεκμηρίωσης που βασίζεται στο MkDocs γίνεται μέσω ειδικών επεκτάσεων (extensions) και με την προσθήκη της βιβλιοθήκης στον φάκελο `extra_javascript`. Αφού ενεργοποιηθεί, ο χρήστης μπορεί να εισάγει διαγράμματα με τρόπο απλό και ευανάγνωστο, απευθείας μέσα στο κείμενο.

Για να δημιουργηθεί μια οπτικοποίηση με το Mermaid, χρειάζεται ένα περιβάλλον το οποίο υποστηρίζει την ενσωμάτωσή του. Αυτό μπορεί να είναι από ένα Markdown αρχείο που τροποποιείται σε έναν επεξεργαστή κειμένου όπως το Obsidian μέχρι έναν online επεξεργαστή όπως το mermaid.live.

Ένα από τα σημαντικά χαρακτηριστικά είναι η σύνταξη που ξεκινάει με τον προσδιορισμό του τύπου του διαγράμματος. Ο προσδιορισμός `graph TD` σε ένα διάγραμμα ροής υποδηλώνει την κατεύθυνση όπου TD είναι από επάνω προς τα κάτω (Top Down) [6]. Μετά την αρχική δήλωση για το είδος του διαγράμματος ακολουθούν οι δηλώσεις για την ποσότητα των κόμβων αλλά και για τις μεταξύ τους σχέσεις.

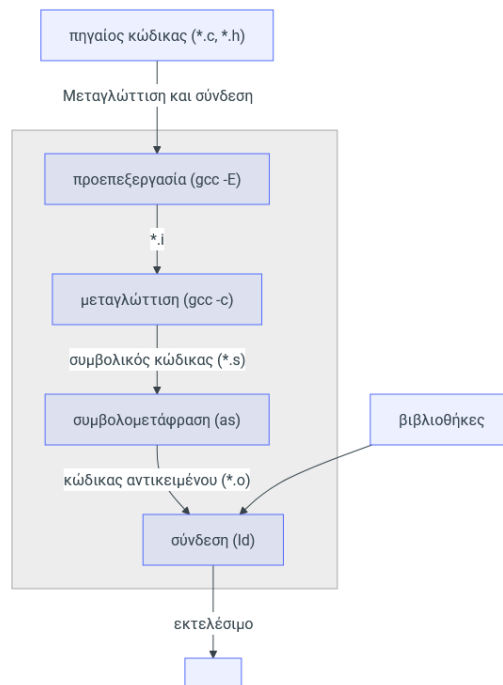
Η δημιουργία του διαγράμματος βασίζεται στην καταγραφή των στοιχείων και των σχέσεων τους με συγκεκριμένη σύνταξη. Δεν απαιτεί καμία σχεδιαστική ικανότητα από τον χρήστη αρκεί η επαρκής κατανόηση του διαγράμματος και το γράψιμο του αντίστοιχου κώδικα. Μετά την ολοκλήρωση του κώδικα αυτόματα γίνεται οπτικοποίηση του διαγράμματος σύμφωνα με τον κώδικά που βασίστηκε. Η αποθήκευση του κειμένου μπορεί να γίνει μέσα σε αρχείο Markdown με την κατάλληλη σήμανση στο code block ````mermaid`. Το Mermaid έτσι αποδίδει ένα διάγραμμα με τόσο απλό τρόπο χωρίς την χρήση εργαλείων γραφικής διεπαφής.

Ακολουθεί ένα παράδειγμα από το project για την χρήση Mermaid.

```
```mermaid
graph TD
 A["πηγαίος κώδικας (*.c, *.h)"] -->|Μεταγλώττιση και σύνδεση| B
 subgraph " "
 B["προεπεξεργασία (gcc -E)"]
 B -->|".i"| C["μεταγλώττιση (gcc -c)"]
 C -->|"συμβολικός κώδικας (*.s)"| D["συμβολομετάφραση (as)"]
 D -->|"κώδικας αντικειμένου (*.o)"| E["σύνδεση (ld)"]
 end
 E -->|"εκτελέσιμο"| F[" "]
 G["βιβλιοθήκες"] --> E
```
```

Εικόνα 29. Κώδικας Mermaid στο project.

Και η οπτικοποίηση του κώδικα Mermaid στο site.



Σχήμα 2.1: Στάδια μεταγλώττισης και σύνδεσης.

Εικόνα 30. Η οπτικοποίηση του κώδικα Mermaid.

Excalidraw

Το Excalidraw αποτελεί ένα διαδικτυακό εργαλείο σχεδίασης διαγραμμάτων το οποίο προσφέρει τη δυνατότητα δημιουργίας σχημάτων με ελεύθερη μορφή, δίνοντας την αίσθηση χειρόγραφου σκίτσου. Χαρακτηρίζεται από απλότητα στη χρήση, υποστήριξη συνεργατικής σχεδίασης σε πραγματικό χρόνο και δυνατότητες εξαγωγής του τελικού διαγράμματος σε διάφορες μορφές, όπως SVG και PNG.

Η χρήση του είναι ιδανική σε περιπτώσεις όπου απαιτείται η δημιουργία αρχιτεκτονικών σχεδιαγραμμάτων, mockups, brainstorming σχεδίων και εν γένει ελεύθερων σχηματικών απεικονίσεων. Η ενσωμάτωση διαγραμμάτων Excalidraw σε documentation γίνεται μέσω της εισαγωγής των εικόνων που έχουν παραχθεί από την πλατφόρμα στο κατάλληλο σημείο του κειμένου.

Ένα διάγραμμα για να δημιουργηθεί με το Excalidraw ξεκινά με την πρόσβαση στην ιστοσελίδα www.excalidraw.com [5]. Εκεί, ο χρήστης αποκτά πρόσβαση σε έναν καμβά σχεδίασης, στον οποίο μπορεί να τοποθετήσει χειροκίνητα διάφορα σχήματα και συνδέσεις και πλαίσια κειμένου μεταξύ αυτών. Το περιβάλλον εργασίας είναι ιδιαίτερα φιλικό και προσφέρει επιλογές για χρώματα, πάχος γραμμής και τοποθέτηση στοιχείων καθιστώντας το έτσι αρκετά απλό στην χρήση του.

Η σχεδίαση γίνεται με εναπόθεση των αντικειμένων επάνω στον καμβά, και τα αντικείμενα μπορούν να προσαρμοστούν εύκολα μέσω ποντικιού ή γραφίδας. Επιπλέον, το Excalidraw προσφέρει δυνατότητες συνεργασίας σε πραγματικό χρόνο, όπου πολλαπλοί χρήστες

μπορούν να επεξεργάζονται το ίδιο διάγραμμα ταυτόχρονα, καθιστώντας το ιδανικό για ομαδική εργασία ή εξ αποστάσεως μαθήματα.

Μετά την ολοκλήρωση το διάγραμμα αποθηκεύεται σε μορφή “.excalidraw” για επεξεργασία ή στην περίπτωση εξαγωγής σε μορφή “.md” για εισαγωγή σε Markdown διαφορετικά σε “.png” ή “.svg”. Υποστηρίζεται επίσης η ενσωμάτωσή του σε πλατφόρμες όπως το Obsidian μέσω σχετικών plugins.

Σε αντίθεση με το Mermaid το Excalidraw δεν έχει παραμετροποίηση μέσα από κώδικα αλλά στηρίζεται εξ ολοκλήρου στο σχεδιαστικό κομμάτι. Αυτό γίνεται κυρίως για καταγραφή και αποτύπωση ιδεών ή αποτύπωση εννοιών σε μορφή γραφικής παρουσίασης.

reStructuredText (reST)

Η γλώσσα reStructuredText, γνωστή και ως reST, αποτελεί μια ισχυρή και πιο αυστηρή σε σύνταξη εναλλακτική του Markdown. Αναπτύχθηκε στο πλαίσιο του έργου Docutils και χρησιμοποιείται ευρέως από το εργαλείο Sphinx [9], το οποίο είναι το βασικό σύστημα τεκμηρίωσης για τη γλώσσα Python.

Με το reST μπορούμε να δημιουργήσουμε περίπλοκα έγγραφα με δομημένη μορφή, υποστηρίζοντας έννοιες όπως αριθμημένες παραπομπές, αυτόματη δημιουργία πίνακα περιεχομένων, ενσωμάτωση μαθηματικών παραστάσεων, παραπομπές σε αρχεία και άλλα έγγραφα, καθώς και πρότυπα τεκμηρίωσης για συναρτήσεις, κλάσεις και modules.

Λόγω της αυξημένης πολυπλοκότητας, το reST είναι πιο δύσκολο στην εκμάθηση σε σχέση με το Markdown, ωστόσο είναι εξαιρετικά ισχυρό όταν πρόκειται για την τεκμηρίωση μεγάλων έργων που απαιτούν ακρίβεια, συνέπεια και υψηλή τυποποίηση.

Η δημιουργία ενός reST αρχείου ξεκινά με την χρήση ενός επεξεργαστή κειμένου και την αποθήκευσή του με την σωστή κατάληξη “.rst”. Όπως και στο Markdown η σύνταξη είναι αυστηρή καθώς βασίζεται στην ευθυγράμμιση του κειμένου και την χρήση συγκεκριμένων χαρακτήρων, όπως παύλες, αστερίσκοι και τελείες για τη δημιουργία επικεφαλίδων, λιστών και υποσημειώσεων.

Στην reST οι λίστες δημιουργούνται με αστερίσκους “*”, ενώ οι ταξινομημένες λίστες με αριθμούς και τελείες. Για έντονη και πλάγια γραφή χρησιμοποιούνται διπλά και μονά backticks “`” ή υπογράμμιση. Επίσης, το reST επιτρέπει τη δημιουργία πινάκων, σημειώσεων, συνδέσμων, ακόμα και την ενσωμάτωση κώδικα.

Το reStructuredText (reST) είναι ιδιαίτερα χρήσιμο όταν χρειάζεται να γράψουμε documentation για λογισμικό, ειδικά σε έργα που βασίζονται στη γλώσσα python. Χάρη στην πολύ καλή συνεργασία του με εργαλεία όπως το Sphinx, μπορούμε να μετατρέψουμε αυτόματα τα αρχεία reST σε καλοσχεδιασμένες σελίδες HTML, χωρίς την ανάγκη να γράψουμε κάποιο κώδικα για το στήσιμο του περιβάλλοντος.

Η σύνταξή του, αν και λίγο πιο απαιτητική από του Markdown, είναι καθαρή και δομημένη, κάτι που βοηθά ιδιαίτερα όταν έχουμε να διαχειριστούμε μεγάλα ή σύνθετα κείμενα. Δεν είναι τυχαίο ότι χρησιμοποιείται ευρέως σε έργα ανοιχτού κώδικα, όπου η ποιότητα και η αυτοματοποίηση της τεκμηρίωσης παίζουν σημαντικό ρόλο.

Πίνακας 2. Συγκεντρωτική παρουσίαση των τεσσάρων τεχνολογιών τεκμηρίωσης.

| Εργαλείο | Είδος | Τρόπος
χρήσης | Μορφή
εισόδου | Έξοδος |
|-------------------------|---------------------------------|--------------------------|--------------------------|-----------------------------|
| <i>Markdown</i> | Γλώσσα markup | Χρήση αρχείων .md | Απλό κείμενο | HTML, μορφοποιημένο κείμενο |
| <i>reStructuredText</i> | Γλώσσα markup για τεκμηρίωση | Χρήση αρχείων .rst | Δομημένο απλό κείμενο | HTML, PDF (Sphinx) |
| <i>Mermaid</i> | Εργαλείο διαγραμμάτων με κώδικα | Χρήση μέσα από Markdown | Κώδικας διαγράμματος | Αυτόματα διαγράμματα |
| <i>Excalidraw</i> | Διαδραστικό εργαλείο σχεδίασης | Σχεδίαση μέσα στο site | Γραφική σχεδίαση | Εικόνα ή αρχείο .excalidraw |

Επίλογος

Η τεκμηρίωση λογισμικού αποτελεί θεμέλιο λίθο στην ανάπτυξη, χρήση και συντήρηση πληροφοριακών συστημάτων, ιδιαίτερα στον τομέα της εκπαιδευτικής τεχνολογίας. Η παρούσα εργασία εξερεύνησε τη χρήση του MkDocs και του θέματος MkDocs Material ως βασικά εργαλεία για τη δημιουργία δομημένου, φιλικού και διαδραστικού εκπαιδευτικού περιεχομένου, καθώς και τη σημασία των σύγχρονων τεχνικών συγγραφής τεκμηρίωσης.

Αρχικά, αναλύθηκε η γενικότερη έννοια της τεκμηρίωσης, η οποία δεν περιορίζεται μόνο σε οδηγίες χρήσης αλλά καλύπτει ευρύτερες πτυχές, όπως η εγκατάσταση, η αρχιτεκτονική του λογισμικού, οι τεχνικές λεπτομέρειες και η ευκολία που προσφέρει σε ομάδες.

Στη συνέχεια, παρουσιάστηκε το MkDocs ως ένα εργαλείο βασισμένο στη γλώσσα Markdown, το οποίο επιτρέπει την εύκολη σύνθεση τεχνικών και εκπαιδευτικών εγχειριδίων. Το θέμα MkDocs Material ενισχύει την αισθητική και λειτουργική εμπειρία, προσφέροντας χαρακτηριστικά όπως dark mode, έξυπνη πλοήγηση, υποστήριξη για plugins και πολυγλωσσικό περιβάλλον. Οι δυνατότητες του εργαλείου υποστηρίχθηκαν από αναλυτική περιγραφή πραγματικού παραδείγματος, μέσω της υλοποίησης μιας εκπαιδευτικής εφαρμογής.

Η μετατροπή του εκπαιδευτικού βιβλίου «Μια σύγχρονη προσέγγιση στη γλώσσα C»⁷ σε στατικό site βοήθησε στην κατανόηση των τεχνολογιών αλλά και εξυπηρέτησε στην δημιουργία μιας διαδραστικής σελίδας για την καλύτερη κατανόηση των εννοιών του βιβλίου. Υπήρξε εκτενής καταγραφή των διαδικασιών που ακολουθήθηκαν για την εγκατάσταση των κατάλληλων προγραμμάτων αλλά και την δημιουργία του έργου.

Η εργασία παρουσίασε επίσης τις δυνατότητες αυτόματης ανάπτυξης και ενημέρωσης της τεκμηρίωσης μέσω της πλατφόρμας GitHub και του αρχείου ci.yml, που αποτελεί κρίσιμο στοιχείο αυτοματοποίησης. Με αυτόν τον τρόπο, εξασφαλίζεται ότι κάθε αλλαγή στον πηγαίο κώδικα ενημερώνει αυτόματα και την τεκμηρίωση, διατηρώντας την επίκαιρη και έγκυρη. Ειδική αναφορά έγινε στην ερμηνεία κάθε βήματος του αρχείου CI, ενισχύοντας την κατανόηση των διαδικασιών συνεχούς ενσωμάτωσης (CI).

Τέλος, έγινε μία σύντομη συγκριτική προσέγγιση άλλων εργαλείων τεκμηρίωσης όπως τα GitBook, Jekyll, Sphinx, ενώ εξετάστηκαν και τεχνολογίες σχεδίασης και συγγραφής τεχνικών εγγράφων όπως το Mermaid, το Excalidraw και το reStructuredText.

Συνοψίζοντας, το MkDocs Material αναδεικνύεται σε μία ιδιαίτερα ισχυρή λύση για την τεκμηρίωση λογισμικού σε περιβάλλοντα εκπαίδευσης και ανάπτυξης. Η απλότητα στη χρήση, η δυνατότητα παραμετροποίησης και η ενσωμάτωσή του σε μοντέρνες πλατφόρμες ανάπτυξης καθιστούν το εργαλείο απαραίτητο για σύγχρονες ομάδες λογισμικού. Η τεκμηρίωση καθίσταται έτσι όχι απλώς μια υποστηρικτική ενέργεια, αλλά θεμελιώδης διαδικασία για την ποιότητα και τη βιωσιμότητα των έργων λογισμικού.

⁷ <https://repository.kallipos.gr/handle/11419/11683>

Βιβλιογραφία

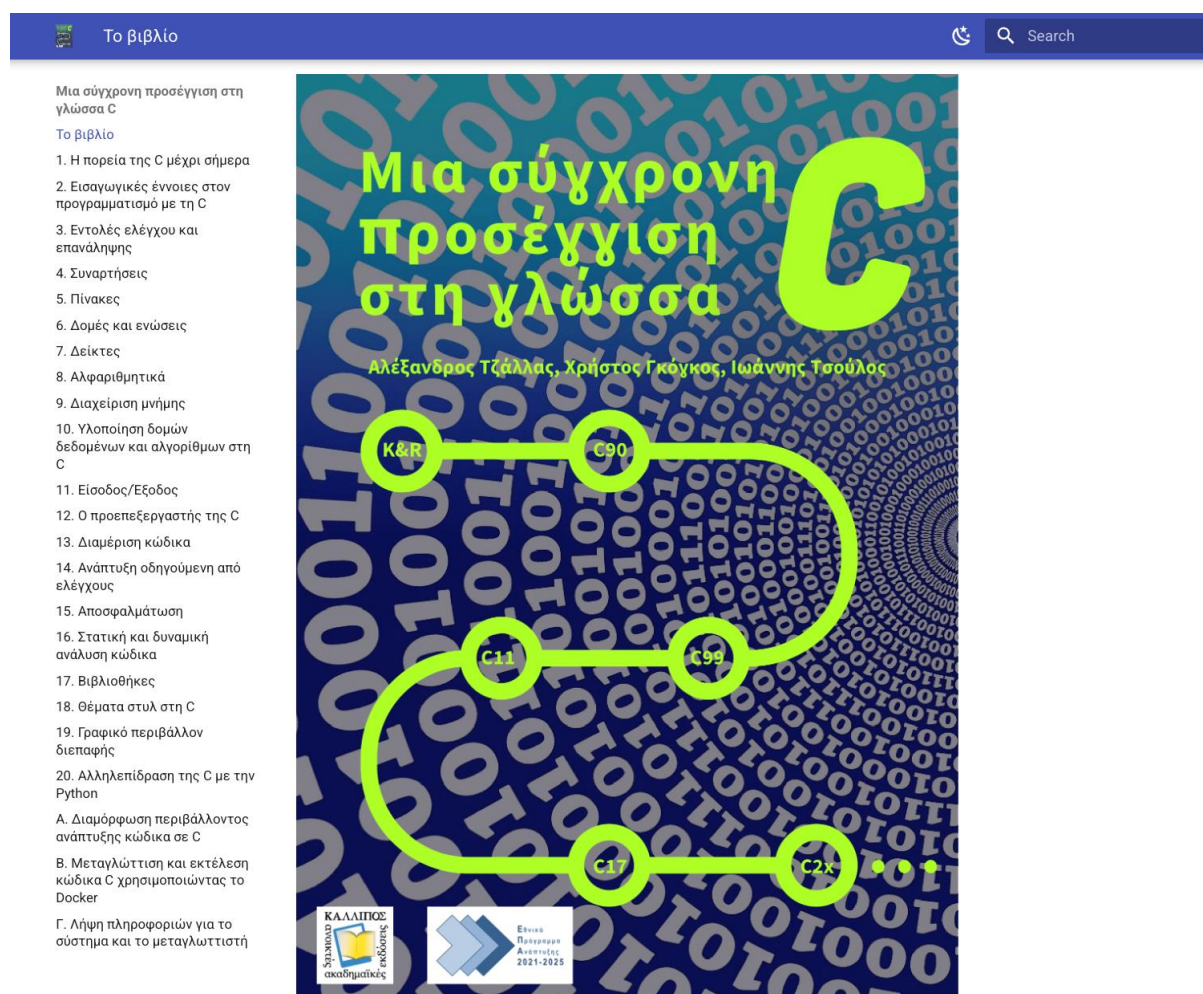
1. MkDocs Official Documentation. MkDocs User Guide. Διαθέσιμο από: <https://www.mkdocs.org/user-guide/>
2. Squidfunk. Material for MkDocs Documentation. Διαθέσιμο από: <https://squidfunk.github.io/mkdocs-material/>
3. Gruber, J. (2004). Markdown: Syntax. Διαθέσιμο από: <https://daringfireball.net/projects/markdown/syntax>
4. Python Software Foundation. reStructuredText Primer. Διαθέσιμο από: <https://docutils.sourceforge.io/docs/user/rst/quickstart.html>
5. Excalidraw. Excalidraw – Virtual whiteboard for sketching diagrams. Διαθέσιμο από: <https://excalidraw.com>
6. Mermaid.js. Mermaid Live Editor and Syntax Guide. Διαθέσιμο από: <https://mermaid.js.org>
7. Chacon, S., & Straub, B. (2014). Pro Git. Apress. Διαθέσιμο από: <https://git-scm.com/book/en/v2>
8. Zettlr GmbH. (2022). Writing with Markdown: A Comprehensive Guide. Διαθέσιμο από: <https://www.zettlr.com/learn/markdown>
9. The Sphinx Documentation Project. Sphinx reStructuredText Markup. Διαθέσιμο από: <https://www.sphinx-doc.org/en/master/usage/restructuredtext/>
10. GitHub Docs. Working with Markdown on GitHub. Διαθέσιμο από: <https://docs.github.com/en/get-started/writing-on-github>
11. Markdown Guide. The Comprehensive Markdown Guide. Διαθέσιμο από: <https://www.markdownguide.org>
12. Stack Overflow. MkDocs Questions and Discussions. Διαθέσιμο από: <https://stackoverflow.com/questions/tagged/mkdocs>
13. Read the Docs. Hosting MkDocs Projects. Διαθέσιμο από: <https://docs.readthedocs.io/en/stable/intro/import-guide.html#mkdocs>
14. Real Python. (2022). Documentation Best Practices with Markdown and MkDocs. Διαθέσιμο από: <https://realpython.com/documenting-python-code/>
15. Dev.to. How to Build Beautiful Docs with MkDocs Material. Διαθέσιμο από: <https://dev.to/jeffreycoder/how-to-build-beautiful-docs-with-mkdocs-material-1f5j>

16. YouTube – Step-by-Step Guide To Making An MkDocs Material Website. (2023). MkDocs Tutorial for Beginners. Διαθέσιμο από: <https://www.youtube.com/watch?v=NY7DHvo1XVM>
17. Mermaid.js Project. Mermaid Live Editor – Interactive Diagram Editor for Mermaid Syntax. Διαθέσιμο από: <https://mermaid.live>
18. Spinellis, D. (2012). GitHub as a collaborative writing platform. IEEE Software, 29(5), 63–67.
19. GitHub Docs. (2024). Official Documentation. Διαθέσιμο από: <https://docs.github.com/>
20. Preston-Werner, T. (2010). Readme Driven Development. Διαθέσιμο από: <https://tom.preston-werner.com/2010/08/23/readme-driven-development.html>

Παράρτημα

Screenshots από τη μεταφορά του βιβλίου «Μια σύγχρονη προσέγγιση στη γλώσσα C»⁸ του ΚΑΛΛΙΠΟΥ στο MkDocs στη σελίδα <https://chgogos.github.io/kallipos-C/>

Ακολουθούν μερικά αντιπροσωπευτικά screenshots από τις σελίδες του βιβλίου. Για παράδειγμα στην Εικόνα 31 παρουσιάζεται η αρχική σελίδα του βιβλίου ως ιστοσελίδα. Στην Εικόνα 32 φαίνεται ένα παράδειγμα με κώδικα που είναι αρχικά κρυμμένος και όταν ο χρήστης πατήσει τον τίτλο του κώδικα τότε εμφανίζεται και μπορεί όπως και οι άλλο κώδικες να αντιγραφεί με κλικ στο ειδικό εικονίδιο πάνω δεξιά. Τέλος, στην Εικόνα 33 εμφανίζεται ένα παράδειγμα της ισχυρής λειτουργίας του MkDocs για γρήγορη αναζήτηση οποιοδήποτε κειμένου σε όλες τις σελίδες του βιβλίου. Η λειτουργία της αναζήτησης είναι ενσωματωμένη στο MkDocs και υποστηρίζεται από το lunr.js⁹, μια ελαφριά μηχανή αναζήτησης πλήρους κειμένου για χρήση σε φυλλομετρητή, που καθιστά την ανάγκη χρήσης εξωτερικών υπηρεσιών περιττή. Στη συγκεκριμένη περίπτωση εμφανίζεται το αποτέλεσμα για την αναζήτηση του όρου `memcpy`.



Εικόνα 31. Αρχική σελίδα του βιβλίου

⁸ <https://repository.kallipos.gr/handle/11419/11683>

⁹ <https://lunrjs.com/>

6. Δομές και ενώσεις

Μια σύγχρονη προσέγγιση στη γλώσσα C

Το βιβλίο

1. Η πορεία της C μέχρι σήμερα

2. Εισαγωγικές έννοιες στον προγραμματισμό με τη C

3. Εντολές ελέγχου και επανάληψης

4. Συναρτήσεις

5. Πίνακες

6. Δομές και ενώσεις

7. Δείκτες

8. Αλφαριθμητικά

9. Διαχείριση μνήμης

10. Υλοποίηση δομών δεδομένων και αλγορίθμων στη C

11. Είσοδος/Εξοδος

12. Ο προπεξεργαστής της C

13. Διαμέριση κώδικα

14. Ανάπτυξη οδηγούμενη από ελέγχους

15. Αποσφαλμάτωση

16. Στατική και δυναμική ανάλυση κώδικα

17. Βιβλιοθήκες

18. Θέματα στυλ στη C

19. Γραφικό περιβάλλον διεπαφής

20. Αλληλεπίδραση της C με την Python

A. Διαμόρφωση περιβάλλοντος ανάπτυξης κώδικα σε C

B. Μεταγλώττιση και εκτέλεση κώδικα C χρησιμοποιώντας το Docker

Γ. Λήψη πληροφοριών για το σύστημα και το μεταγλωττιστή

6.10 Ασκήσεις

Άσκηση 1

Να γραφεί πρόγραμμα που να χρησιμοποιεί εγγραφές ενός τύπου δεδομένων που αναπαριστά κύκλους θεωρώντας ότι κάθε κύκλος ορίζεται από το κέντρο του (δύο τιμές για τις συντεταγμένες x και y) και την ακτίνα του (radius). Το πρόγραμμα να διαθέτει συνάρτηση ανάγνωσης των στοιχείων ενός κύκλου και συναρτήσεις εκτύπωσης του εμβαδού και της περιμέτρου του κύκλου.

Λύση άσκησης 1

```
1 #include <stdio.h>
2 #include <math.h>
3
4 // Δήλωση του τύπου δεδομένων για τον κύκλο
5 typedef struct {
6     double x;
7     double y;
8     double radius;
9 } circle;
10
11 // Συνάρτηση ανάγνωσης των στοιχείων ενός κύκλου
12 circle read_circle(void) {
13     circle c;
14     printf("Δώσε τη συντεταγμένη x του κέντρου του κύκλου: ");
15     scanf("%lf", &c.x);
16     printf("Δώσε τη συντεταγμένη y του κέντρου του κύκλου: ");
17     scanf("%lf", &c.y);
18     printf("Δώσε την ακτίνα του κύκλου: ");
19     scanf("%lf", &c.radius);
20     return c;
21 }
22
23 // Συνάρτηση εκτύπωσης του εμβαδού του κύκλου
24 void print_area(circle c) {
25     double area = M_PI * c.radius * c.radius;
26     printf("Το εμβαδόν του κύκλου είναι: %lf\n", area);
27 }
28
29 // Συνάρτηση εκτύπωσης της περιμέτρου του κύκλου
30 void print_perimeter(circle c) {
31     double perimeter = 2 * M_PI * c.radius;
32     printf("Η περιμέτρος του κύκλου είναι: %lf\n", perimeter);
33 }
34
35 int main(void) {
36     circle c = read_circle();
37     print_area(c);
38     print_perimeter(c);
39     return 0;
40 }
```

Πίνακας περιεχομένων

6.1 Δήλωση δομών

6.2 Χρήση του typedef

6.3 Αρχικοποίηση δομής

6.4 Συναρτήσεις και δομές

6.5 Σύγκριση δομών

6.6 Δομές με δομές ως πεδία

6.7 Πίνακες δομών

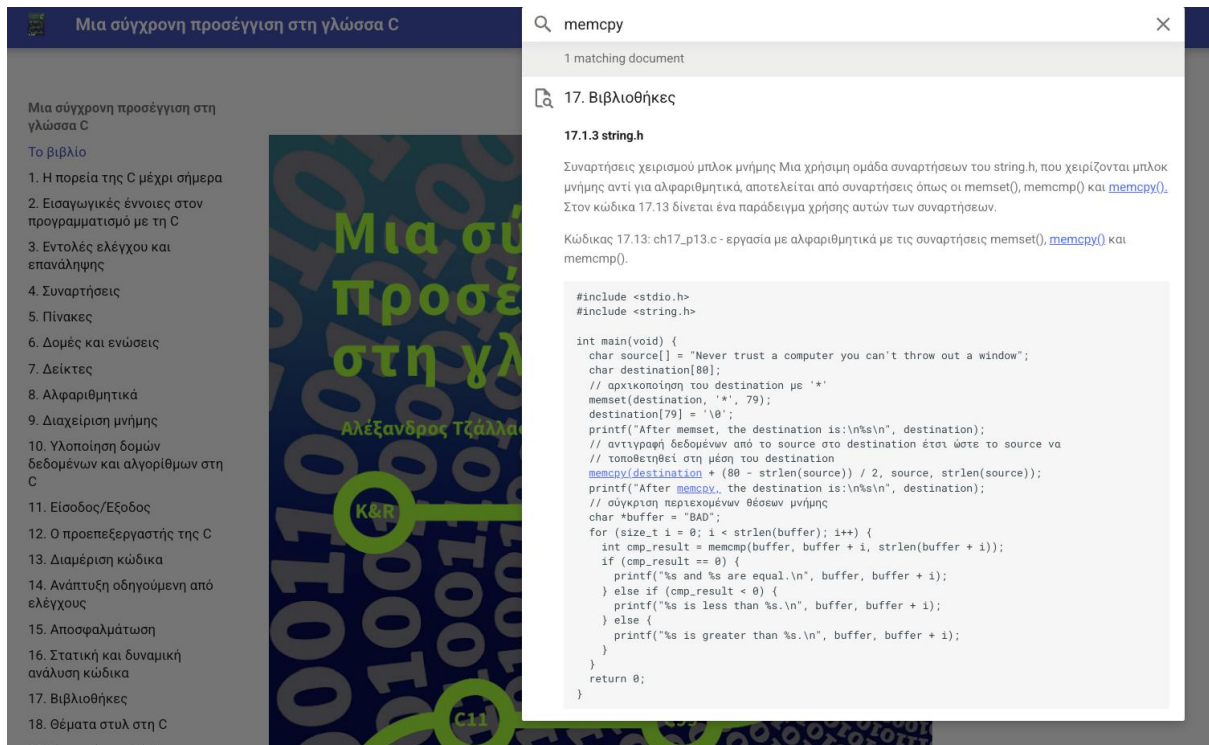
6.8 Ενώσεις

6.9 Δομές και ενώσεις

6.10 Ασκήσεις

Εικόνα 32. Παράδειγμα σελίδας με κώδικα (foldable)

45



Εικόνα 33. Αναζήτηση σε όλο το κείμενο του βιβλίου