

Νίκος Μ. Χατζηγιαννάκης

Η γλώσσα **Python** σε βάθος

Καλύπτει την έκδοση **3** της Python

Περιλαμβάνει εισαγωγή
στην επιστήμη των υπολογιστών
και τον προγραμματισμό



Πρόλογος

Όταν συνειδητοποίησα ότι η χρήση της Python αυξάνεται με πολύ γρήγορους ρυθμούς και άρχισε να διδάσκεται όλο και περισσότερο στα ξένα αλλά και στα ελληνικά πανεπιστήμια, από περιέργεια άρχισα να ρίχνω κάποιες κλεφτές ματιές στη νέα για μένα αυτή γλώσσα. Βλέποντάς την επίσης να διδάσκεται ως πρώτη γλώσσα προγραμματισμού αναρωτιόμουν αν επιτέλους, μετά από πολλούς πειραματισμούς, η ακαδημαϊκή κοινότητα βρήκε την κατάλληλη γλώσσα για την εισαγωγή σπουδαστών στον προγραμματισμό.

Προερχόμενος από μια πολύ μακρά θητεία σε άλλες γλώσσες προγραμματισμού και ιδιαίτερα στη C και C++, τις οποίες διδάσκω και έχω ασχοληθεί συγγραφικά, η πρώτη μου εντύπωση, όταν άρχισα να ασχολούμαι με την Python, ήταν λίγο αρνητική.

Λόγω της ενασχόλησής μου με την εκπαίδευση, πάντα αξιολογώ μια γλώσσα από την εκπαιδευτική και όχι την παραγωγική της πλευρά. Οι ευκολίες και τα πλούσια εργαλεία που παρέχει η Python σίγουρα κάνουν τη ζωή του προγραμματιστή πολύ πιο εύκολη. Σε τελική ανάλυση, όσα περισσότερα κάνει μια γλώσσα προγραμματισμού τόσο το καλύτερο για εμάς. Ένας έμπειρος προγραμματιστής γνωρίζει τις αρχές του προγραμματισμού και τους μηχανισμούς που κρύβονται πίσω από τις λειτουργίες μιας γλώσσας, και τον ενδιαφέρει μόνο η παραγωγικότητα. Τι γίνεται όμως με έναν αρχάριο χρήστη; Μπορεί αυτός ο χρήστης να διδαχθεί και να κατανοήσει σε βάθος τη φιλοσοφία του προγραμματισμού μέσω της Python;

Όταν άρχισα να γράφω αυτό το βιβλίο, οι γνώσεις μου στην Python ήταν τόσες όσες είναι πιθανόν τώρα οι δικές σας, δηλαδή μηδενικές! Το βιβλίο το έγραφα σταδιακά και παράλληλα με τη δική μου ενασχόληση με τη γλώσσα. Διάλεξα αυτόν τον θρασύ και ανορθόδοξο τρόπο γιατί ήθελα να αντιμετωπίσω τα ίδια ερωτήματα και προβληματισμούς με τον αρχάριο που θα έρθει για πρώτη φορά αντιμέτωπος με τη γλώσσα Python.

Η Python διαφέρει αρκετά από τις «κλασικές» γλώσσες προγραμματισμού όπως η C, η C++ και η Java, οι οποίες ούτως ή άλλως έχουν κοινή ρίζα, και απαιτεί διαφορετική εκπαιδευτική προσέγγιση. Έχοντας την εμπειρία από τη διδασκαλία άλλων γλωσσών προγραμματισμού, προσπάθησα να αποδώσω κάθε έννοια της γλώσσας με απλό, κατανοητό και εποπτικό τρόπο. Δεν έμεινα όμως μόνο σε αυτό, αφού στόχος μου ήταν να αναδείξω τους μηχανισμούς που κρύβονται πίσω από μια τόσο υψηλού επιπέδου γλώσσα, χρησιμοποιώντας ενίοτε παραδείγματα και τεχνικές άλλων γλωσσών προγραμματισμού.

Ο κύριος σκοπός μου με αυτό το βιβλίο δεν είναι μόνο να βοηθήσω τον αναγνώστη να μάθει τη γλώσσα Python, αλλά να αποκτήσει επίσης τις βάσεις ώστε να μπορέσει να μάθει σχετικά εύκολα οποιαδήποτε άλλη γλώσσα προγραμματισμού συναντήσει στο μέλλον!

Το αν πέτυχα αυτόν τον σκοπό, θα το κρίνετε εσείς...

Ο σκοπός του βιβλίου

Ο βασικός σκοπός του βιβλίου είναι η εις βάθος κατανόηση των εντολών και μηχανισμών της Python 3 και παράλληλα η εκμάθηση των αρχών προγραμματισμού, οι οποίες μπορούν να εφαρμοστούν σε κάθε γλώσσα.

Όσοι αναγνώστες έχουν κάποια εμπειρία στον προγραμματισμό και έχουν ήδη χρησιμοποιήσει άλλες γλώσσες, ίσως θεωρήσουν ορισμένα κεφάλαια, ιδιαίτερα τα πρώτα, κάπως «φλύαρα».

Βασικό μου μέλημα είναι η βαθιά κατανόηση των γενικών αρχών μιας γλώσσας προγραμματισμού, από τα πρώτα κίολας κεφάλαια, ώστε ο αναγνώστης να αποκτήσει το υπόβαθρο για να κατανοεί πιο σύνθετες έννοιες και εντολές που θα συναντήσει αργότερα. Το βιβλίο δεν στοχεύει στην επιφανειακή γνώση και στη «μηχανική» χρήση της γλώσσας, αλλά στην ουσιαστική κατανόηση των ιδιαίτερων χαρακτηριστικών και της φιλοσοφίας της.

Πώς να διαβάσετε αυτό το βιβλίο

Το πρώτο κεφάλαιο του βιβλίου απευθύνεται κυρίως σε όσους δεν έχουν βασικές γνώσεις πληροφορικής, και εισάγει τον αναγνώστη στις βασικές έννοιες της επιστήμης της Πληροφορικής. Ακόμα και αν πιστεύετε ότι γνωρίζετε τα βασικά για τους ηλεκτρονικούς υπολογιστές (H/Y), θα σας συμβούλευα να διαβάσετε έστω και γρήγορα αυτό το κεφάλαιο. Είμαι βέβαιος ότι όλο και κάτι καινούριο θα μάθετε, θα φρεσκάρετε τις γνώσεις σας, και πιθανώς να ξεκαθαρίσετε περισσότερο κάποια πράγματα που ήδη γνωρίζετε.

Το πρώτο κεφάλαιο δεν αποτελεί προϋπόθεση για την κατανόηση των υπόλοιπων κεφαλαίων και επομένως, αν σας ενδιαφέρει να ασχοληθείτε άμεσα με την Python, μπορείτε να το παραλείψετε, ή ίσως να το διαβάσετε τμηματικά, παράλληλα με τα υπόλοιπα κεφάλαια.

Το δεύτερο κεφάλαιο απευθύνεται σε όσους ξεκινούν τώρα με τον προγραμματισμό. Αν είστε γνώστης μιας οποιασδήποτε άλλης γλώσσας προγραμματισμού, μπορείτε να το παραλείψετε και αυτό και να προχωρήσετε κατευθείαν στο επόμενο κεφάλαιο.

Το τρίτο κεφάλαιο είναι ό,τι ακριβώς δηλώνει ο τίτλος του: μια πρώτη ματιά στην Python! Γίνεται μια πρώτη γνωριμία με τη γλώσσα ώστε να αποκτήσετε μια σφαιρική εικόνα για τη δομή και τα χαρακτηριστικά της και να μπορέσετε να σχεδιάσετε τα πρώτα σας απλά προγράμματα. Αν έχετε εμπειρία στον προγραμματισμό, ίσως σας φανεί κάπως «φλύαρο». Σε αυτήν την περίπτωση, θα σας συμβούλευα να του ρίξετε μια γρήγορη ματιά και να προχωρήσετε στο επόμενο κεφάλαιο.

Το τέταρτο κεφάλαιο είναι από τα πιο σημαντικά κεφάλαια του βιβλίου. Σε αυτό αναλύεται ο ιδιαίτερος τρόπος με τον οποίο η Python διαχειρίζεται τη μνήμη και παρουσιάζονται οι βασικές οντότητες (τύποι δεδομένων) που υποστηρίζει η γλώσσα.

Τα Κεφάλαια 5 και 6 αναλύουν τις δομές ελέγχου και επανάληψης της γλώσσας, ενώ στα Κεφάλαια 7 και 8 περιγράφεται σε βάθος η δυνατότητα τμηματοποίησης των προγραμμάτων με χρήση συναρτήσεων, καθώς και κάποιες εξειδικευμένες τεχνικές προγραμματισμού με απλό και εποπτικό τρόπο.

Στα Κεφάλαια 9 και 10 παρουσιάζονται οι σύνθετες οντότητες της Python με ιδιαίτερη έμφαση στις λίστες, οι οποίες αποτελούν τον κύριο τύπο αποθήκευσης και διαχείρισης μεγάλου όγκου δεδομένων. Γίνεται επίσης αναφορά στους πίνακες (ή *συστοιχίες*, arrays), που χρησιμοποιούνται σε άλλες γλώσσες προγραμματισμού, και αναλύονται εξειδικευμένοι αλγόριθμοι, όπως της ταξινόμησης και της αναζήτησης.

Το Κεφάλαιο 11 αναφέρεται στον τρόπο με τον οποίο η Python διαχειρίζεται τα αρχεία. Δηλαδή, τον τρόπο με τον οποίο μπορούμε να διαβάζουμε και να γράφουμε πληροφορίες στην περιφερειακή μνήμη του Η/Υ μας.

Η Python είναι μια κατ' εξοχήν αντικειμενοστρεφής γλώσσα. Όλες τις οι οντότητές της είναι αντικείμενα κλάσεων. Στα πρώτα κεφάλαια χρησιμοποιούμε τις ενσωματωμένες κλάσεις της Python για να δημιουργούμε και να διαχειριζόμαστε αντικείμενα. Στο Κεφάλαιο 12 θα μάθουμε πώς μπορούμε να ορίζουμε τις δικές μας κλάσεις, καθώς και αντικείμενα αυτών των κλάσεων. Στα Κεφάλαια 13, 14 και 15, που ακολουθούν, αναλύονται και παρουσιάζονται στη πράξη όλες οι έννοιες του αντικειμενοστρεφούς προγραμματισμού.

Τέλος, στο Κεφάλαιο 16 παρουσιάζεται ο τρόπος με τον οποίο μπορούμε να διαχειριστούμε τα σφάλματα τα οποία είναι πιθανόν να προκύψουν κατά την εκτέλεση ενός προγράμματος της Python.

Στο Παράρτημα Α θα βρείτε γενικές οδηγίες για την εγκατάσταση της Python, καθώς και λεπτομέρειες για το ολοκληρωμένο περιβάλλον ανάπτυξης **IDLE**, το οποίο είναι ενσωματωμένο στις τελευταίες εκδόσεις της γλώσσας. Όλα τα προγράμματα που υπάρχουν σε αυτό το βιβλίο έχουν δοκιμαστεί στο περιβάλλον του **IDLE** (Python 3.11).

Το Παράρτημα Β περιέχει ένα ελληνοαγγλικό λεξικό της ορολογίας που θα συναντήσετε στο παρόν βιβλίο, το οποίο θα σας φανεί πολύ χρήσιμο αν χρειαστεί να διαβάσετε κάποιο αγγλικό βιβλίο, ή για να αναζητήσετε πληροφορίες στο Διαδίκτυο.

Το Σχήμα Π.1 σάς ξαναγεί στο βιβλίο, και περιγράφει συνοπτικά τα περιεχόμενα των κεφαλαίων και των παραρτημάτων του.

Στο **συνοδευτικό υλικό** του βιβλίου θα βρείτε τον κώδικα των προγραμμάτων του βιβλίου, καθώς και τις απαντήσεις όλων των ασκήσεων και των εργασιών. Μπορείτε να κατεβάσετε το συμπιεσμένο αρχείο του συνοδευτικού υλικού από την ιστοσελίδα www.klidarithmos.gr/PythonInDepth, καθώς και από τον ιστότοπο του βιβλίου, στη διεύθυνση <http://python.bytes.gr>.

Σας συνιστώ να κατεβάσετε τώρα το συνοδευτικό υλικό του βιβλίου και να εγκαταστήσετε την Python, ώστε να μπορείτε να δοκιμάζετε τα παραδείγματα του βιβλίου αλλά και τα δικά σας προγράμματα.

Το σύμβολο ☉ δίπλα από τον τίτλο μιας ενότητας υποδεικνύει ότι η ενότητα παρέχει εξειδικευμένη γνώση, την οποία μπορείτε να παραλείψετε αν δεν θέλετε να εμβαθύνετε σε τέτοιο βαθμό.

Στην πρώτη γραμμή πάνω από τον κώδικα των περισσότερων παραδειγμάτων αυτού του βιβλίου αναφέρεται το όνομα του αρχείου στο οποίο υπάρχει ο πηγαίος κώδικας του παραδείγματος. Αν έχετε ήδη εγκαταστήσει στα Windows την Python και το IDLE που τη συνοδεύει, αρκεί να πατήσετε με το δεξιό πλήκτρο του ποντικιού επάνω στο αρχείο του παραδείγματος (με προέκταση .py) και να επιλέξετε την επιλογή «Edit with IDLE», για να ανοίξετε το αρχείο μέσα στο ολοκληρωμένο περιβάλλον του IDLE. Έπειτα μπορείτε να εκτελέσετε το πρόγραμμα με ένα απλό πάτημα του πλήκτρου F5, ή επιλέγοντας την αντίστοιχη εντολή από το μενού του περιβάλλοντος (δείτε το Παράρτημα Α).

Διαβάστε το βιβλίο με... παρέα

Και δεν εννοώ με κάποιον φίλο ή φίλη σας, αλλά με τον αχώριστο σύντροφο που θα πρέπει να έχετε κάθε στιγμή στο πλευρό σας όταν διαβάζετε αυτό το βιβλίο, και δεν είναι άλλος από το περιβάλλον του IDLE!

Όπως συνηθίζω να λέω στους φοιτητές μου, ο προγραμματισμός είναι σαν το ποδήλατο! Για να μάθεις ποδήλατο πρέπει να ανέβεις σε αυτό και να προσπαθήσεις, δεν γίνεται να μάθεις διαβάζοντας απλώς κάποιο εγχειρίδιο χρήσης.

Κατά την ανάγνωση του βιβλίου, σας συμβουλεύω να έχετε πάντα ανοιχτό το IDLE, ώστε να δοκιμάζετε άμεσα ότι διδάσκεστε αλλά και να κάνετε τους δικούς σας πειραματισμούς.

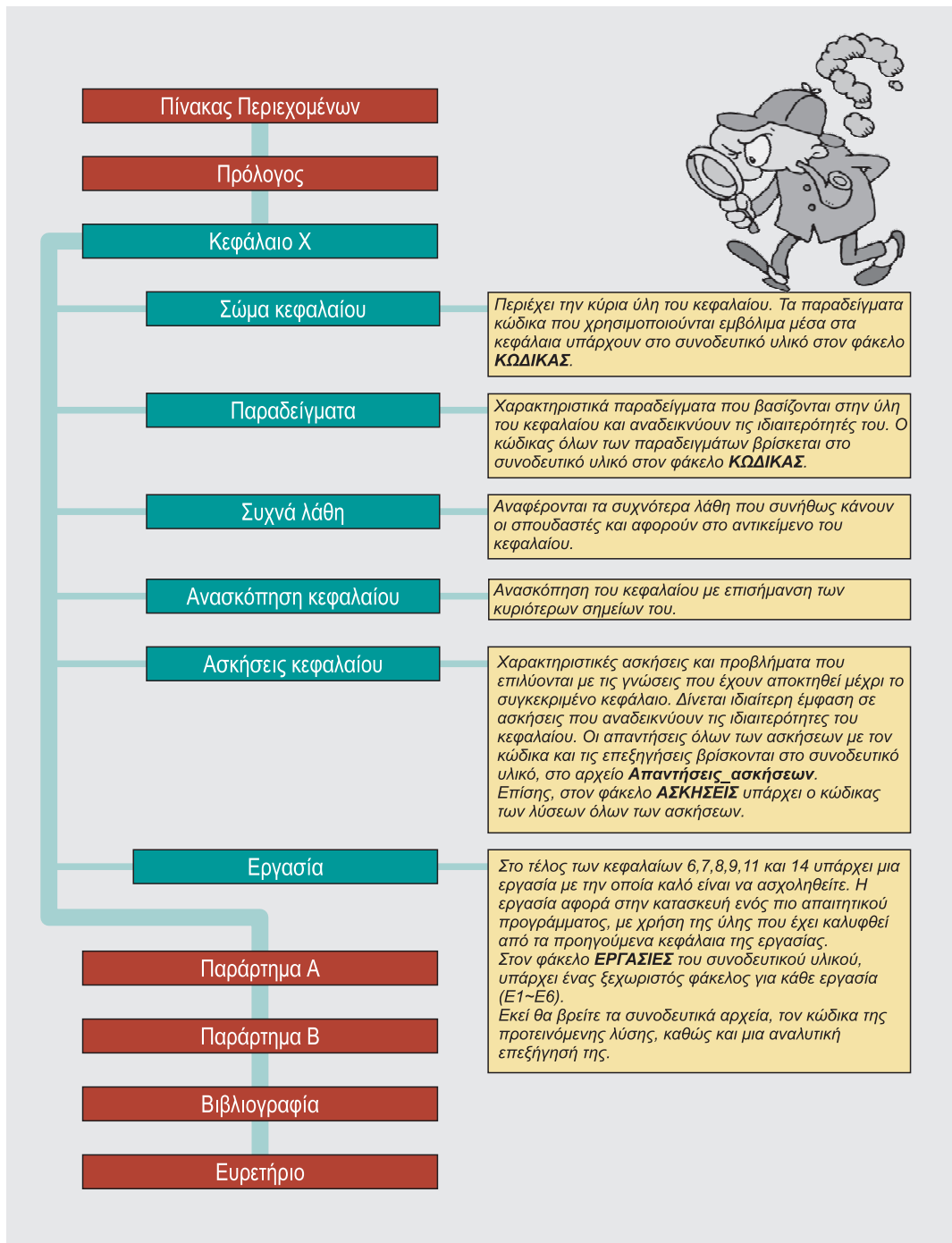
Η δομή των κεφαλαίων

Όλα τα κεφάλαια του βιβλίου έχουν την ίδια δομή (Σχήμα Π.2). Στο τέλος κάθε κεφαλαίου υπάρχουν λυμένα παραδείγματα και ασκήσεις προς επίλυση. Ο βαθμός δυσκολίας κάθε άσκησης υποδεικνύεται από έναν αριθμό αστερίσκων (από έναν αστερίσκο για τις εύκολες ασκήσεις μέχρι τρεις για τις πιο απαιτητικές). Σε κάθε κεφάλαιο αναφέρονται τα συχνότερα λάθη που κάνουν οι σπουδαστές, ενώ υπάρχει επίσης μια μικρή ανασκόπηση με τα βασικότερα σημεία του κεφαλαίου. Μετά από ορισμένα κεφάλαια υπάρχουν εργασίες τις οποίες προτείνω να προσπαθήσετε να ολοκληρώσετε. Για την ολοκλήρωση κάθε εργασίας ίσως χρειαστεί να αφιερώσετε αρκετές ώρες ή και ημέρες. Στο συνοδευτικό υλικό υπάρχουν οι απαντήσεις των ασκήσεων, ο κώδικας των λύσεών τους, καθώς και ο κώδικας όλων των προγραμμάτων που υπάρχουν στο βιβλίο. Επίσης, στον φάκελο ΕΡΓΑΣΙΕΣ θα βρείτε τις επεξηγήσεις και τον κώδικα των λύσεων των εργασιών.

Το παρόν βιβλίο αφορά την έκδοση 3 της Python, στην οποία έχουν γίνει σημαντικές προσθήκες και αλλαγές σε σχέση με την προηγούμενη έκδοση 2. Αυτό έχει ως αποτέλεσμα να μην υπάρχει συμβατότητα μεταξύ των δύο εκδόσεων και προγράμματα που γράφονται με την έκδοση 3 της Python δεν μπορούν να εκτελεστούν στη 2 και αντίστροφα.

1	Βασικές έννοιες πληροφορικής Εισάγει τον αναγνώστη στην επιστήμη της Πληροφορικής, γνωρίζοντάς του τις βασικές της έννοιες.
2	Εισαγωγή στον προγραμματισμό Απευθύνεται στους αρχάριους στον προγραμματισμό. Αναφέρει και εξηγεί βασικές έννοιες του προγραμματισμού, απαραίτητες για τον αναγνώστη που έρχεται για πρώτη φορά σε επαφή με μια γλώσσα προγραμματισμού.
3	Μια πρώτη ματιά στην Python Παρουσιάζει τα πολύ βασικά στοιχεία της γλώσσας και στις έννοιες που πρέπει να γνωρίζετε για να δημιουργήσετε τα πρώτα σας απλά προγράμματα.
4	Διαχείριση μνήμης και οντότητες της Python Εμβαθύνει στον τρόπο με τον οποίο η Python διαχειρίζεται τη μνήμη του υπολογιστή και εξηγείται πλήρως ο δυναμικός τρόπος δημιουργίας και χρήσης των μεταβλητών στη Python. Αναφέρονται οι διάφορες οντότητες της γλώσσας και γίνεται μια πρώτη εισαγωγή στις βασικές έννοιες του αντικειμενοστρεφούς προγραμματισμού.
5	Η δομή επιλογής if Εδώ παρουσιάζεται η εντολή if με την οποία μπορούμε να εκτελούμε προτάσεις του προγράμματος «πό συνθήκη».
6	Δομές επανάληψης Το κεφάλαιο αυτό ασχολείται με τις εντολές μέσω των οποίων μπορούμε να επιτύχουμε επαναληπτική εκτέλεση προτάσεων.
7	Συναρτήσεις Στο κεφάλαιο αυτό παρουσιάζεται η δυνατότητα τμηματοποίησης του προγράμματος με τη χρήση συναρτήσεων.
8	Εξειδικευμένα θέματα συναρτήσεων Εδώ παρουσιάζονται εξειδικευμένα θέματα συναρτήσεων τα οποία συναντούμε σπανιότερα.
9	Λίστες Αναλύεται σε βάθος η χρήση των λιστών για την αποθήκευση και διαχείριση δεδομένων.
10	Περισσότερα για τους σύνθετους τύπους δεδομένων Εδώ εξετάζονται σε μεγαλύτερο βάθος οι σύνθετοι τύποι δεδομένων, όπως οι συμβολοσειρές, οι πλειάδες, τα σύνολα και τα λεξικά.
11	Διαχείριση αρχείων Εδώ παρουσιάζεται ο τρόπος με τον οποίο η Python διαχειρίζεται τα αρχεία.
12	Κλάσεις και αντικείμενα Το κεφάλαιο αυτό παρέχει μια ομαλή μετάβαση από τη φιλοσοφία του «κλασικού» διαδικασιακού προγραμματισμού στον μαγικό κόσμο του αντικειμενοστρεφούς προγραμματισμού.
13	Επιπρόσθετα θέματα κλάσεων Εδώ παρουσιάζονται τα περισσότερο εξειδικευμένα θέματα σχετικά με τις κλάσεις.
14	Υπερφόρτωση τελεστών Στο κεφάλαιο αυτό αναλύεται η υπερφόρτωση τελεστών, δηλαδή η δυνατότητα να καθορίσουμε τη συμπεριφορά ενός τελεστή σε σχέση με τα αντικείμενα των κλάσεων που δημιουργούμε.
15	Κληρονομικότητα Εδώ αναλύεται η κληρονομικότητα, δηλαδή η δυνατότητα να δημιουργούμε νέες κλάσεις που να βασίζονται σε άλλες υπάρχουσες κλάσεις.
16	Εξαιρέσεις Στο κεφάλαιο αυτό παρουσιάζεται η δυνατότητα διαχείρισης των εξαιρέσεων, δηλαδή των σημαντικών χρόνου εκτέλεσης ενός προγράμματος.
A	Εγκατάσταση της Python Εδώ παρουσιάζεται η διαδικασία εγκατάστασης της Python και η βασική χρήση του IDLE.
B	Αντιστοιχισμός ελληνικών και αγγλικών όρων Εδώ παρατίθενται οι ελληνικοί όροι του βιβλίου με τους αντίστοιχους αγγλικούς.

Σχήμα Π.1 Μια ξενάγηση στο βιβλίο



Σχήμα Π.2 Η δομή των κεφαλαίων

Κεφάλαιο 1

Βασικές έννοιες πληροφορικής

Το κεφάλαιο αυτό εισάγει τον αναγνώστη στην επιστήμη της Πληροφορικής, γνωρίζοντάς του τις βασικές έννοιες.

Ξεκινάει απαντώντας στο πολύ βασικό ερώτημα: Τι είναι ένας ηλεκτρονικός υπολογιστής (Η/Υ); Στη συνέχεια αναλύει τη δομή του, τις βασικές αρχές λειτουργίας του, καθώς και τον τρόπο με τον οποίο ο Η/Υ αντιλαμβάνεται και επεξεργάζεται τις πληροφορίες.

Μια ξεχωριστή ενότητα του κεφαλαίου είναι αφιερωμένη στον προγραμματισμό Η/Υ, όπου αναλύεται σε βάθος η λογική με την οποία ένας Η/Υ αντιλαμβάνεται και εκτελεί τις εντολές ενός προγράμματος.

Γίνεται ειδική αναφορά στα δίκτυα και στις επικοινωνίες, και επεξηγείται πλήρως η φιλοσοφία λειτουργίας τους.

Επισημαίνεται η αναγκαιότητα της ασφάλειας των δεδομένων και παρουσιάζονται οι διάφοροι τρόποι με τους οποίους επιτυγχάνεται. Τέλος, παρουσιάζονται οι βασικές αρχές της ρομποτικής, μιας επιστήμης που θα μας απασχολήσει πολύ στο κοντινό μέλλον.

Ακόμα και αν πιστεύετε ότι γνωρίζετε τα βασικά για τους Η/Υ, θα σας συμβούλευα να διαβάσετε προσεκτικά αυτό το κεφάλαιο. Είμαι βέβαιος ότι όλο κάτι καινούριο θα μάθετε, θα φρεσκάρετε τις γνώσεις σας, και πιθανώς να ξεκαθαρίσετε περισσότερο κάποια πράγματα που ήδη γνωρίζετε.

1.1 Τι είναι η Πληροφορική

Η Πληροφορική είναι η επιστήμη που ασχολείται με τη διαχείριση πληροφοριών, δηλαδή με τη συλλογή, την επεξεργασία, την αποθήκευση και τη μετάδοσή τους. Με την κυριολεκτική έννοια του όρου, η Πληροφορική υπάρχει από τότε που εμφανίστηκε ο άνθρωπος. Για παράδειγμα, το 450 π.Χ., στη μάχη του Μαραθώνα, ο Φειδιππίδης συνέλεξε την πληροφορία (παρατήρησε τη μάχη), την επεξεργάστηκε (διαπίστωσε ότι νίκησαν οι Έλληνες), τη μετέδωσε (έτρεξε με τα πόδια στην Αθήνα για να μεταφέρει τα καλά νέα) και ίσως να την αποθήκευσε χαράζοντάς την πάνω σε κάποια μαρμάρινη στήλη!

Όμως, η Πληροφορική άρχισε να μας απασχολεί σοβαρά με την εμφάνιση των ηλεκτρονικών υπολογιστών οι οποίοι έδωσαν τη δυνατότητα της επεξεργασίας, αποθήκευσης και μετάδοσης μεγάλου όγκου πληροφοριών, με πολύ μεγάλη ταχύτητα.

1.2 Τι είναι ένας ηλεκτρονικός υπολογιστής;

Η πιο γνωστή «μηχανή» που μπορεί να κάνει υπολογισμούς και να παίρνει αποφάσεις είναι ο ανθρώπινος νους.

Παρά το γεγονός ότι ο νους του ανθρώπου είναι ένα τρομερά πολύπλοκο υπολογιστικό σύστημα, παρουσιάζει περιορισμένη ικανότητα στην ταχύτητα, απομνημόνευση και επεξεργασία μεγάλου όγκου πληροφοριών.

Για να αντιμετωπίσει αυτό το μειονέκτημα, ο άνθρωπος, κατά καιρούς, επινόησε και κατασκεύασε μηχανές που θα μπορούσαν να βοηθήσουν τη μνήμη του και να αυξήσουν τις περιορισμένες υπολογιστικές του ικανότητες.

Οι υπολογιστές, κατά κάποιο τρόπο, όχι μόνο υποκαθιστούν αλλά και βελτιώνουν ορισμένες λειτουργίες του ανθρώπινου νου. Είναι ικανοί να επιλύουν υπολογιστικά και λογικά προβλήματα, να απομνημονεύουν στοιχεία, να τα επεξεργάζονται και να εξάγουν αποτελέσματα.

Ας δούμε όμως πού υπερτερεί ο ηλεκτρονικός υπολογιστής έναντι του ανθρώπου. Ασφαλώς, ένας Η/Υ επεξεργάζεται τα στοιχεία με πολύ μεγαλύτερη ταχύτητα και ακρίβεια από τον άνθρωπο. Ένας απλός Η/Υ μπορεί να εκτελέσει δισεκατομμύρια πράξεις το δευτερόλεπτο! Επίσης, ο Η/Υ μπορεί να αποθηκεύει τεράστιους όγκους δεδομένων σε πολύ μικρό χώρο με τη δυνατότητα να μπορεί να τις εντοπίζει πολύ γρήγορα. Όμως, ο Η/Υ εξακολουθεί να είναι μια μηχανή που δεν μπορεί να σκέφτεται και να αναλαμβάνει πρωτοβουλίες από μόνος του. Για να κάνει οτιδήποτε, πρέπει να του το υποδείξει ένας άνθρωπος.

Ηλεκτρονικός υπολογιστής είναι ένα μηχάνημα το οποίο δέχεται δεδομένα, τα επεξεργάζεται με μεγάλη ταχύτητα (εκτελεί τις λογικές και αριθμητικές πράξεις) με τρόπο προκαθορισμένο από έναν άνθρωπο χρήστη, και εξάγει αποτελέσματα. Ο Η/Υ επίσης διαθέτει μορφές μνήμης στις οποίες μπορεί να αποθηκεύσει στοιχεία, μόνιμα ή προσωρινά, κατά τέτοιο τρόπο ώστε να είναι ανά πάσα στιγμή διαθέσιμα.

Όπως ήδη έχει αναφερθεί, οι λειτουργίες του Η/Υ προκαθορίζονται από τον άνθρωπο. Αυτό επιτυγχάνεται με τη χρήση ειδικών «προτάσεων» (εντολών), τις οποίες ο Η/Υ αντιλαμβάνεται και οι οποίες αντιστοιχούν σε συγκεκριμένες λειτουργίες.

Ένα σύνολο εντολών το οποίο καθορίζει μια ακολουθία λειτουργιών του Η/Υ καλείται **πρόγραμμα**.

1.3 Υλικό και λογισμικό

Ο Η/Υ είναι ένα μηχάνημα χωρίς καμιά δυνατότητα σκέψης και δημιουργικότητας. Οι λειτουργίες που εκτελεί είναι προκαθορισμένες και βασίζονται σε ένα σύνολο εντολών (το πρόγραμμα), το οποίο έχει σχεδιαστεί από έναν άνθρωπο χρήστη και είναι διαφορετικό για την κάθε εφαρμογή. Τα ηλεκτρονικά κυκλώματα του Η/Υ ερμηνεύουν τις εντολές και εκτελούν τις λειτουργίες που είναι απαραίτητες για κάθε εντολή.

Η επιστήμη των Η/Υ χωρίζεται επομένως σε δύο μεγάλα πεδία, το πεδίο του **υλικού** (hardware) και το πεδίο του **λογισμικού** (software).

Ο τομέας του υλικού ασχολείται με τον εξοπλισμό των Η/Υ, δηλαδή με τη σχεδίαση των ηλεκτρονικών κυκλωμάτων και των περιφερειακών μονάδων που τον υποστηρίζουν (οθόνη, εκτυπωτής, πληκτρολόγιο, κλπ.). Το υλικό βασίζεται εξ ολοκλήρου στην επιστήμη της ηλεκτρονικής, χάρη στην οποία μπορούμε να μιλάμε σήμερα για υπολογιστικά συστήματα. Έτσι, με τη λέξη «υλικό» (hardware) αναφερόμαστε στα κύρια ηλεκτρονικά συστήματα του Η/Υ, καθώς και στις περιφερειακές μονάδες που τον απαρτίζουν. Το πεδίο του λογισμικού (software) ασχολείται με τον σχεδιασμό και την κατασκευή των προγραμμάτων που καθορίζουν τη λειτουργία του Η/Υ.

Το λογισμικό και το υλικό είναι αλληλένδετα πεδία. Δεν νοείται Η/Υ χωρίς πρόγραμμα και πρόγραμμα χωρίς Η/Υ. Αν παραλληλίσουμε τη βιομηχανία των Η/Υ με τη βιομηχανία των αυτοκινήτων, το υλικό ασχολείται με την κατασκευή του αυτοκινήτου και το λογισμικό με την παρασκευή των καυσίμων (βενζίνη, πετρέλαιο, υγραέριο). Πρόκειται για δύο εντελώς διαφορετικά πεδία, αλλά άρρηκτα συνδεδεμένα μεταξύ τους. Ένας Η/Υ χωρίς πρόγραμμα είναι τόσο χρήσιμος όσο ένα αυτοκίνητο χωρίς καύσιμα!

1.4 Η δομή ενός Η/Υ

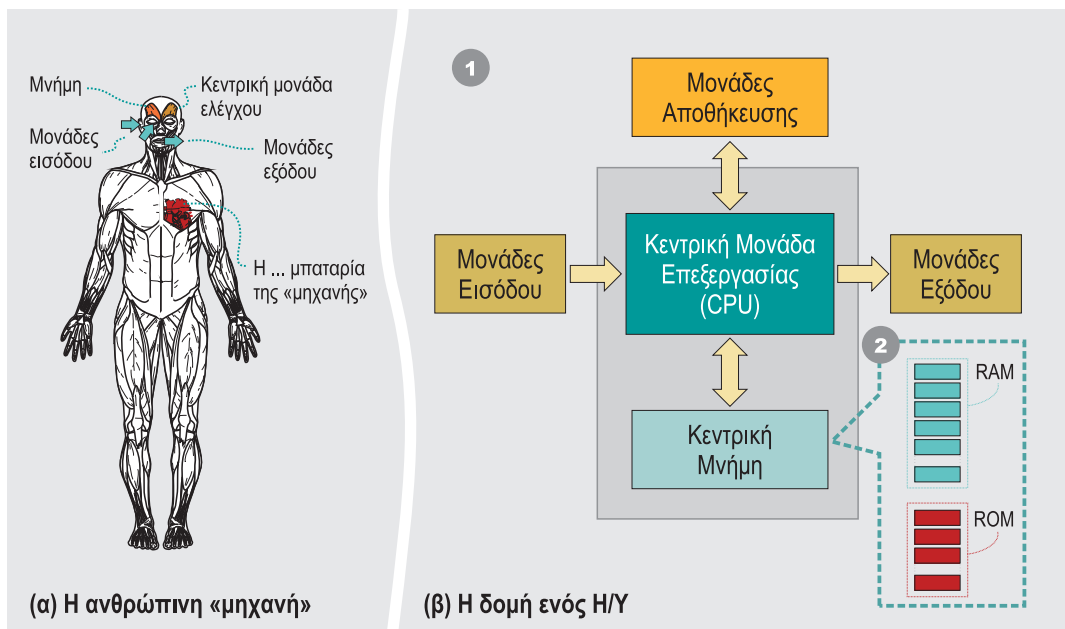
Για να γίνει πιο κατανοητή η δομή ενός Η/Υ, θα παραλληλίσουμε τη λειτουργία του με τις λειτουργίες του ανθρώπινου σώματος. Ας θεωρήσουμε λοιπόν τον άνθρωπο, όχι ως έναν βιολογικό οργανισμό, αλλά ως μια εκτελεστική μηχανή. Προφανώς, η κεντρική μονάδα ελέγχου αυτής της μηχανής είναι ο ανθρώπινος εγκέφαλος. Αυτός είναι υπεύθυνος για τη λειτουργία των οργάνων, δίνει εντολές στα διάφορα τμήματα του σώματος, δέχεται και αξιολογεί τα διάφορα ερεθίσματα από το περιβάλλον και είναι υπεύθυνος για τη

λήψη αποφάσεων. Επίσης ο εγκέφαλος έχει τη δυνατότητα απομνημόνευσης πληροφοριών, γεγονός που προσδίδει στον άνθρωπο την ικανότητα της μνήμης.

Ο άνθρωπος διαθέτει μονάδες με τις οποίες επικοινωνεί με το εξωτερικό του περιβάλλον, όπως μονάδες από τις οποίες λαμβάνει πληροφορίες (π.χ. τα μάτια, τα αυτιά, οι αισθητήρες αφής) οι οποίες λέγονται μονάδες εισόδου, και μονάδες μέσω των οποίων δίνει πληροφορίες στο εξωτερικό του περιβάλλον (π.χ. το στόμα για ομιλία), οι οποίες λέγονται μονάδες εξόδου (Σχήμα 1.1α).

Ο ανθρώπινος εγκέφαλος επικοινωνεί με τα διάφορα όργανα του σώματος μέσω του νευρικού συστήματος, από το οποίο δέχεται τα διάφορα ερεθίσματα και στέλνει τις διαταγές του. Η καρδιά του ανθρώπου παίζει τον ρόλο της «μπαταρίας» η οποία παρέχει συνεχώς την ενέργεια που χρειάζεται για να δουλέψει η μηχανή που λέγεται άνθρωπος.

Η δομή του Η/Υ είναι παρόμοια με τη δομή της «ανθρώπινης μηχανής» και φαίνεται στο Σχήμα 1.1β1. Ο «εγκέφαλος» του Η/Υ καλείται Κεντρική Μονάδα Επεξεργασίας (Central Processing Unit -CPU) και δεν είναι τίποτε άλλο από ένα ηλεκτρονικό κύκλωμα το οποίο ελέγχει όλα τα επί μέρους τμήματα του Η/Υ και είναι υπεύθυνο για την εκτέλεση όλων των λειτουργιών του. Είναι αυτό το οποίο εκτελεί τις εντολές, τις αριθμητικές και τις λογικές πράξεις. Η CPU διαθέτει επίσης περιορισμένο αριθμό εσωτερικών θέσεων μνήμης που λέγονται καταχωρητές. Οι καταχωρητές χρησιμοποιούνται για την προσωρινή αποθήκευση των δεδομένων που διαβάζονται από την κεντρική μνήμη αλλά και των αποτελεσμάτων της επεξεργασίας τους.



Σχήμα 1.1 Η δομή ενός Η/Υ

Η CPU συνδέεται με τις μονάδες εισόδου (Input) και εξόδου (Output) για την επικοινωνία του Η/Υ με το περιβάλλον του. Η CPU συνδέεται επίσης με την κεντρική μνήμη του Η/Υ, η οποία αποτελεί μια προσωρινή μονάδα αποθήκευσης πληροφοριών. Μπορούμε να φανταστούμε την κεντρική μνήμη του Η/Υ ως δισεκατομμύρια κουτάκια το ένα επάνω στο άλλο, μέσα στα οποία ο Η/Υ μπορεί να καταχωρίζει πληροφορίες.

Η κεντρική μνήμη χωρίζεται σε δύο επιμέρους τμήματα: τη μνήμη ROM (Read Only Memory – Μνήμη μόνο για ανάγνωση) και τη μνήμη RAM (Random Access Memory – Μνήμη τυχαίας προσπέλασης). Από τη μνήμη ROM ο Η/Υ μπορεί να διαβάσει μόνον τις πληροφορίες οι οποίες έχουν αποθηκευτεί σε αυτήν από τον κατασκευαστή. Τα περιεχόμενα της μνήμης ROM δεν μπορούν να τροποποιηθούν. Από την άλλη, ο Η/Υ χρησιμοποιεί τη μνήμη RAM για να αποθηκεύει αλλά και για να διαβάζει πληροφορίες (Σχήμα 1.1β2).

Μια άλλη ουσιαστική διαφορά μεταξύ ROM και RAM είναι ότι οι πληροφορίες που υπάρχουν στη ROM διατηρούνται και όταν ο Η/Υ τεθεί εκτός λειτουργίας. Αντίθετα, ό,τι πληροφορία υπάρχει αποθηκευμένη στη RAM χάνεται σε περίπτωση διακοπής ρεύματος ή απενεργοποίησης του Η/Υ.

Αν και η μνήμη ROM διατηρεί τις πληροφορίες της, δεν μπορούμε να αποθηκεύσουμε τίποτε σε αυτή. Αντίθετα, στη μνήμη RAM μπορούμε να αποθηκεύσουμε πληροφορίες οι οποίες όμως χάνονται μόλις ο Η/Υ τεθεί εκτός λειτουργίας.

Αυτή η ιδιαιτερότητα της μνήμης RAM κάνει επιτακτική την ανάγκη χρήσης και ενός άλλου τύπου μνήμης για τη μόνιμη αποθήκευση πληροφοριών. Αυτή η μνήμη μόνιμης αποθήκευσης λέγεται περιφερειακή μνήμη και υλοποιείται με τη χρήση μαγνητικών και οπτικών μέσων (σκληροί δίσκοι, μαγνητικές ταινίες, CD, DVD), αλλά και μέσων στερεάς κατάστασης (π.χ. μονάδες αποθήκευσης USB, κάρτες μνήμης), όπου καταγράφονται μόνιμα οι πληροφορίες.

1.5 Πώς επεξεργάζεται ο Η/Υ τις πληροφορίες;

Όλα τα επιμέρους τμήματα που αποτελούν έναν Η/Υ δεν είναι τίποτε άλλο παρά ηλεκτρονικά κυκλώματα. Πώς όμως αυτά τα ηλεκτρονικά κυκλώματα αντιλαμβάνονται, επεξεργάζονται και αποθηκεύουν τις πληροφορίες που δίνουμε; Πώς επιτυγχάνεται η επικοινωνία μεταξύ των διαφόρων αυτών μονάδων ενός Η/Υ;

Πριν όμως προσπαθήσουμε να απαντήσουμε στις παραπάνω ερωτήσεις, ας αναφέρουμε τους τρόπους με τους οποίους είναι δυνατόν να γίνει μετάδοση πληροφοριών.

Θα χρησιμοποιήσω ένα πολύ απλό παράδειγμα ανταλλαγής πληροφοριών για να γίνουν κατανοητοί οι δύο τρόποι μετάδοσης πληροφοριών, ο αναλογικός και ο ψηφιακός. Ο ψηφιακός είναι ο τρόπος μετάδοσης πληροφοριών στους σύγχρονους Η/Υ.

Αναλογική και ψηφιακή επικοινωνία

Ας θεωρήσουμε δύο άτομα, ένα αγόρι και ένα κορίτσι. Το αγόρι θέλει να στείλει μηνύματα στο κορίτσι και ως μέσο επικοινωνίας χρησιμοποιεί μια λάμπα της οποίας η φωτεινότητα μπορεί να μεταβάλλεται με έναν ειδικό διακόπτη (Σχήμα 1.2α₁). Η λάμπα μπορεί να είναι τελείως σβηστή, με πλήρη ένταση, ή με οποιοδήποτε ενδιάμεση ένταση φωτεινότητας. Ο τρόπος μετάδοσης πληροφοριών επιτυγχάνεται με ρύθμιση της φωτεινότητας της λάμπας. Όταν ρυθμίζει τη λάμπα σε διαφορετική ένταση, το αγόρι μπορεί να στείλει διαφορετικά μηνύματα στο κορίτσι. Έτσι, στο παράδειγμα του Σχήματος 1.2α₂, το αγόρι μπορεί να ρυθμίσει την ένταση της λάμπας σε τέσσερις διαφορετικές διαβαθμίσεις για να στείλει στο κορίτσι ισάριθμα μηνύματα.

Η μέθοδος με την οποία χρησιμοποιούμε ένα κανάλι επικοινωνίας (π.χ. μια λάμπα), το οποίο μπορεί να πάρει διάφορες τιμές (ένταση φωτεινότητας), καλείται **αναλογική**.

Στην αναλογική επικοινωνία, το πλήθος των πληροφοριών που μπορούν να μεταδοθούν από το αγόρι στο κορίτσι του παραδείγματος, εξαρτάται από τον αριθμό των διαφορετικών εντάσεων που μπορεί να πάρει η λάμπα, οι οποίες είναι θεωρητικά άπειρες.

Όσο πιο πολλές όμως είναι οι διαφορετικές διαβαθμίσεις, τόσο μεγαλύτερη είναι η πιθανότητα λάθους στην αναγνώριση της σωστής έντασης από το κορίτσι. Επομένως, με λιγότερες διαβαθμίσεις της έντασης του φωτός η πιθανότητα λάθους αναγνώρισης είναι μικρότερη. Την μικρότερη πιθανότητα λάθους την έχουμε όταν οι διαβαθμίσεις της έντασης είναι μόνο δύο: Σβηστή λάμπα και αναμμένη λάμπα! Με αυτόν τον τρόπο, όμως, μπορούν να σταλούν μόνο δύο μηνύματα, για παράδειγμα:

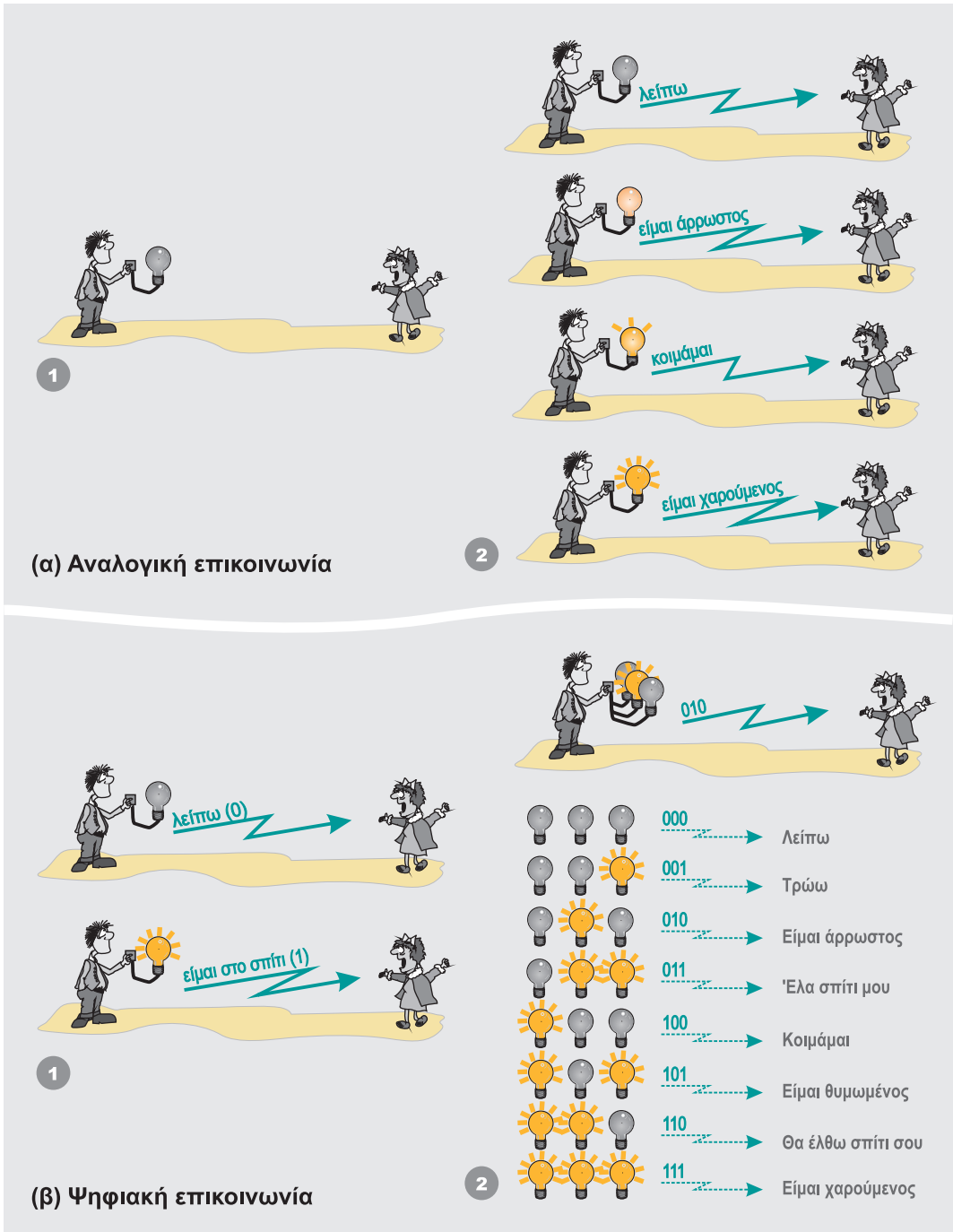
- Σβηστή λάμπα → Λείπω
- Αναμμένη λάμπα → Είμαι σπίτι

Στην περίπτωση αυτή το κανάλι επικοινωνίας (η λάμπα) μπορεί να λάβει μόνο δύο τιμές (αναμμένη ή σβηστή) και συνεπώς ο αριθμός των πληροφοριών που μπορεί να μεταδώσει είναι μόνο δύο (Σχήμα 1.2β₁).

Ο τρόπος μετάδοσης πληροφοριών που χρησιμοποιεί κανάλια επικοινωνίας τα οποία μπορούν να μεταδώσουν μόνο δύο καταστάσεις (την ύπαρξη ή μη ύπαρξη κάποιου συμβάντος) καλείται **ψηφιακός**.

Η μόνη λύση για να μπορέσουμε να μεταδώσουμε περισσότερες πληροφορίες με αυτόν τον τρόπο είναι να χρησιμοποιήσουμε περισσότερα κανάλια.

Έτσι, στο παραπάνω παράδειγμα, αν το αγόρι έχει στη διάθεσή του δύο λάμπες, ο αριθμός των πληροφοριών που θα μπορούσε να μεταδώσει με ψηφιακό τρόπο θα ήταν τέσσερις, ενώ αν είχε 3 λάμπες θα ήταν 8 (Σχήμα 1.2β₂).



Σχήμα 1.2 Αναλογική και ψηφιακή επικοινωνία

Όταν αυξάνεται ο αριθμός των λαμπτήρων αυξάνεται και το πλήθος των πληροφοριών που μπορούν να μεταδοθούν. Με n λάμπες, το πλήθος των διαφορετικών πληροφοριών είναι 2^n (π.χ. με 8 λάμπες είναι 2^8 δηλαδή 256). Αν συμβολίσουμε την ύπαρξη φωτός με 1 και τη μη ύπαρξη με 0 μπορούμε να πούμε ότι το αγόρι στέλνει στο κορίτσι μια σειρά από 1 και 0. Ανάλογα με τον συνδυασμό που λαμβάνει το κορίτσι, αντιλαμβάνεται και το αντίστοιχο μήνυμα. Στο παράδειγμα του Σχήματος 1.2β₂ με τις τρεις λάμπες, οι πιθανοί συνδυασμοί είναι 8, επομένως ισάριθμο είναι και το πλήθος μηνυμάτων που είναι δυνατόν να σταλούν. Παρατηρούμε ότι ο αναλογικός τρόπος μετάδοσης εκμεταλλεύεται περισσότερο τα υπάρχοντα κανάλια επικοινωνίας, μεταδίδοντας μέσα από κάθε κανάλι ένα μεγάλο πλήθος πληροφοριών, αλλά με αυξημένες πιθανότητες λάθους. Αντίθετα, η ψηφιακή μέθοδος μεταδίδει μόνο δύο καταστάσεις μέσα από κάθε κανάλι, κάνοντας επομένως επιτακτική τη χρήση περισσότερων καναλιών για τη μετάδοση περισσότερων πληροφοριών. Όμως η ψηφιακή επικοινωνία μάς εξασφαλίζει τη μικρότερη δυνατή πιθανότητα λάθους και είναι η μέθοδος που χρησιμοποιείται στους Η/Υ.

Οι σύγχρονοι Η/Υ αποτελούνται εξ ολοκλήρου από ψηφιακά κυκλώματα τα οποία αναγνωρίζουν μόνο δύο καταστάσεις: την ύπαρξη (1) ή μη ύπαρξη (0) τάσης στους ακροδέκτες τους.

Τα διάφορα κυκλώματα που αποτελούν τον Η/Υ συνδέονται μεταξύ τους με καλώδια. Οι πληροφορίες από το ένα στο άλλο μεταδίδονται με σειρές παλμών τάσης.

Bit και byte

Η πιο βασική μορφή πληροφορίας είναι η ύπαρξη ή η μη ύπαρξη κάποιου συμβάντος (στη συγκεκριμένη περίπτωση, ηλεκτρικής τάσης). Κωδικοποιώντας τις δύο αυτές καταστάσεις, αντιστοιχίζουμε την τιμή 1 στην ύπαρξη ενός συμβάντος (π.χ. ύπαρξη τάσης, ή αναμμένη λάμπα) και την τιμή 0 στη μη ύπαρξη του συμβάντος (π.χ. μη ύπαρξη τάσης, ή σβηστή λάμπα). Αυτός ακριβώς είναι και ο ορισμός του **bit** που αποτελεί τη μικρότερη μονάδα πληροφορίας στους Η/Υ.

Το bit είναι μια μονάδα πληροφορίας η οποία μπορεί να λάβει μόνο δύο τιμές. Την τιμή 1 ή την τιμή 0 (Σχήμα 1.3α₁, στη σελίδα 43).

Με τον συνδυασμό περισσότερων bit, το πλήθος των πληροφοριών που μπορούν να μεταδοθούν αυξάνεται. Για παράδειγμα, ενώ με 1 bit μπορούν να κωδικοποιηθούν μόνο δύο καταστάσεις, με 2 bit μπορούν να κωδικοποιηθούν τέσσερις, με 3 bit μπορούν να κωδικοποιηθούν οκτώ και, γενικά, με n bit μπορούν να κωδικοποιηθούν 2^n καταστάσεις.

Ένας συνδυασμός 8 bit ονομάζεται **byte** και μπορεί να δημιουργήσει 256 διαφορετικούς συνδυασμούς τού 0 και τού 1 (Σχήμα 1.3α₂, στη σελίδα 43). Το byte είναι η βασική μονάδα μέτρησης πληροφοριών στους Η/Υ. Η ανταλλαγή πληροφοριών μεταξύ των τμημάτων του Η/Υ και μεταξύ του Η/Υ και των περιφερειακών μονάδων του γίνεται με byte.

Πέρα από το byte υπάρχουν και τα πολλαπλάσιά του, τα οποία αναφέρονται στον Πίνακα 1.1. Οι περισσότεροι θα έχετε σίγουρα ακούσει τις έννοιες του kilobyte, megabyte και gigabyte ως μονάδες μέτρησης ψηφιακών δεδομένων.

Τα τελευταία χρόνια, που το μέγεθος των σκληρών δίσκων έχει αυξηθεί, αρχίζει να μας απασχολεί ως μονάδα μέτρησης της χωρητικότητας και το terabyte. Οι υπόλοιπες μονάδες του πίνακα αφορούν πολύ μεγάλα μεγέθη τα οποία δεν συναντά συχνά ο μέσος χρήστης, όμως αυτό είναι βέβαιο ότι θα αλλάξει τα επόμενα χρόνια.

Όσοι έχετε ασχοληθεί με την πληροφορική, αλλά και όσοι έχετε αποκτήσει μόνο τις βασικές γνώσεις Η/Υ από το σχολείο, ενδεχομένως να έχετε κάποιες ενστάσεις παρατηρώντας την πρώτη στήλη του πίνακα. Θα μάθατε ή θα διαβάσατε ότι ένα kilobyte είναι 1024 και όχι 1000 byte, ότι ένα megabyte είναι 1024 kilobytes, ένα gigabyte 1024 megabytes, κοκ.

Η αλήθεια είναι ότι για πολλά χρόνια αυτό ίσχυε, και πολλοί μεγαλώσαμε με αυτές τις μονάδες. Αλήθεια είναι επίσης ότι οι μονάδες αυτές δημιουργούσαν μια αναντιστοιχία: Για παράδειγμα γιατί ένα **kilobyte** να είναι 1024 byte και ένα **kilometer** να είναι 1000 μέτρα, ένα **megabyte** να είναι 1.048.576 ($1024 \cdot 1024$) byte και ένα **megawatt** να είναι 1.000.000 ($1000 \cdot 1000$) watt;

Υπήρχε λοιπόν το παράδοξο το ίδιο πρόθεμα μονάδων (kilo, mega, giga, κ.λπ.) να σημαίνει διαφορετικό πολλαπλάσιο όταν πρόκειται για byte και διαφορετικό όταν πρόκειται για άλλα μεγέθη (κιλά, μέτρα, ενέργεια, κ.λπ.). Ο λόγος για αυτήν την αναντιστοιχία είναι ότι για τη μέτρηση των μονάδων byte χρησιμοποιείτο το δυαδικό σύστημα ενώ για τα υπόλοιπα μεγέθη το δεκαδικό (μετρικό). Όμως το πρόθεμα ήταν το ίδιο!

Το 1998 αποφασίστηκε και θεσπίστηκε ένα διαφορετικό πρόθεμα για το δυαδικό σύστημα μονάδων. Έτσι, το δυαδικό kilobyte (kB) έγινε kibibyte (KiB), το δυαδικό megabyte (MB) mebibyte (MiB), κοκ. Οι μονάδες αυτές αναγράφονται στη δεύτερη στήλη του πίνακα. Το πρόβλημα είναι ότι πολλοί λίγοι γνωρίζουν την ύπαρξή τους και ακόμα λιγότεροι τις χρησιμοποιούν. Οι εταιρείες σκληρών δίσκων, πάντως, έχουν υιοθετήσει το δεκαδικό σύστημα, διότι έτσι εμφανίζουν μεγαλύτερες τιμές για τις χωρητικότητες των δίσκων τους!

Το γεγονός είναι ότι όταν κάποιος αναφέρεται, για παράδειγμα, σε kilobyte (kB), δεν είμαστε σίγουροι τι εννοεί. Χρησιμοποιεί τον παλαιό δυαδικό ορισμό με 1024 byte, ή τον καινούριο δεκαδικό ορισμό με 1000; Σας αφήνω να αποφασίσετε μόνοι σας αν θέλετε να ακολουθήσετε τα σημερινά ισχύοντα πρότυπα του πίνακα, ή αν θέλετε να παραμείνετε αγκυλωμένοι σε αυτό που οι περισσότεροι θεωρούν σωστό και δυστυχώς αναφέρεται ακόμα συχνά στη σύγχρονη βιβλιογραφία!

Πολλαπλάσια του byte			
Δεκαδικό σύστημα μονάδων ¹		Δυαδικό σύστημα μονάδων ²	
Συντομογραφία και περιγραφή	Τιμή	Συντομογραφία και περιγραφή	Τιμή
kB kilobyte = 1000 byte	1000 (10^3)	KiB kibibyte = 1024 byte	1024 (2^{10})
MB megabyte = 1000 kB	1000 ² (10^6)	MiB mebibyte = 1024 KiB	1024 ² (2^{20})
GB gigabyte = 1000 MB	1000 ³ (10^9)	GiB gibibyte = 1024 MiB	1024 ³ (2^{30})
TB terabyte = 1000 GB	1000 ⁴ (10^{12})	TiB tebibyte = 1024 GiB	1024 ⁴ (2^{40})
PB petabyte = 1000 TB	1000 ⁵ (10^{15})	PiB pebibyte = 1024 TiB	1024 ⁵ (2^{50})
EB exabyte = 1000 PB	1000 ⁶ (10^{18})	EiB exbibyte = 1024 PiB	1024 ⁶ (2^{60})
ZB zettabyte = 1000 EB	1000 ⁷ (10^{21})	ZiB zebibyte = 1024 EiB	1024 ⁷ (2^{70})
YB yottabyte = 1000 ZB	1000 ⁸ (10^{24})	YiB yobibyte = 1024 ZiB	1024 ⁸ (2^{80})

Πίνακας 1.1 Πολλαπλάσια του byte

Κάθε μορφή πληροφορίας και χωρητικότητας στους Η/Υ μετρείται σε byte. Για παράδειγμα, οι απαντήσεις σε ερωτήσεις όπως οι παρακάτω, αναφέρονται όλες σε byte.

- Πόσες πληροφορίες χωράει μια εξωτερική μονάδα αποθήκευσης USB;
- Ποιο είναι το μέγεθος του σκληρού μου δίσκου;
- Πόσο χώρο καταλαμβάνει αυτό το κείμενο;
- Πόση είναι η κεντρική μνήμη του Η/Υ μου;
- Τι μέγεθος έχει αυτή η ψηφιακή φωτογραφία;
- Τι χωρητικότητα έχει η κάρτα μνήμης του κινητού μου;

Μετατροπή κάθε πληροφορίας σε 1 και 0

Παρατηρούμε ότι το bit, ως μονάδα πληροφορίας με δύο μόνο δυνατές καταστάσεις, 1 και 0, μας παραπέμπει στο δυαδικό σύστημα, όπου τα μοναδικά ψηφία είναι το 1 και το 0. Όλες οι επικοινωνίες και λειτουργίες του Η/Υ βασίζονται σε δυαδικούς αριθμούς. Είναι λοιπόν εμφανές ότι για να επεξεργαστεί μια πληροφορία ο Η/Υ, αυτή θα πρέπει να βρίσκεται σε δυαδική μορφή.

Από την εμπειρία μας στη χρήση Η/Υ γνωρίζουμε ότι ένας Η/Υ μπορεί να επεξεργαστεί και να αποθηκεύσει πληροφορίες σε μορφή δεκαδικών αριθμών, κειμένων, εικόνων, ήχων και βίντεο. Πώς είναι δυνατόν όμως ο Η/Υ να επεξεργάζεται αυτές τις πληροφορίες από τη στιγμή που, σύμφωνα με τα παραπάνω, κατανοεί μόνο δυαδικούς αριθμούς;

¹ Αναφέρεται στο διεθνές σύστημα μονάδων (International System of Units – SI) το οποίο χρησιμοποιεί ως βάση αρίθμησης τις δυνάμεις του δέκα. Εφαρμόζεται στα περισσότερα μετρήσιμα μεγέθη όπως η απόσταση, η μάζα, η θερμοκρασία, η ένταση ρεύματος, κ.λπ.

² Αναφέρεται σε ένα νέο πρότυπο της διεθνούς ηλεκτροτεχνικής επιτροπής (International Electrotechnical Commission – IEC) το οποίο εισήχθη το 1998 και αφορά ένα σύστημα μονάδων με βάση αρίθμησης τις δυνάμεις του 2. Εφαρμόζεται σε μεγέθη ψηφιακών δεδομένων όπως μέγεθος μνήμης, χωρητικότητα περιφερειακών μονάδων, μέγεθος αρχείων, κ.λπ.

Η απάντηση είναι η εξής:

Οποιαδήποτε πληροφορία χειρίζεται ένας Η/Υ μετατρέπεται σε 1 και 0, που είναι και το μοναδικό είδος πληροφοριών το οποίο μπορεί να κατανοήσει και να επεξεργαστεί. Η διαδικασία αυτή λέγεται **ψηφιοποίηση** (digitization).

Με ποια λογική όμως γίνεται η ψηφιοποίηση; Πώς μετατρέπονται σε 1 και 0 όλες οι πληροφορίες που αναφέραμε προηγουμένως; Ας μελετήσουμε κάθε περίπτωση ξεχωριστά:

Δεκαδικοί και δυαδικοί αριθμοί ☉

Το σύμβολο ☉ υποδεικνύει ενότητα η οποία μπορεί να παραληφθεί σε πρώτη ανάγνωση.

Υπάρχει ένας μαθηματικός τρόπος μετατροπής των δεκαδικών αριθμών που χρησιμοποιεί ο άνθρωπος σε δυαδικούς αριθμούς, με μοναδικά ψηφία το 1 και το 0. Για παράδειγμα, το 5 μετατρέπεται σε 101, το 10 σε 1010, το 15 σε 1111, κλπ. Επομένως, είναι μια αυτοματοποιημένη και εύκολη υπολογιστικά διαδικασία.

Ένας δυαδικός αριθμός έχει ως βάση αρίθμησης το 2 (όπως ο δεκαδικός το 10). Κάθε δυαδικός αριθμός έχει έναν αντίστοιχο δεκαδικό, και αντιστρόφως. Τα ψηφία ενός αριθμού (δεκαδικού ή δυαδικού) αποτελούν τους συντελεστές των δυνάμεων της βάσης του (10 ή 2 αντίστοιχα).

Συντελεστής	Δύναμη του 10	Σύνολο ψηφίου
4	$\times 1 (10^0)$	4
0	$\times 10 (10^1)$	0
2	$\times 100 (10^2)$	200
1	$\times 1000 (10^3)$	1000
Σύνολο		1204

Ας αναλύσουμε, για παράδειγμα, τον δεκαδικό αριθμό 1204. Από τα δεξιά προς τα αριστερά, το 4 είναι ο συντελεστής του 1 (10^0), το 0 του 10 (10^1), το 2 του 100 (10^2), και το 1 του 1000 (10^3). Επομένως ο αριθμός είναι:

$$1 \cdot 1000 + 2 \cdot 100 + 0 \cdot 10 + 4 \cdot 1$$

δηλαδή 1204.

Αντίστοιχα, τα ψηφία ενός δυαδικού αριθμού αποτελούν τους συντελεστές των δυνάμεων του 2 (και όχι του 10), δηλαδή 1, 2, 4, 8, 16, 32, 64, κοκ.

Ας εξετάσουμε, για παράδειγμα, τον δυαδικό αριθμό $(1101)_2$. Από τα δεξιά προς τα αριστερά και πάλι, το 1 είναι ο συντελεστής του 1 (2^0), το 0 του 2 (2^1), το 1 του 4 (2^2), και το 1 του 8 (2^3). Επομένως, ο αριθμός υπολογίζεται από την παράσταση:

$$1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1$$

δηλαδή είναι ο δεκαδικός αριθμός 13.

Συντελεστής	Δύναμη του 2	Σύνολο ψηφίου
1	$\times 1 (2^0)$	1
0	$\times 2 (2^1)$	0
1	$\times 4 (2^2)$	4
1	$\times 8 (2^3)$	8
Σύνολο		13

Μετατροπή δεκαδικού σε δυαδικό

Κάθε δεκαδικός αριθμός έχει έναν αντίστοιχο δυαδικό. Ο τρόπος με τον οποίο βρίσκουμε τον αντίστοιχο δυαδικό αριθμό ενός δεκαδικού αναλύεται με το παρακάτω παράδειγμα, όπου βρίσκουμε τον αντίστοιχο δυαδικό αριθμό του 52. Θα ακολουθήσουμε τα επόμενα βήματα:

$52/2 \rightarrow$ πηλίκο 26, υπόλοιπο 0 (το 0 αποτελεί τον συντελεστή του 2^0)

$26/2 \rightarrow$ πηλίκο 13, υπόλοιπο 0 (το 0 αποτελεί τον συντελεστή του 2^1)

$13/2 \rightarrow$ πηλίκο 6, υπόλοιπο 1 (το 1 αποτελεί τον συντελεστή του 2^2)

$6/2 \rightarrow$ πηλίκο 3, υπόλοιπο 0 (το 0 αποτελεί τον συντελεστή του 2^3)

$3/2 \rightarrow$ πηλίκο 1, υπόλοιπο 1 (το 1 αποτελεί τον συντελεστή του 2^4)

$1/2 \rightarrow$ πηλίκο 0, υπόλοιπο 1 (το 1 αποτελεί τον συντελεστή του 2^5)

Το γεγονός ότι στην τελευταία διαίρεση το πηλίκο είναι 0 σηματοδοτεί το τέλος της διαδικασίας. Επομένως, ο αντίστοιχος δυαδικός αριθμός του 52 είναι ο $(110100)_2$. Πράγματι, αν εκτελέσουμε τις πράξεις θα προκύψει ο αριθμός 52:

$$1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 \rightarrow 32 + 16 + 0 + 4 + 0 + 0 \rightarrow 52$$

Η διαδικασία μετατροπής του αριθμού 52 στον αντίστοιχο δυαδικό αριθμό φαίνεται στο διπλανό σχήμα. Διαιρούμε το 52 με το 2 και παίρνουμε υπόλοιπο 0 και πηλίκο 26. Το υπόλοιπο αποτελεί το τελευταίο ψηφίο του δυαδικού αριθμού. Έπειτα διαιρούμε το πηλίκο που βρήκαμε (26) πάλι με το 2. Το νέο υπόλοιπο αποτελεί το προτελευταίο ψηφίο του δυαδικού αριθμού. Αυτό συνεχίζεται μέχρι το πηλίκο να γίνει 0. Το τελευταίο υπόλοιπο αποτελεί το πρώτο ψηφίο του δυαδικού αριθμού. Έτσι, το **52** μετατρέπεται στον δυαδικό αριθμό: **$(110100)_2$** .



Δεκαεξαδικοί αριθμοί και δεκαεξαδικές σταθερές ☉

Υπάρχουν περιπτώσεις (κυρίως όταν προγραμματίζουμε συστήματα ή συσκευές σε πολύ χαμηλό επίπεδο) στις οποίες θα μας διευκόλυνε να χρησιμοποιούσαμε ακέραιες σταθερές στο δεκαεξαδικό σύστημα αρίθμησης.

Το δεκαεξαδικό σύστημα αρίθμησης έχει ως βάση τον αριθμό 16 (αντί του 10), και περιλαμβάνει 16 ψηφία: Τα ψηφία από το 0 μέχρι το 9 και τα γράμματα από το A μέχρι το F (τα οποία χρησιμοποιούνται για τους δεκαδικούς αριθμούς από το 10 μέχρι το 15).

Μια ακέραια δεκαεξαδική σταθερά πρέπει να αρχίζει με 0x (μηδέν και λατινικό x).

0xFF → αντιστοιχεί στον δεκαδικό αριθμό 255 ($15 \cdot 16^1 + 15 \cdot 16^0$)

0x1A → αντιστοιχεί στον δεκαδικό αριθμό 26 ($1 \cdot 16^1 + 10 \cdot 16^0$)

Οι δεκαεξαδικοί αριθμοί χρησιμοποιούνται πολύ συχνά για τον έλεγχο συσκευών σε επίπεδο δυαδικών αριθμών, αφού ένας δεκαεξαδικός αριθμός μπορεί να μετατραπεί άμεσα και εύκολα στον αντίστοιχο δυαδικό του. Ο διπλάνος πίνακας δείχνει τα 16 ψηφία του δεκαεξαδικού συστήματος αρίθμησης, με τον αντίστοιχο δεκαδικό και δυαδικό αριθμό στον οποίον μετατρέπονται. Παρατηρούμε ότι ένα δεκαεξαδικό ψηφίο μετατρέπεται σε 4 bit, δηλαδή σε μισό byte. Ένα byte επομένως αναπαρίσταται με δύο δεκαεξαδικά ψηφία.

11111111 → **FF**

00011010 → **1A**

Ας μελετήσουμε τον δυαδικό αριθμό 00011010 και ας χωρίσουμε το byte σε δύο τμήματα των τεσσάρων bit το καθένα: 0001 και 1010. Σύμφωνα με τον παραπάνω πίνακα, το πρώτο τμήμα μετατρέπεται στο δεκαεξαδικό ψηφίο 1 και το δεύτερο στο ψηφίο A, οπότε ο δεκαεξαδικός που προκύπτει είναι ο 1A. Το αντίστροφο είναι επίσης πολύ εύκολο: για παράδειγμα ο δεκαεξαδικός B6DE μετατρέπεται άμεσα στον δυαδικό **101101101101110** (ο δεκαδικός 46814)! Αντίθετα, όπως είδαμε, η μετατροπή ενός δεκαδικού αριθμού σε δυαδικό και αντίστροφα είναι μια πιο πολύπλοκη διαδικασία.

Χαρακτήρες

Η μετατροπή χαρακτήρων κειμένου σε δυαδικούς αριθμούς απαιτεί μια διαφορετική προσέγγιση. Σε αυτήν την περίπτωση, η μετατροπή δεν γίνεται με κάποιο μαθηματικό τρόπο, αλλά κατόπιν συμφωνημένων συμβάσεων (προτύπων), και κάθε χαρακτήρας μετατρέπεται σε μια συγκεκριμένη ακολουθία από 1 και 0. Η διαδικασία αυτή λέγεται κωδικοποίηση χαρακτήρων (character encoding) και βασίζεται σε κάποια πρότυπα.

Δεκαεξαδικό ψηφίο	Δεκαδικό ψηφίο/αριθμός	Δυαδικός αριθμός
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Κωδικοποίηση χαρακτήρων – Πρότυπο ASCII

Από τα πρώτα και πιο δημοφιλή πρότυπα κωδικοποίησης είναι το εκτεταμένο πρότυπο ASCII³, σύμφωνα με το οποίο κάθε χαρακτήρας μετατρέπεται σε οχτώ 1 και 0, δηλαδή σε ένα byte. Με τα 8 bit του προτύπου ASCII μπορούμε να κωδικοποιήσουμε μέχρι 256 χαρακτήρες, αριθμός πολύ περιορισμένος αν αναλογιστούμε τα αλφάβητα που υπάρχουν στις διαφορετικές γλώσσες του κόσμου. Ο αριθμός που προκύπτει από την κωδικοποίηση είναι ένας δεκαδικός αριθμός από το 0 μέχρι το 255 και λέγεται κωδικός ASCII και μπορεί να υπολογιστεί με βάση τον Πίνακα 1.2.

Ο κωδικός ASCII ενός χαρακτήρα προκύπτει από το άθροισμα του αριθμού της γραμμής και της στήλης που βρίσκεται στον πίνακα. Έτσι, ο χαρακτήρας 'Γ' έχει κωδικό 130 (128+2), ενώ ο χαρακτήρας 'W' έχει κωδικό 87 (80+7). Η τυπική κωδικοποίηση ASCII περιλαμβάνει τους 128 πρώτους χαρακτήρες, με κωδικούς από το 0 έως το 127. Οι χαρακτήρες από το 128 μέχρι το 255 αποτελούν επέκταση του πρότυπου πίνακα χαρακτήρων, και διαφέρουν από σύστημα σε σύστημα και από γλώσσα σε γλώσσα. Οι χαρακτήρες με κωδικούς από 0 μέχρι 31 δεν είναι εκτυπώσιμοι χαρακτήρες, και κάποιοι από αυτούς χρησιμοποιούνται ως χαρακτήρες ελέγχου. Χαρακτηριστικό παράδειγμα χαρακτήρων ελέγχου αποτελούν οι χαρακτήρες διαφυγής της Python, στους οποίους θα αναφερθούμε στη συνέχεια.

Κωδικοποίηση χαρακτήρων – Πρότυπο Unicode

Με βάση τον επεκταμένο πίνακα κωδικοποίησης ASCII μπορούν να κωδικοποιηθούν το πολύ 256 χαρακτήρες, συμπεριλαμβανομένων και των χαρακτήρων ελέγχου. Είναι λοιπόν προφανές ότι είναι αδύνατον να συμπεριληφθούν κωδικοί για αλφάβητα άλλων γλωσσών εκτός της Αγγλικής, αλλά και γλωσσών με περισσότερους χαρακτήρες στο αλφάβητό τους (π.χ. Κινέζικα). Για την αντιμετώπιση αυτού του προβλήματος χρησιμοποιείται το πρότυπο Unicode, το οποίο παρέχει ένα μόνο σύνολο χαρακτήρων με τη δυνατότητα αποθήκευσης πολλών διαφορετικών αλφάβητων. Το πρότυπο Unicode υποστηρίζει διάφορους τρόπους κωδικοποίησης χαρακτήρων οι οποίοι απαιτούν περισσότερα από ένα byte για κάθε χαρακτήρα.

Η πλέον διαδεδομένη έκδοση του προτύπου Unicode είναι η UTF-8. Η έκδοση αυτή χρησιμοποιεί από ένα μέχρι τέσσερα byte για την κωδικοποίηση ενός χαρακτήρα, δίνοντας τη δυνατότητα κωδικοποίησης τεράστιου πλήθους χαρακτήρων ικανό να υπερκαλύψει τα αλφάβητα όλων των γλωσσών του κόσμου! Οι πρώτοι 128 χαρακτήρες των προτύπων ASCII και UTF-8 είναι κοινοί. Στο Κεφάλαιο 10 αναλύονται σε βάθος τα διάφορα πρότυπα κωδικοποίησης.

³ Το ακρωνύμιο **ASCII** προέρχεται από τα αρχικά **American Standard Code for Information Interchange** (Αμερικανικός Πρότυπος Κώδικας για την Ανταλλαγή Πληροφοριών).

ASCII	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
32		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
48	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
128	A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N	Ξ	O	Π
144	P	Σ	T	Υ	Φ	X	Ψ	Ω	α	β	γ	δ	ε	ζ	η	θ
160	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	ς	τ	υ	φ	χ	ψ
176	█	█	█		┌	┐	└	┘	┌	┐	└	┘	┌	┐	└	┘
192	L	└	T	┌	—	┐	└	┘	└	┘	└	┘	└	┘	└	┘
208	└	┐	└	┐	└	┐	└	┐	└	┐	└	┐	└	┐	└	┐
224	ω	ά	έ	ή	ϊ	ί	ό	ύ	ϋ	ώ	Ά	Έ	Ή	Ί	Ό	Υ
240	Ω	±	≥	≤	ï	ÿ	÷	≈	°	·	·	√	π	²	■	

Πίνακας 1.2 Πίνακας κωδικοποίησης του τυπικού και επεκταμένου προτύπου ASCII

Εικόνες

Η ψηφιοποίηση μιας εικόνας επιτυγχάνεται με τη διάσπαση της εικόνας σε πολλές κατακόρυφες και οριζόντιες λωρίδες που σχηματίζουν πολλά μικρά τετράγωνα, τα οποία ονομάζονται pixel (ή *εικονοστοιχεία*). Για κάθε pixel ο Η/Υ αντιστοιχίζει μια σειρά από 1 και 0, ανάλογα με το χρώμα του συγκεκριμένου pixel (Σχήμα 1.3β).

Για μια ασπρόμαυρη εικόνα επαρκεί ένα bit για κάθε pixel (1 για άσπρο και 0 για μαύρο). Όταν όμως μια εικόνα διαθέτει και άλλα χρώματα, για την απεικόνιση κάθε pixel απαιτούνται περισσότερα bit. Ο αριθμός των bit που χρησιμοποιούμε για την κωδικοποίηση ενός pixel αναφέρεται ως *βάθος χρώματος*. Στις έγχρωμες εικόνες το βάθος χρώματος εκφράζεται σε πολλαπλάσια του byte. Έτσι, μπορούμε να πάρουμε βάθος χρώματος 8, 16, 24 και 32 bit, μετατρέποντας κάθε pixel σε 1, 2, 3 ή 4 byte, αντίστοιχα. Το βάθος χρώματος καθορίζει και το πλήθος των διαθέσιμων χρωμάτων για την εικόνα. Έτσι, με βάθος χρώματος 8 bit μπορούμε να έχουμε μόνο 256 διαφορετικά χρώματα, με 16 bit 65.536 διαφορετικά χρώματα, ενώ με 32 bit περισσότερα από τέσσερα δισεκατομμύρια!

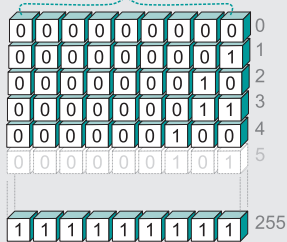
Το πλήθος των pixel στα οποία θα χωριστεί μια εικόνα καθορίζεται από την ανάλυση (resolution) της ψηφιοποίησης. Όσο μεγαλύτερη είναι η ανάλυση, τόσο περισσότερα και μικρότερα είναι τα pixel μιας εικόνας και τόσο καλύτερη είναι η ποιότητα ψηφιοποίησής της. Η ανάλυση καθορίζεται από τα pixel στα οποία χωρίζεται η εικόνα κατακόρυφα και οριζόντια.



Το **bit** είναι η μικρότερη μονάδα πληροφορίας και μπορεί να λάβει την τιμή 1 ή την τιμή 0.

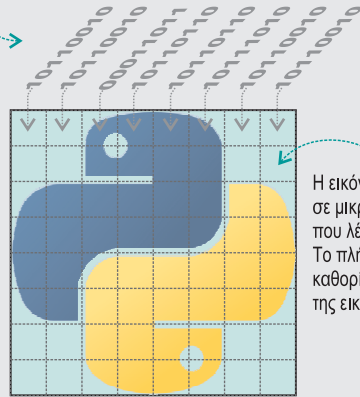
1

Ο συνδυασμός 8 bit λέγεται **byte** και είναι η βασική μονάδα μέτρησης πληροφορίας.



2
Με ένα byte μπορούν να γίνουν **256** συνδυασμοί.

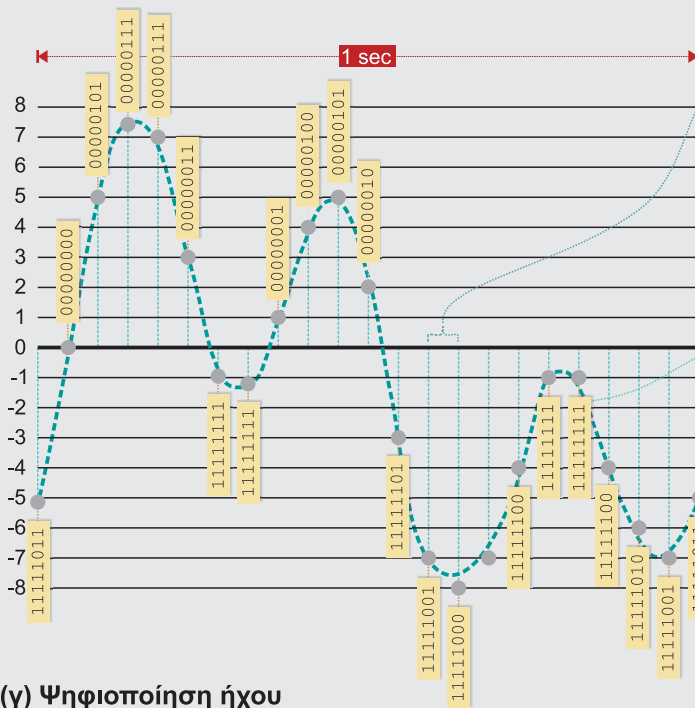
Κάθε ριxel μετατρέπεται σε μια σειρά από 1 και 0, το πλήθος των οποίων καθορίζεται από το βάθος χρώματος που επιλέγουμε.



Η εικόνα χωρίζεται σε μικρά τετραγώνια που λέγονται ριxel. Το πλήθος των ριxel καθορίζει την ανάλυση της εικόνας.

(α) Bit και Byte

(β) Ψηφιοποίηση εικόνας



Γίνεται δειγματοληψία του ήχου ανά συγκεκριμένα χρονικά διαστήματα. Η συχνότητά της μετρείται σε Hz.

Η τιμή που προκύπτει από κάθε δειγματοληψία μετατρέπεται σε σειρές από 1 και 0. Το πλήθος των bit καθορίζει την ποιότητα του ήχου. Μπορεί να είναι 8, 16, 24, 32, ακόμα και 64 bit.

(γ) Ψηφιοποίηση ήχου

Σχήμα 1.3 Bit, Byte και ψηφιοποίηση πληροφορίας

Στο παράδειγμα του Σχήματος 1.3β, η ανάλυση της εικόνας είναι 8x8, επομένως αποτελείται από 64 συνολικά pixel. Το βάθος χρώματος στο συγκεκριμένο παράδειγμα είναι ένα byte, οπότε το συνολικό μέγεθος της εικόνας είναι 64 byte. Η ανάλυση και το βάθος χρώματος καθορίζουν το μέγεθος μιας ψηφιακής εικόνας. Για παράδειγμα, μια εικόνα με ανάλυση 800x600, και βάθος χρώματος 24 bit μετατρέπεται σε 1.440.000 byte ($800*600*3$), δηλαδή σε 1,44 MB. Υπάρχουν τεχνικές συμπίεσης, όπως το δημοφιλές πρωτόκολλο JPEG, με τις οποίες μια εικόνα μπορεί να μετατραπεί σε λιγότερα byte με κάποια απώλεια ποιότητας, η οποία συνήθως δεν γίνεται αντιληπτή.

Βίντεο

Ένα βίντεο αποτελείται από πολλές εικόνες (ή καρτέ) οι οποίες διαδέχονται η μία την άλλη. Συνήθως, σε ένα βίντεο καλής ποιότητας προβάλλονται διαφορετικές εικόνες με ρυθμό 30 καρτέ το δευτερόλεπτο (30 fps – frames per second). Κάθε μια από τις εικόνες του ψηφιοποιείται με τον τρόπο που αναλύθηκε στην προηγούμενη παράγραφο.

Ας πάρουμε για παράδειγμα, ένα βίντεο 30 fps με διαστάσεις 800x600 διάρκειας ενός λεπτού. Το βίντεο αυτό αποτελείται από 1800 (30x60) εικόνες ανάλυσης 800x600. Όπως υπολογίσαμε στην προηγούμενη παράγραφο, κάθε μία τέτοια εικόνα μετατρέπεται σε 1,44 MB, επομένως το συνολικό μέγεθος του βίντεο θα είναι 2.592 MB (1.800x1,44) δηλαδή περίπου 2,6 GB!!! Παρατηρούμε ότι το μέγεθος ενός βίντεο, ακόμα και πολύ μικρής διάρκειας, είναι τεράστιο. Όμως, με χρήση ειδικών τεχνικών συμπίεσης μπορούμε να μειώσουμε κατά πολύ το μέγεθος ενός ψηφιακού βίντεο. Το πιο δημοφιλές από τα πρότυπα συμπίεσης βίντεο είναι το MPEG-4⁴.

Ήχος

Ελπίζω να θυμάστε από τη Φυσική ότι ο ήχος είναι μια κυματομορφή η οποία καθορίζει τόσο την ένταση όσο και τη συχνότητά του. Η ψηφιοποίηση του ήχου βασίζεται στη δειγματοληψία αυτής της κυματομορφής, αρκετές χιλιάδες φορές το δευτερόλεπτο! (Σχήμα 1.3γ). Αυτό πρακτικά σημαίνει ότι ανά τακτά χρονικά διαστήματα μετράμε το εύρος της κυματομορφής, τη συγκεκριμένη χρονική στιγμή. Στη συνέχεια, η μέτρηση αυτή μετατρέπεται σε μια σειρά από 1 και 0.

Ο ρυθμός δειγματοληψίας εκφράζεται σε Hz και αναφέρεται στο πλήθος των δειγματοληψιών ανά δευτερόλεπτο. Στο παράδειγμα του Σχήματος 1.3γ, ο ρυθμός δειγματοληψίας είναι 24 Hz, διότι σε διάστημα ενός δευτερολέπτου έγιναν 24 δειγματοληψίες. Είναι προφανές ότι υψηλότεροι ρυθμοί δειγματοληψίας έχουν ως αποτέλεσμα την πιστότερη

⁴ Η ονομασία MPEG προέρχεται από τα αρχικά των λέξεων Moving Picture Experts Group (Ομάδα Ειδικών Κινούμενης Εικόνας).

αναπαραγωγή της αρχικής κυματομορφής. Κάθε μέτρηση που προκύπτει από τη δειγματοληψία μετατρέπεται σε 8, 16, 24, 32 ακόμα και 64 bit, δηλαδή σε 1, 2, 3, 4 και 8 byte αντίστοιχα. Όσα περισσότερα είναι τα bit, τόσο καλύτερη είναι η ποιότητα του ψηφιοποιημένου ήχου.

Ο συνδυασμός του ρυθμού δειγματοληψίας και του αριθμού bit μετατροπής καθορίζει την τελική ποιότητα του ήχου. Με δειγματοληψία 44.1 KHz (44.100 μετρήσεις το δευτερόλεπτο) και μετατροπή 16 bit, παράγεται ήχος αντίστοιχης ποιότητας με ένα CD ήχου. Με ψηφιοποίηση υψηλής ανάλυσης, στα 96KHz/24bit, παράγεται ήχος ποιότητας αντίστοιχης ενός στούντιο ηχογράφησης.

Σύμφωνα με τα παραπάνω, το μέγεθος ενός ψηφιοποιημένου τραγουδιού διάρκειας τριών λεπτών, δηλαδή 180 δευτερολέπτων, θα είναι περίπου 16 MB σε ποιότητα CD (44.100x180x2), και περίπου 51 MB σε ψηφιοποίηση υψηλής ανάλυσης (96.000x180x3). Και εδώ υπάρχουν πρότυπα συμπίεσης τα οποία μειώνουν σημαντικά το μέγεθος ενός ψηφιακού ήχου, με το πιο δημοφιλές να είναι το πρωτόκολλο MP3⁵.

Λογικές πύλες

Ελπίζω να έγινε κατανοητό από όλα τα προηγούμενα ότι ο H/Y αντιλαμβάνεται μόνο το 1 και το 0, και επομένως οι μοναδικές πράξεις που πραγματοποιεί γίνονται μόνο με το 1 και το 0! Τις πράξεις αυτές τις αναλαμβάνουν ηλεκτρονικά κυκλώματα τα οποία αποτελούνται από **λογικές πύλες**.

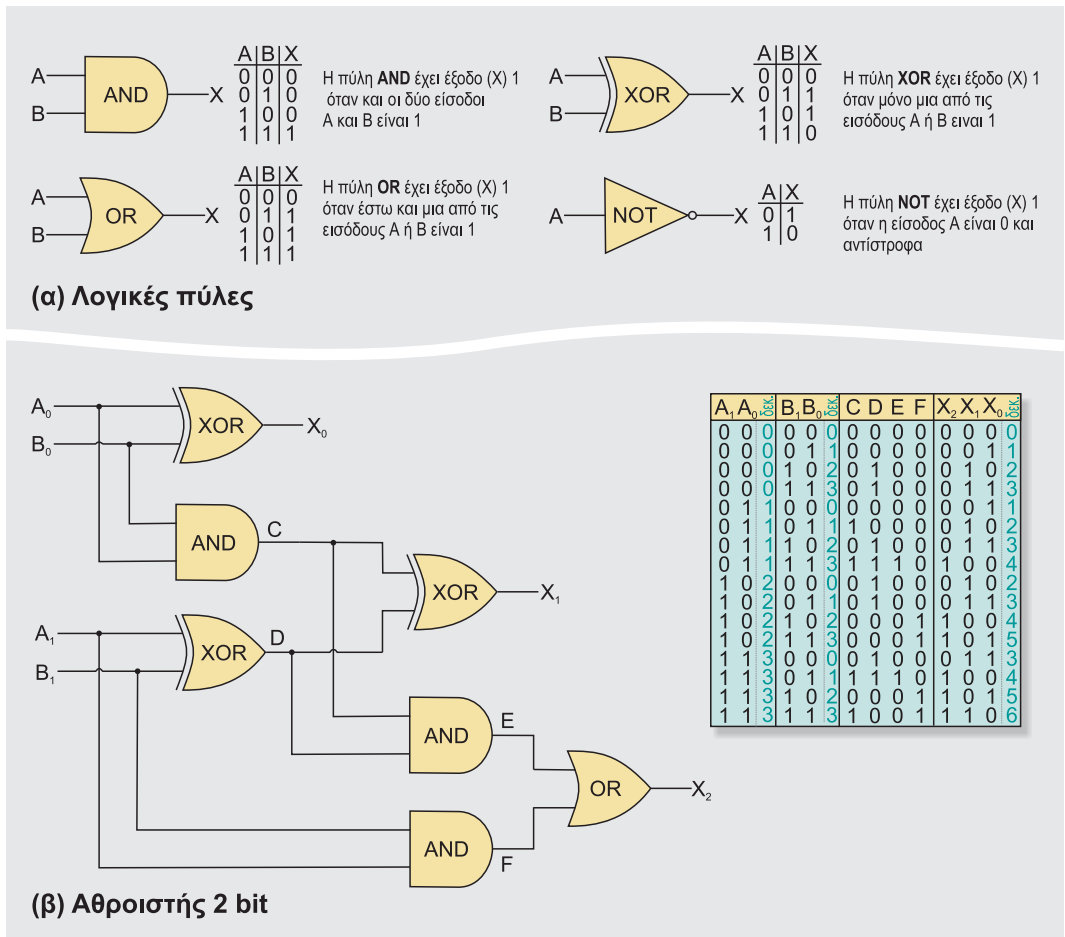
Μια λογική πύλη είναι ένα κύκλωμα το οποίο πραγματοποιεί μια λογική πράξη στις εισόδους του και παράγει μια έξοδο. Οι λογικές πύλες αντιλαμβάνονται το 1 ως ύπαρξη τάσης και το 0 ως απουσία τάσης. Πιο συγκεκριμένα, το λογικό 0 θεωρείται η τάση εκείνη η οποία είναι κάτω από ένα κατώφλι που έχουν ορίσει οι κατασκευαστές της λογικής πύλης (π.χ. 0,5 V) και το λογικό 1 αντιστοιχεί σε τάση η οποία υπερβαίνει κάποια τιμή (π.χ. 5 V). Ένα ολοκληρωμένο κύκλωμα σήμερα μπορεί να περιέχει περισσότερες από 100 εκατομμύρια λογικές πύλες.

Οι βασικές λογικές πύλες είναι τέσσερις: **AND**, **OR**, **XOR** και **NOT**. Οι πράξεις σε επίπεδο δυαδικών αριθμών πραγματοποιούνται με διάφορους συνδυασμούς αυτών των πυλών. Οι πρώτες τρεις πύλες διαθέτουν δύο εισόδους και μία έξοδο, ενώ η πύλη NOT διαθέτει μία μόνο είσοδο και φυσικά μία έξοδο. Στο Σχήμα 1.4α εμφανίζονται οι τέσσερις παραπάνω πύλες καθώς και οι πίνακες αλήθειας για κάθε μία. Οι πίνακες αλήθειας δείχνουν τη συμπεριφορά της εξόδου μιας πύλης σε σχέση με τις τιμές στις εισόδους της.

⁵ Το πρωτόκολλο MP3 σχεδιάστηκε επίσης από την Moving Picture Experts Group (MPEG), αρχικά ως τμήμα του προτύπου MPEG-1 και στη συνέχεια του MPEG-2.

Ο Η/Υ εκτελεί αριθμητικές πράξεις μεταξύ δυαδικών αριθμών χρησιμοποιώντας αρκετά απλούς μέχρι και ιδιαίτερα πολύπλοκους συνδυασμούς αυτών των πυλών. Το Σχήμα 1.4β δείχνει ένα πιο σύνθετο κύκλωμα το οποίο αποτελείται από 7 λογικές πύλες. Το κύκλωμα αυτό είναι ένας αθροιστής 2 bit, δηλαδή προσθέτει δύο αριθμούς που αποτελούνται από δύο δυαδικά ψηφία ο καθένας. Από δεξιά προς αριστερά, τα A_0 και A_1 είναι τα ψηφία του ενός αριθμού και τα B_0 και B_1 του άλλου.

Στον πίνακα αλήθειας του σχήματος φαίνονται όλοι οι δυνατοί συνδυασμοί πράξεων μεταξύ των αριθμών A και B , καθώς και το αποτέλεσμα X . Δίπλα από κάθε δυαδική τιμή των A , B και X , με ανοιχτό πράσινο χρώμα, εμφανίζεται ο αντίστοιχος δεκαδικός αριθμός. Για να γίνει πιο κατανοητός ο αθροιστής, οι ενδιάμεσες τιμές C , D , E και F παρουσιάζονται στα αντίστοιχα σημειωμένα σημεία του κυκλώματος. Μελετήστε αυτό το σχήμα με προσοχή ώστε να κατανοήσετε τη λογική της λειτουργίας τού αθροιστή.



Σχήμα 1.4 Λογικές πύλες

1.6 Μονάδες αποθήκευσης

Όπως ήδη αναφέρθηκε, οι μονάδες αποθήκευσης (περιφερειακή μνήμη) χρησιμοποιούνται για τη μόνιμη αποθήκευση πληροφοριών σε αποθηκευτικά μέσα διαφόρων τεχνολογιών.

Ο σκληρός δίσκος (Hard Disk Drive – HDD) ήταν για πολλά χρόνια το σύνηθες μέσο αποθήκευσης πληροφοριών στους Η/Υ. Ο σκληρός δίσκος αποτελείται από έναν ή περισσότερους μεταλλικούς δίσκους, τοποθετημένους σε κυλινδρική διάταξη, με επιστρωση μαγνητικού υλικού στις επιφάνειές τους. Οι πληροφορίες αποθηκεύονται με τον κατάλληλο μαγνητισμό αυτού του μαγνητικού υλικού. Ο σκληρός δίσκος συνήθως τοποθετείται στο εσωτερικό των Η/Υ, όμως υπάρχουν και εξωτερικοί σκληροί δίσκοι που συνδέονται με τον Η/Υ μέσω κάποιας θύρας επικοινωνίας (π.χ. USB).

Οι σκληροί δίσκοι καταναλώνουν αρκετό ρεύμα, θερμαίνονται αρκετά, κάνουν θόρυβο και είναι αρκετά ευάλωτοι σε απότομες κινήσεις διότι περιέχουν κινητά μηχανικά μέρη.

Οι σκληροί δίσκοι σύντομα θα αντικατασταθούν πλήρως από τους δίσκους στερεάς κατάστασης (Solid State Disk – SSD), οι οποίοι χρησιμοποιούν ολοκληρωμένα κυκλώματα μνήμης για την αποθήκευση πληροφοριών και όχι κάποιο μαγνητικό μέσο. Οι δίσκοι SSD έχουν μικρότερο μέγεθος, είναι περισσότερο αξιόπιστοι, αθόρυβοι, και ταχύτεροι, καταναλώνουν ελάχιστο ρεύμα, και δεν θερμαίνονται πολύ! Το μοναδικό τους μειονέκτημα είναι η αρκετά υψηλή τιμή τους σε σχέση με τους μηχανικούς σκληρούς δίσκους.

Για την καταγραφή και την ανάγνωση των πληροφοριών σε οπτικούς δίσκους χρησιμοποιείται μια ακτίνα λέιζερ. Υπάρχουν τρεις κατηγορίες οπτικών δίσκων: τα CD, τα DVD και οι δίσκοι Blu-ray (BD). Χρησιμοποιούνται κυρίως για τη μεταφορά και διάθεση πληροφοριών, καθώς και για την τήρηση αντιγράφων ασφαλείας.

Ένα ακόμα μαγνητικό μέσο είναι η μαγνητική ταινία, την οποία όμως είναι μάλλον απίθανο να συναντήσετε ποτέ, αφού χρησιμοποιείται αποκλειστικά για τη λήψη αντιγράφων ασφαλείας δεδομένων που βρίσκονται σε σκληρούς δίσκους μεγάλων κεντρικών συστημάτων.

Τέλος, οι φορητές συσκευές αποθήκευσης, όπως οι μονάδες USB (USB-sticks) και οι κάρτες μνήμης (SD, MD), που χρησιμοποιούνται στα κινητά τηλέφωνα και σε άλλες ψηφιακές συσκευές, χρησιμοποιούν την τεχνολογία στερεάς κατάστασης που αναφέρθηκε πιο πάνω. Συνδέονται στις κατάλληλες θύρες ενός Η/Υ και είναι ένας εξαιρετικά άμεσος και εύχρηστος τρόπος για την αποθήκευση και μεταφορά πληροφοριών.

Τι είναι ένα αρχείο;

Σε μια μονάδα αποθήκευσης μπορούμε να αποθηκεύσουμε κάθε είδους δεδομένα (εικόνα, κείμενο, βίντεο, κ.λπ.), αρκεί να υπάρχει ο απαιτούμενος ελεύθερος χώρος.

Η γλώσσα Python σε βάθος

Καλύπτει την έκδοση 3 της Python

Το βιβλίο ξεκινά με μια αναλυτική εισαγωγή στην επιστήμη των Η/Υ και τον προγραμματισμό, εξοικειώνοντας τον αναγνώστη με τις βασικές έννοιες και την αλγοριθμική σκέψη. Στη συνέχεια, αναλύει όλα τα χαρακτηριστικά της Python, με έναν ιδιαίτερα εποπτικό και επεξηγηματικό τρόπο. Το βιβλίο δεν προϋποθέτει γνώσεις προγραμματισμού, και βοηθά τον αναγνώστη να εξοικειωθεί εύκολα και γρήγορα με τη φιλοσοφία του προγραμματισμού και να κατανοήσει πλήρως τα στοιχεία της γλώσσας. Ο πεπειραμένος προγραμματιστής θα βρει μια εμπειριστατωμένη ανάλυση των μηχανισμών της γλώσσας, πλήρη κάλυψη των ιδιαίτερων χαρακτηριστικών της, καθώς και υλοποιήσεις εξειδικευμένων αλγορίθμων και τεχνικών προγραμματισμού.

Στο βιβλίο εξετάζονται διεξοδικά τα παρακάτω θέματα:

- Διαχείριση μνήμης και οντότητες της Python
- Βασικοί τύποι δεδομένων, παραστάσεις, εντολές ελέγχου και επανάληψης
- Συναρτήσεις, γεννήτριες και επαναλήπτες
- Λίστες, πλειάδες, σύνολα λεξικά, bytes και bytearray
- Χειρισμός αρχείων
- Αναδρομικές συναρτήσεις, αλγόριθμοι αναζήτησης και ταξινόμησης
- Ουρές και στοίβες, και ο τρόπος υλοποίησής τους στην Python
- Αντικειμενοστρεφής προγραμματισμός, κλάσεις και αντικείμενα
- Πολυμορφισμός, υπερφόρτωση τελεστών και κληρονομικότητα
- Διαχείριση εξαιρέσεων
- Το IDLE, το ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών της Python



Ο Νίκος Χατζηγιαννάκης είναι πτυχιούχος του Τμήματος Ηλεκτρολόγων Μηχανικών Η/Υ της Πολυτεχνικής Σχολής του Πανεπιστημίου Πατρών και κάτοχος Master στην

Επιστήμη των Υπολογιστών από το Πανεπιστήμιο Bath της Αγγλίας. Από το 1987 δραστηριοποιείται στην εκπαίδευση, με χιλιάδες ώρες διδασκαλίας σε θέματα προγραμματισμού και χειρισμού Η/Υ. Παράλληλα, ως ειδικός επιστήμων σε θέματα ανάλυσης και σχεδίασης συστημάτων Η/Υ της Γ.Γ. Αιγαίου και Νησιωτικής Πολιτικής, έχει αναλάβει τον σχεδιασμό και την ανάπτυξη πρωτοποριακών πληροφοριακών συστημάτων για τον Δημόσιο Τομέα και είναι επιστημονικός υπεύθυνος έργων που αφορούν την ηλεκτρονική διακυβέρνηση, την εξυπηρέτηση του πολίτη, και τον εκσυγχρονισμό της Δημόσιας Διοίκησης. Το 2009 ξεκίνησε να διδάσκει διαδικασιακό και αντικειμενοστρεφή προγραμματισμό στο τμήμα Πολιτισμικής Τεχνολογίας και Επικοινωνίας του Πανεπιστημίου Αιγαίου.

Έχει επίσης γράψει τα βιβλία *Η γλώσσα C σε βάθος* και *Η γλώσσα C++ σε βάθος*, τα οποία κυκλοφορούν από τις Εκδόσεις Κλειδάριθμος.

Επικοινωνία: nikos@bytes.gr

- Κάθε κεφάλαιο περιλαμβάνει επίσης λυμένα παραδείγματα, μια σύντομη ανασκόπηση, τα συχνότερα λάθη που γίνονται, καθώς και ασκήσεις διαφορετικών βαθμών δυσκολίας. Η δομή αυτή καθιστά το βιβλίο ιδανικό για εκπαιδευτική χρήση.
- Το συνοδευτικό υλικό του βιβλίου περιέχει τον κώδικα των προγραμμάτων του βιβλίου, τις απαντήσεις όλων των ασκήσεων, καθώς και τον κώδικα των λύσεών τους.
- Με αποσπώμενη κάρτα αναφοράς.
- Δυνατότητα για διαδικτυακά (online) μαθήματα στο <http://python.bytes.gr/elessons>
- Ιστότοπος με χρήσιμο εκπαιδευτικό υλικό στη διεύθυνση <http://python.bytes.gr>



ΕΚΔΟΣΕΙΣ
ΚΛΕΙΔΑΡΙΘΜΟΣ

Κεντρική διεύθυνση: Μάρνη 8, Πολυτεχνείο, 10433, ΑΘΗΝΑ, Τηλ: 210-3300104, sales@klidarithmos.gr
Έδρα: Στουρνάρη 27B, 10682, ΑΘΗΝΑ, Τηλ: 210-3832044
info@klidarithmos.gr • www.klidarithmos.gr • www.facebook.com/klidarithmos.gr

ISBN 978-960-645-471-4



9 789606 454714