

ΚΕΦΑΛΑΙΟ 3

1. Τι είναι τα δεδομένα και που βρίσκονται. (σελ 29)
2. Τι ονομάζουμε «τύπο δεδομένων». (σελ 29)
3. Ποιοι τύποι δεδομένων υπάρχουν στην Python. (σελ 29)
4. Τι περιλαμβάνει ο αριθμητικός τύπος δεδομένων. Τι τιμές μπορεί να λάβει. (σελ 29)
5. Τι τιμές μπορεί να λάβει ο λογικός τύπος δεδομένων. Γιατί χρησιμοποιείται. (σελ 30)
6. Τι είναι οι συμβολοσειρές (αλφαριθμητικά). (σελ 30 και σελ 127)
7. Με ποια εντολή μπορούμε να ελέγξουμε τον τύπο δεδομένων. Να γράψετε τη σύνταξη και τη λειτουργία της. (σελ 30 και φυλλάδιο)
8. Τι είναι οι τελεστές. Ποιες κατηγορίες τελεστών υπάρχουν. (σελ 30-32)
9. Ποιοι είναι οι αριθμητικοί τελεστές. (σελ 32)
10. Ποια είναι η προτεραιότητα (ιεραρχία) μεταξύ αριθμητικών τελεστών. (σελ 31)
11. Πώς μπορούμε να αλλάξουμε την ιεραρχία των τελεστών. (σελ 31)
12. Τι είναι οι σχεσιακοί τελεστές. Ποιοι σχεσιακοί τελεστές υπάρχουν στην Python. (σελ 31)
13. Πως γίνεται η εφαρμογή των σχεσιακών τελεστών σε συμβολοσειρές. (Φυλλάδιο και σελ 128)
14. Ποιοι είναι οι λογικοί τελεστές και ποια πράξη μπορούμε να κάνουμε με τον καθένα. (σελ 32)
15. Ποιο μπορεί να είναι το αποτέλεσμα μιας λογικής πράξης. (σελ 32)
16. Ποια είναι η προτεραιότητα μεταξύ λογικών τελεστών. (σελ 32)
17. Να περιγράψετε τη λειτουργία του λογικού τελεστή and. (Φυλλάδιο)
18. Να περιγράψετε τη λειτουργία του λογικού τελεστή or. (Φυλλάδιο)
19. Να περιγράψετε τη λειτουργία του λογικού τελεστή not. (Φυλλάδιο)
20. Ποια είναι η προτεραιότητα μεταξύ όλων των πράξεων/τελεστών στην Python. (Φυλλάδιο)
21. Τι είναι εκφράσεις. (Φυλλάδιο)
22. Ποιες κατηγορίες εκφράσεων υπάρχουν. (Φυλλάδιο)
23. Τι είναι μια μεταβλητή. (SOS - 2 απαντήσεις: Φυλλάδιο και σελ 35)
24. Τι χρειάζεται για να χρησιμοποιήσουμε μια μεταβλητή. (σελ 33)
25. Ποιους κανόνες πρέπει να ακολουθούμε κατά την ονομασία μιας μεταβλητής. (σελ 33-34)
26. Τι είναι ο τελεστής εκχώρησης τιμής (=). (σελ 34)
27. Πώς γίνεται η εκχώρηση τιμής σε μια μεταβλητή. Εξηγήστε με ένα παράδειγμα. (σελ 32)
28. Πώς αναπαρίστανται τα δεδομένα σε ένα πρόγραμμα Python. (σελ 34-35)
29. Τι ονομάζουμε «συλλογή σκουπιδιών» στην Python. (σελ 35)
30. Ποιες είναι οι ενσωματωμένες συναρτήσεις της Python και ποια η λειτουργία τους. (σελ 39)
31. Τι είναι οι εξωτερικές βιβλιοθήκες. Πως χρησιμοποιούνται στην Python. (σελ 40)
32. Ποιες καλές πρακτικές πρέπει να τηρούμε κατά τη συγγραφή ενός προγράμματος. (σελ 41)
33. Τι είναι τα σχόλια σε ένα πρόγραμμα Python. (σελ 38)
34. Τι είναι ένας Τύπος Δεδομένων. Ποιες μεγάλες κατηγορίες υπάρχουν. Πώς διακρίνονται ανάλογα με τη σύσταση των μερών τους; (σελ 42)
35. Τι είναι οι απλοί τύποι δεδομένων; Ποιοι απλοί τύποι δεδομένων υπάρχουν στις γλώσσες προγραμματισμού. (σελ 43)

36. Τι είναι οι σύνθετοι τύποι δεδομένων; Ποιοι σύνθετοι τύποι δεδομένων υπάρχουν στις γλώσσες προγραμματισμού. (σελ 43)
37. Τι ονομάζουμε «αφηρημένο τύπο δεδομένων». (σελ 44 ή φυλλάδιο)
38. Να περιγράψετε τη λειτουργία της εντολής εξόδου `print`. (σελ 36 και φυλλάδιο)
39. Να περιγράψετε τη λειτουργία της εντολής εισόδου `input` και `raw_input`. (σελ 37-38 και φυλλάδιο)

ΚΕΦΑΛΑΙΟ 4

40. Τι ονομάζουμε αλγοριθμική δομή. (Φυλλάδιο)
41. Να αναφέρετε ποιες βασικές αλγοριθμικές δομές χρησιμοποιούνται για την ανάπτυξη ενός προγράμματος. (Φυλλάδιο)
42. Να περιγράψετε την αλγοριθμική δομή ακολουθίας. (σελ 45)
43. Τι είδους προβλήματα μπορούμε να λύσουμε με τη δομή ακολουθίας. (σελ 45)
44. Πότε χρησιμοποιείται η δομή επιλογής. (σελ 46)
45. Με ποια εντολή υλοποιείται η δομή επιλογής στην Python. (σελ 46 ή φυλλάδιο)
46. Πότε χρησιμοποιείται η απλή δομή επιλογής. (σελ 46)
47. Να γράψετε τον τρόπο χρήσης της εντολής `if` στην Python και να περιγράψετε τη λειτουργία της. (σελ 46)
48. Πότε χρησιμοποιείται η σύνθετη δομή επιλογής. (σελ 47)
49. Να γράψετε τον τρόπο χρήσης της συνθέτης δομής επιλογής στην Python και να περιγράψετε τη λειτουργία της. (σελ 47-48)
50. Πότε χρησιμοποιείται η πολλαπλή επιλογή. (σελ 49)
51. Με ποια εντολή υλοποιείται η πολλαπλή επιλογή στην Python. Να περιγράψετε τον τρόπο χρήσης της και τη λειτουργία της. (σελ 49)
52. Πότε χρησιμοποιείται η εμφωλευμένη δομή επιλογής. (Φυλλάδιο)
53. Με ποιον τρόπο υλοποιείται η εμφωλευμένη επιλογή στην Python. (Φυλλάδιο)
54. Πότε χρησιμοποιείται η δομή επανάληψης. (σελ 51)
55. Ποιοι τύποι επανάληψης υπάρχουν. (σελ 51)
56. Ποιες εντολές επανάληψης χρησιμοποιεί Python. (σελ 51)
57. Να εξηγήσετε τη λειτουργία της συνάρτησης `range()` στην Python. (σελ 52 και σελ 137)
58. Πότε χρησιμοποιείται η εντολή `for`. Να γράψετε τον τρόπο χρήσης της εντολής `for` στην Python και να περιγράψετε τη λειτουργία της. (Φυλλάδιο)
59. Πότε χρησιμοποιείται η εντολή `while`. (σελ 53)
60. Να γράψετε τον τρόπο χρήσης της εντολής `while` στην Python και να περιγράψετε τη λειτουργία της. (σελ 53)
61. Ποιες είναι οι προϋποθέσεις για τη σωστή λειτουργία της εντολής `while`. (σελ 51, Σημείωση 1 και 2)
62. Πώς μπορούμε να ελέγξουμε την εγκυρότητα των δεδομένων εισόδου από το χρήστη. (σελ 54)
63. Τι είναι η εμφωλευμένη δομή επανάληψης. (Φυλλάδιο)
64. Τι είναι οι συναρτήσεις. (σελ 57 και σελ 106)
65. Τι ονομάζεται κλήση της συνάρτησης. (σελ 57 και σελ 107-108)
66. Πώς μπορούμε να ορίσουμε μία δική μας συνάρτηση. (σελ 57 και σελ 107)
67. Τι είναι οι παράμετροι μιας συνάρτησης. (σελ 58 και σελ 110)
68. Πώς χρησιμοποιούνται οι παράμετροι μιας συνάρτησης. (σελ 58 και σελ 110 - 111)

- 69. Τι ονομάζονται ορίσματα. (σελ 58 και σελ 110 - 111)
- 70. Ποια η διαφορά ανάμεσα στις παραμέτρους και τα ορίσματα. (σελ 58 και σελ 110 - 111)
- 71. Τι κάνει η εντολή `import random`. (σελ 54)

ΚΕΦΑΛΑΙΟ 7

- 72. Τι ονομάζεται τμηματικός προγραμματισμός. (σελ 105)
- 73. Ποια ανάγκη οδήγησε στην ανάπτυξη του τμηματικού προγραμματισμού. (σελ 104 - 105)
- 74. Τι είναι ένα υποπρόγραμμα. (σελ 105 και φυλλάδιο)
- 75. Τη βοήθειά μας προσφέρει ο τμηματικός προγραμματισμός. (σελ 105)
- 76. Ποια είναι τα χαρακτηριστικά ενός υποπρογράμματος. (σελ 105)
- 77. Ποιες καλές πρακτικές πρέπει να εφαρμόζουμε κατά την υλοποίηση του τμηματικού προγραμματισμού. (σελ 106)
- 78. Πώς υλοποιείται ο τμηματικός προγραμματισμός στην Python. (σελ 106)
- 79. Πόσοι τύποι υποπρογραμμάτων υπάρχουν στην Python. (σελ 106)
- 80. Να γράψετε και να εξηγήσετε πώς ορίζεται μία συνάρτηση στην Python. (σελ 107)
- 81. Από ποια σημεία μπορεί να καλείται μια συνάρτηση. (σελ 107)
- 82. Να γράψετε τον τρόπο χρήσης της εντολής `return` στην Python και να περιγράψετε τη λειτουργία της. (Φυλλάδιο)
- 83. Πώς καλείται μία συνάρτηση στην Python. (σελ 108)
- 84. Να εξηγήσετε το μηχανισμό κλήσης μιας συνάρτησης από το κύριο πρόγραμμα μέσω ενός παραδείγματος. (σελ 108)
- 85. Τι ονομάζεται επιστρεφόμενη τιμή. (σελ 108)
- 86. Με ποιους τρόπους μπορούμε να ακολουθήσουμε τη ροή εκτέλεσης ενός προγράμματος. (σελ 108)
- 87. Ποιες κατηγορίες συναρτήσεων υπάρχουν. (σελ 109 και σελ 110)
- 88. Τι είναι μία κενή συνάρτηση. (σελ 110)
- 89. Τι ονομάζεται παράμετρος ενός υποπρογράμματος. (σελ 110)
- 90. Ποια είναι η διαφορά μεταξύ μεταβλητής και παραμέτρου. (σελ 110)
- 91. Τι σημαίνει ότι το πέρασμα παραμέτρων είναι κατά τιμή. (σελ 110-112)
- 92. Τι σημαίνει ότι το πέρασμα παραμέτρων είναι με αναφορά. (σελ 112)
- 93. Τι ονομάζεται εμβέλεια μεταβλητών. (σελ 113)
- 94. Πώς καθορίζεται η εμβέλεια μιας μεταβλητής. (σελ 113)
- 95. Ποια είδη εμβέλειας μεταβλητών υπάρχουν. (σελ 113)
- 96. Πότε οι μεταβλητές έχουν απεριόριστη εμβέλεια. Ποιο μειονέκτημα έχει η απεριόριστη εμβέλεια. (σελ 113)
- 97. Πότε οι μεταβλητές έχουν περιορισμένη εμβέλεια. Τι απαιτεί η περιορισμένη εμβέλεια. (σελ 113)
- 98. Ποιες μεταβλητές ονομάζονται τοπικές. (σελ 113 - 114)
- 99. Ποια είναι η σχέση μεταξύ τοπικών μεταβλητών και συνάρτησης. (σελ 113 - 115)
- 100. Ποιες μεταβλητές ονομάζονται καθολικές. (σελ 113 - 115)
- 101. Πώς ορίζω μία μεταβλητή ως καθολική. (σελ 115)
- 102. Πώς μπορούμε να αλλάξουμε την τιμή μιας καθολικής μεταβλητής μέσα σε μία συνάρτηση. (σελ 115 - 116)
- 103. Ποια είναι η χρήση της εντολής `global` στην Python. (σελ 115 - 116)

104. Ποιες είναι οι διαφορές μεταξύ τοπικών και Καθολικών μεταβλητών στην Python. (σελ 113 - 116)

ΚΕΦΑΛΑΙΟ 8

105. Τι είναι μία δομή δεδομένων. (σελ 126)
106. Γιατί είναι σημαντική η δομή δεδομένων. (Φυλλάδιο)
107. Ποιες κατηγορίες δομών δεδομένων υπάρχουν. (Φυλλάδιο)
108. Ποια είναι τα χαρακτηριστικά των μη μεταβαλλόμενων (ή στατικών – immutable) δομών δεδομένων. (σελ 126 και φυλλάδιο)
109. Ποια είναι τα χαρακτηριστικά των δυναμικών (mutable) δομών δεδομένων. (σελ 130 και φυλλάδιο)
110. Ποιες στατικές δομές δεδομένων υπάρχουν στην Python. (σελ 127 και φυλλάδιο)
111. Ποιες δυναμικές δομές δεδομένων υπάρχουν στην Python. (Φυλλάδιο)
112. Τι είναι οι συμβολοσειρές στην Python. (σελ 127)
113. Γιατί οι συμβολοσειρές ανήκουν στις στατικές δομές δεδομένων. (σελ 127 και φυλλάδιο)
114. Πώς δημιουργείται μία συμβολοσειρά. (σελ 127)
115. Πώς αποτυπώνεται μία συμβολοσειρά. (σελ 127 και φυλλάδιο)
116. Ποιος είναι ο τύπος δεδομένων μιας συμβολοσειράς στην Python. (σελ 127)
117. Ποια η λειτουργία της συνάρτησης len() στην Python. (σελ 127)
118. Ποιο είναι το αποτέλεσμα της εφαρμογής του τελεστή της πρόσθεσης (+) σε συμβολοσειρές. (σελ 127)
119. Ποια είναι η λειτουργία του υπαρξιακού τελεστή in. (σελ 128)
120. Ποια είναι η χρήση του υπαρξιακού τελεστή in στις συμβολοσειρές. (σελ 128-129)
121. Ποια είναι η χρήση του τελεστή διαμέρισης (:) στις συμβολοσειρές. (σελ 138)
122. Πώς γίνεται η σάρωση των χαρακτήρων μιας συμβολοσειράς. (σελ 128)
123. Τι είναι μία λίστα στην Python. (σελ 130)
124. Γιατί η λίστα είναι μία δυναμική δομή δεδομένων. (σελ 130 και φυλλάδιο)
125. Πώς δημιουργείται μία λίστα. (σελ 130)
126. Πώς αποτυπώνεται μία λίστα. (σελ 130 και φυλλάδιο)
127. Πώς γίνεται η προσπέλαση ενός στοιχείου μιας λίστας. (σελ 130 και 133-134)
128. Τι είναι η άδεια λίστα. (Φυλλάδιο)
129. Πώς προσθέτουμε ένα στοιχείο στο τέλος μιας λίστας. (σελ 131)
130. πώς προσθέτουμε ένα στοιχείο στην αρχή μιας λίστας. (σελ 131)
131. Ποιο είναι το αποτέλεσμα της εφαρμογής του τελεστή της πρόσθεσης (+) σε λίστες. (σελ 132)
132. Ποια είναι η χρήση του υπαρξιακού τελεστή in στις λίστες. (σελ 132)
133. Ποια είναι η χρήση του not in στις λίστες. (σελ 132)
134. Πώς εφαρμόζεται η συνάρτηση len() σε μία λίστα. (σελ 132)
135. Πώς λειτουργεί η συνάρτηση list(). Σε ποια άλλα είδη δομών μπορεί να χρησιμοποιηθεί. (σελ 132)
136. Ποια είναι η χρήση του τελεστή διαμέρισης (:) στις λίστες. (σελ 138)
137. Γιατί ο τελεστής διαμέρισης (:) είναι πολύ σημαντικός στην επεξεργασία λιστών. (σελ 139)
138. Ποια είναι η λειτουργία της μεθόδου append() στην Python. (σελ 133)
139. Ποια είναι η λειτουργία της μεθόδου insert() στην Python. (σελ 133)

140. Ποια είναι η λειτουργία της μεθόδου pop() στην Python. (σελ 133)
141. Πώς γίνεται η διάσχιση των στοιχείων μιας λίστας. (σελ 133-134)
142. Πώς βρίσκουμε τον μέσο όρο των στοιχείων μιας λίστας. (σελ 134)
143. Ποιο είναι το πλεονέκτημα του αλγορίθμου συγχώνευσης διατεταγμένων λιστών. (σελ 136)

ΚΕΦΑΛΑΙΟ 5

144. Ποιο μειονέκτημα παρουσιάζει ο αλγόριθμος της σειριακής αναζήτησης. (σελ 63)
145. Πότε μπορεί να χρησιμοποιηθεί η δυαδική αναζήτηση. Γιατί ονομάζεται έτσι. (σελ 68 και 69)
146. Τι ονομάζεται πολυμορφισμός στην Python. (σελ 72)
147. Ποια είναι τα χαρακτηριστικά της δυαδικής αναζήτησης. (σελ 72)
148. Ποιο είναι το χαρακτηριστικό του αλγόριθμου ταξινόμησης με επιλογή. (σελ 72)
149. Με ποια εντολή μπορούμε να αντιμετωπίσουμε το περιεχόμενο των μεταβλητών a και b. (σελ 74)
150. Να γράψετε ισοδύναμο τμήμα εντολών με το $a, b = b, a$. (σελ 74)
151. Που βασίζεται ο αλγόριθμος ταξινόμησης ευθείας ανταλλαγής. (σελ 78)
152. Πόσο γρήγορος θεωρείται ο αλγόριθμος ευθείας ανταλλαγής και ποιο είναι το βασικό του πλεονέκτημα. (σελ 78)
153. Να αναπτύξετε τη βελτιωμένη έκδοση του αλγόριθμου ταξινόμησης ευθείας ανταλλαγής η οποία τερματίζει, όταν διαπιστώσει ότι η λίστα είναι ταξινομημένη, ώστε να αποφεύγονται οι περιττές συγκρίσεις. (Φυλλάδιο).