80 exercices on Python and 20 on Django to prepare

Python Tutorial:

1. Write a program that prints the numbers from 1 to 100.
2. Write a program that prints the numbers from 100 to 1.
3. Write a program that prints the even numbers from 2 to 100.
4. Write a program that prints the odd numbers from 1 to 100.
5. Write a program that prints the multiples of 5 from 5 to 100.
6. Write a program that prints the square of the numbers from 1 to 10.
7. Write a program that calculates the sum of the numbers from 1 to 100.
8. Write a program that calculates the average of the numbers from 1 to 100.
9. Write a program that calculates the factorial of a number.
10. Write a program that calculates the Fibonacci sequence.
11. Write a program that checks if a number is prime.
12. Write a program that generates a random number between 1 and 100.
13. Write a program that generates a list of random numbers between 1 and 100.
14. Write a program that sorts a list of numbers in ascending order.
15. Write a program that sorts a list of numbers in descending order.
16. Write a program that finds the maximum number in a list.
17. Write a program that finds the minimum number in a list.
18. Write a program that calculates the mean of a list of numbers.
19. Write a program that calculates the median of a list of numbers.
20. Write a program that calculates the mode of a list of numbers.
21. Write a program that converts a number from decimal to binary.
22. Write a program that converts a number from binary to decimal.
23. Write a program that converts a number from decimal to hexadecimal.
24. Write a program that converts a number from hexadecimal to decimal.
25. Write a program that converts a number from decimal to octal.
26. Write a program that converts a number from octal to decimal.
27. Write a program that reverses a string.
28. Write a program that finds the length of a string.
29. Write a program that finds the number of vowels in a string.
30. Write a program that finds the number of consonants in a string.
31. Write a program that finds the number of words in a string.
32. Write a program that finds the number of characters in a string.
33. Write a program that counts the frequency of each word in a string.
34. Write a program that capitalizes the first letter of each word in a string.
35. Write a program that converts a string to uppercase.
36. Write a program that converts a string to lowercase.
37. Write a program that removes the spaces from a string.

80 exercices on Python and 20 on Django to prepare

38. Write a program that checks if a string is a palindrome.
39. Write a program that checks if two strings are anagrams.
40. Write a program that finds the longest word in a string.
41. Write a program that finds the shortest word in a string.
42. Write a program that finds the most common word in a string.
43. Write a program that finds the least common word in a string.
44. Write a program that removes duplicates from a list.
45. Write a program that merges two lists.
46. Write a program that finds the intersection of two lists.
47. Write a program that finds the union of two lists.
48. Write a program that finds the difference between two lists.

49. Write a program that calculates the mean, median and mode of a list of numbers
50. Write a program that calculates the standard deviation of a list of numbers

51. Write a program that calculates the variance of a list of numbers.
52. Write a program that calculates the covariance of two lists of numbers.
53. Write a program that calculates the correlation coefficient of two lists of numbers.
54. Write a program that calculates the percentile of a list of numbers.
55. Write a program that calculates the Z-score of a list of numbers.
56. Write a program that calculates the cumulative sum of a list of numbers.
57. Write a program that calculates the cumulative product of a list of numbers.
58. Write a program that calculates the moving average of a list of numbers.
59. Write a program that calculates the running total of a list of numbers.
60. Write a program that implements the bubble sort algorithm.
61. Write a program that implements the selection sort algorithm.
62. Write a program that implements the insertion sort algorithm.
63. Write a program that implements the quick sort algorithm.
64. Write a program that implements the merge sort algorithm.
65. Write a program that calculates the Levenshtein distance between two strings.
66. Write a program that calculates the Jaccard similarity between two sets.

80 exercices on Python and 20 on Django to prepare

67. Write a program that calculates the Cosine similarity between two lists of numbers.
68. Write a program that implements the K-Nearest Neighbors algorithm.
69. Write a program that implements the K-Means clustering algorithm.
70. Write a program that implements the Principal Component Analysis (PCA) algorithm.
71. Write a program that implements the linear regression algorithm.
72. Write a program that implements the logistic regression algorithm.
73. Write a program that implements the decision tree algorithm.
74. Write a program that implements the random forest algorithm.
75. Write a program that implements the gradient boosting algorithm.
76. Write a program that implements the neural network algorithm.
77. Write a program that implements the deep learning algorithm.
78. Write a program that calculates the area of a circle.
79. Write a program that calculates the volume of a sphere.
80. Write a program that calculates the surface area of a sphere.

Django Exercises : to do after studying at least the introduction (not yet)

1. Create a Django project with a single app that displays a list of books and their authors.
2. Add the ability to create, edit, and delete books and authors.
3. Add the ability to upload and display cover images for each book.

80 exercices on Python and 20 on Django to prepare

4. Implement user authentication and authorization, allowing only logged-in users to create, edit, and delete books and authors.
5. Add a search feature that allows users to search for books by title, author, or ISBN.
6. Add pagination to the list of books to display a limited number of books per page.
7. Implement a rating system for books, allowing users to rate books on a scale of 1 to 5.
8. Add a form for users to leave reviews for books, including a text field for the review and a rating.
9. Implement a recommendation system that suggests books to users based on their ratings and reviews.
10. Add the ability to filter books by category (e.g. fiction, non-fiction, mystery, romance, etc.).
11. Add the ability to sort books by various criteria, such as average rating, number of reviews, publication date, etc.
12. Implement a "wishlist" feature that allows users to save books they want to read later.
13. Add the ability to export book data as a CSV or Excel file.
14. Integrate with an external API to retrieve book data, such as cover images, descriptions, and prices.
15. Add the ability to purchase books through a shopping cart system.
16. Implement a payment gateway integration, such as PayPal or Stripe, to process payment transactions.
17. Add the ability for users to rate and review authors.
18. Add the ability for users to follow and receive updates from their favorite authors.
19. Add a blog feature to the app, allowing authors to post articles and users to comment on them.
20. Implement a notification system to notify users of new book releases from their favorite authors and other important updates.