



BOSSCODER
ACADEMY

—MASTER—

PYTHON

BASIC → INTERMEDIATE



In Just 15 Days



Disclaimer

Everyone learns uniquely.

What matters is developing the problem solving ability to solve new problems.

This Doc will help you with the same.

INTRODUCTION AND SETUP

Aim

Get acquainted with Python and set up the development environment.

Resources

- Python Official Documentation:
<https://www.python.org/doc/>
- Codecademy Python Course:
<https://www.codecademy.com/learn/learn-python-3>

Example Questions

Q 1: Write a Python program to print "Hello, World!"

Q 2: Calculate the sum of two numbers entered by the user.

Q 3: Convert temperature from Celsius to Fahrenheit.



Practice Questions

1. Write a Python program to calculate the area of a rectangle given its length and width.
2. Create a program that takes a user's name and age as input and prints a greeting message.
3. Write a program to check if a number is even or odd.
4. Given a list of numbers, find the maximum and minimum values.
5. Create a Python function to check if a given string is a palindrome.
6. Calculate the compound interest for a given principal amount, interest rate, and time period.
7. Write a program that converts a given number of days into years, weeks, and days.
8. Given a list of integers, find the sum of all positive numbers.
9. Create a program that takes a sentence as input and counts the number of words in it.
10. Implement a program that swaps the values of two variables.

VARIABLES AND DATA TYPES

Aim

Understand variables and different data types in Python.

Resources

- W3Schools Python Variables:
[https://www.w3schools.com/python/
python_variables.asp](https://www.w3schools.com/python/python_variables.asp)
- Real Python Data Types:
<https://realpython.com/python-data-types/>

Example Questions

Q 1: Create variables for storing a person's name, age, and average test score.

Q 2: Concatenate two strings and print the result.

Q 3: Create a list of fruits and access elements using indexing.



Practice Questions

1. Given a list of numbers, find the sum and average.
2. Create a program that takes a temperature in Celsius and converts it to Kelvin.
3. Implement a program that checks if a given string is a palindrome.
4. Create a function to reverse a given string.
5. Given a list of names, concatenate them into a single string separated by spaces.
6. Write a Python program to check if a given string is a pangram (contains all letters of the alphabet).
7. Calculate the area and circumference of a circle given its radius.
8. Implement a program that converts a given number of minutes into hours and minutes.
9. Create a function to count the number of vowels in a given string.
10. Write a program to check if a number is prime.

CONTROL FLOW AND LOOPS

Aim

Learn about conditional statements and loops in Python.

Resources

- W3Schools Python Conditions:
https://www.w3schools.com/python/python_conditions.asp
- Real Python Python Loops: <https://realpython.com/python-for-loop/>

Example Questions

Q 1: Write a program that checks if a given number is positive, negative, or zero.

Q 2: Create a loop that prints the first 10 even numbers.

Q 3: Implement a program that finds the largest number in a list.



Practice Questions

1. Create a program that takes a year as input and checks if it is a leap year or not.
2. Given a list of integers, find all the even numbers and store them in a new list.
3. Write a Python program to check if a given number is a prime number.
4. Create a program that generates the Fibonacci sequence up to a given number of terms.
5. Given a list of names, print all names starting with the letter 'A'.
6. Implement a program that prints the multiplication table of a given number.
7. Write a program that calculates the factorial of a given number.
8. Create a loop that prints all prime numbers between 1 and 50.
9. Given a list of words, count the number of words with more than five characters.
10. Calculate the sum of digits of a given number.

FUNCTIONS

Aim

Understand functions and how to define and call them.

Resources

- W3Schools Python Functions:
[https://www.w3schools.com/python/
python_functions.asp](https://www.w3schools.com/python/python_functions.asp)
- Real Python Defining Functions:
[https://realpython.com/defining-your-own-python-
function/](https://realpython.com/defining-your-own-python-function/)

Example Questions

- Q 1:** Write a function to calculate the area of a circle given its radius.
- Q 2:** Create a function to check if a number is prime.
- Q 3:** Implement a function that reverses a given string.



Practice Questions

1. Given a list of numbers, create a function to find the sum of all positive numbers.
2. Write a Python function to check if a given string is a palindrome.
3. Implement a function that returns the factorial of a given number using recursion.
4. Create a function to find the square of each element in a given list.
5. Write a function to check if a number is even or odd and return "Even" or "Odd" accordingly.
6. Calculate the area of a triangle given its base and height using a function.
7. Create a function that takes a list of strings and returns the list sorted alphabetically.
8. Write a function that takes two lists and returns their intersection (common elements).
9. Implement a function to check if a given year is a leap year or not.
10. Create a function that takes a number as input and prints its multiplication table.

STRING MANIPULATION

Aim

Learn about common string operations and formatting.

Resources

- W3Schools Python Strings: https://www.w3schools.com/python/python_strings.asp
- Real Python Python String Formatting: <https://realpython.com/python-string-formatting/>

Example Questions

Q 1: Create a program that checks if a given string is a palindrome.

Q 2: Write a function to count the number of vowels in a given string.

Q 3: Write a function to count the number of vowels in a given string.



Practice Questions

1. Given a list of words, concatenate them into a single string separated by spaces.
2. Create a function to reverse a given string.
3. Write a program that takes a sentence as input and counts the number of words in it.
4. Implement a function that checks if a given string is a pangram (contains all letters of the alphabet).
5. Given a string, write a function to remove all vowels from it and return the modified string.
6. Write a Python program to find the length of the longest word in a sentence.
7. Create a function that takes a sentence as input and returns the sentence in reverse order.
8. Given a list of names, count the number of names that start with a vowel.
9. Write a function to remove all duplicate characters from a given string.
10. Implement a program that takes a sentence and a word as input and checks if the word is present in the sentence.

LISTS AND TUPLES

Aim

Understand lists and tuples in Python and their operations.

Resources

- W3Schools Python Lists: https://www.w3schools.com/python/python_lists.asp
- Real Python Lists and Tuples: <https://realpython.com/python-lists-tuples/>

Example Questions

Q 1: Given a list of numbers, find the sum and average using built-in functions.

Q 2: Create a list of fruits and add a new fruit to the list.

Q 3: Access elements in a tuple using indexing.



Practice Questions

1. Given two lists of numbers, concatenate them into a single list.
2. Write a program that finds the largest and smallest elements in a list.
3. Implement a function that takes a list of numbers and returns a new list with the squared values.
4. Create a program that finds the common elements between two lists and stores them in a new list.
5. Given a list of words, find the word with the maximum length and its length.
6. Write a Python program to count the occurrences of each element in a given list.
7. Given a list of names, remove all duplicate names and print the unique names.
8. Create a function that takes a list of strings and returns the list sorted by the length of the strings.
9. Write a program that checks if a given list is sorted in ascending order.
10. Implement a function that takes two lists and returns their union (all unique elements from both lists).

DICTIONARIES AND SETS

Aim

Understand dictionaries and sets in Python and their operations.

Resources

- W3Schools Python Dictionaries: https://www.w3schools.com/python/python_dictionaries.asp
- Real Python Dictionaries and Sets: <https://realpython.com/python-dicts/>

Example Questions

Q 1: Create a dictionary to store information about a person (name, age, address).

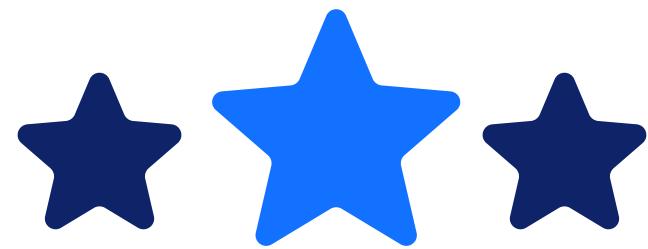
Q 2: Add a new key-value pair to an existing dictionary.

Q 3: Create a set of unique numbers from a list of numbers.



Practice Questions

1. Given two dictionaries, merge them into a single dictionary.
2. Write a program that finds the most frequent element in a list.
3. Implement a function that removes a key-value pair from a dictionary.
4. Create a program that checks if two sets have any elements in common.
5. Given a list of dictionaries, find the dictionary with the highest value for a specific key.
6. Write a Python program that counts the number of occurrences of each character in a given string using a dictionary.
7. Given two sets, find the union, intersection, and difference between them.
8. Create a function that takes a list of dictionaries and sorts them based on a specified key.
9. Write a program that finds the average value of all the elements in a list of dictionaries.
10. Implement a function that takes a list of strings and returns a set of unique characters present in all strings.



WHY BOSSCODER?

 **750+** Alumni placed at Top Product-based companies.

 More than **136% hike** for every **2 out of 3** working professional.

 Average package of **24LPA**.

The syllabus is most up-to-date and the list of problems provided covers all important topics.

Lavanya
 Meta



Course is very well structured and streamlined to crack any MAANG company

Rahul .
 Google



[EXPLORE MORE](#)

FILE HANDLING

Aim

Learn about reading and writing files in Python.

Resources

- W3Schools Python File Handling: https://www.w3schools.com/python/python_file_handling.asp
- Real Python Read and Write Files: <https://realpython.com/read-write-files-python/>

Example Questions

Q 1: Write a program that reads a text file and prints its contents.

Q 2: Create a new text file and write some content into it.

Q 3: Read a CSV file and process its data.



Practice Questions

1. Write a Python program to copy the contents of one text file into another.
2. Given a CSV file with student names and scores, find the student with the highest score.
3. Implement a program that reads a text file and counts the number of words and lines in it.
4. Create a function that takes a list of sentences and writes them to a new text file, each on a new line.
5. Given a CSV file with employee details (name, age, salary), calculate the average salary of all employees.
6. Write a program that reads a CSV file and finds the total sales revenue for a specific product.
7. Given a text file with a list of numbers, write a function that finds the sum of all numbers in the file.
8. Implement a program that reads a CSV file and generates a bar chart to represent the data using Matplotlib.
9. Write a function that reads a JSON file and extracts specific information from it.
10. Given a CSV file with temperature data for each day of the week, find the average temperature for each day.

OBJECT-ORIENTED PROGRAMMING (OOP)

Aim

Introduce Object-Oriented Programming (OOP) concepts in Python.

Resources

- W3Schools Python Classes: https://www.w3schools.com/python/python_classes.asp
- Real Python Python OOP: <https://realpython.com/python3-object-oriented-programming/>

Example Questions

Q 1: Create a class to represent a basic calculator with add, subtract, multiply, and divide methods.

Q 2: Create a class to represent a book with attributes like title, author, and publication year.

Q 3: Define a class for a Circle with methods to calculate its area and circumference.



Practice Questions

1. Create a class to represent a Student with attributes like name, age, and grades.
2. Given a CSV file with employee details (name, position, salary), create a class to represent an Employee.
3. Implement a program that simulates a basic bank account using a BankAccount class.
4. Write a Python program that uses a Rectangle class to calculate the area and perimeter of a rectangle.
5. Create a class to represent a Car with attributes like make, model, and year.
6. Given a JSON file with customer data, create a Customer class to store and manipulate the data.
7. Write a program that uses a Person class to keep track of a person's name, age, and address.
8. Implement a program that uses a Circle class to calculate the area and circumference of multiple circles.
9. Given a CSV file with product details (name, price, quantity), create a Product class to manage the data.
10. Create a class to represent a Movie with attributes like title, director, and rating.

INHERITANCE AND ENCAPSULATION

Aim

Understand inheritance and encapsulation in Object-Oriented Programming.

Resources

- W3Schools Python Inheritance: https://www.w3schools.com/python/python_inheritance.asp
- Real Python Inheritance and Composition: <https://realpython.com/inheritance-composition-python/>

Example Questions

Q 1: Create a base class Animal with a method sound() and create derived classes Dog and Cat with their own sound().

Q 2: Implement a class hierarchy to represent different types of vehicles (Car, Bike) with their own attributes and methods.

Q 3: Create a class Person with private attributes and define methods to get and set the values of those attributes.



Practice Questions

1. Create a base class `Shape` with methods to calculate area and perimeter, and derive classes `Circle` and `Square`.
2. Implement a class hierarchy to represent different types of employees (`Manager`, `Engineer`) with their attributes.
3. Write a Python program that uses inheritance to represent a hierarchy of shapes (`Triangle`, `Rectangle`, etc.).
4. Create a class hierarchy to represent different types of animals (`Bird`, `Fish`) with their own attributes and methods.
5. Given a JSON file with product details (name, price, quantity), create a `Product` class with encapsulated attributes.
6. Implement a program that uses inheritance to represent a hierarchy of vehicles (`Car`, `Bike`, `Truck`, etc.).
7. Write a Python program that uses encapsulation to protect sensitive information in a `User` class.



Practice Questions

8. Create a class hierarchy to represent different types of electronics (Phone, Laptop) with their attributes.
9. Given a CSV file with employee details (name, position, salary), create an Employee class with private attributes.
10. Implement a program that uses inheritance to represent a hierarchy of shapes (Circle, Triangle, Rectangle, etc.).

NUMPY

Aim

Introduction to NumPy for numerical computing in Python.

Resources

- NumPy Official Website: <https://numpy.org/>
- NumPy Quickstart Tutorial: <https://numpy.org/doc/stable/user/quickstart.html>

Example Questions

Q 1: Create a NumPy array from a Python list and perform basic operations like addition, multiplication, etc.

Q 2: Generate a NumPy array of random numbers and calculate its mean, median, and standard deviation.

Q 3: Create a NumPy array and reshape it into a different shape.



Practice Questions

1. Given a list of numbers, create a NumPy array and find the sum and product of its elements.
2. Implement a program that generates a NumPy array with numbers from 0 to 9 and reshapes it into a 3×3 matrix.
3. Write a Python program that uses NumPy to find the mean, median, and standard deviation of a dataset.
4. Create a function that takes a list of numbers and returns the NumPy array sorted in ascending order.
5. Given a list of lists, create a 2D NumPy array and find the sum of elements in each row and column.
6. Implement a program that generates a random NumPy array and finds the maximum and minimum values.
7. Write a function that takes a NumPy array and returns a new array with all elements squared.
8. Given a NumPy array, calculate the dot product of the array with itself.
9. Create a program that uses NumPy to calculate the inverse of a 2×2 matrix.
10. Implement a function that takes a NumPy array and returns the transpose of the array.

PANDAS

Aim

Introduction to Pandas for data manipulation and analysis in Python.

Resources

- Pandas Official Website: <https://pandas.pydata.org/>
- Pandas Getting Started: https://pandas.pydata.org/docs/getting_started/index.html

Example Questions

Q 1: Create a Pandas Series from a Python list and perform basic operations like filtering, sorting, etc.

Q 2: Read a CSV file into a Pandas DataFrame and perform basic data manipulation operations.

Q 3: Create a Pandas DataFrame from a dictionary and perform filtering and grouping operations.



Practice Questions

1. Given a CSV file with student details, read it into a Pandas DataFrame and find the average age of students.
2. Implement a program that generates a Pandas Series with dates and filter it to get dates in a specific range.
3. Write a Python program that uses Pandas to read a CSV file and find the maximum and minimum values in each column.
4. Create a function that takes a Pandas DataFrame and returns a new DataFrame with rows sorted in ascending order.
5. Given a Pandas DataFrame, filter the rows to include only the rows where a specific column meets a condition.
6. Implement a program that reads a CSV file into a Pandas DataFrame and calculates the sum of a specific column.
7. Write a function that takes a Pandas DataFrame and adds a new calculated column to the DataFrame.
8. Given a Pandas DataFrame, group the data by a specific column and calculate the mean of another column.
9. Create a program that reads a JSON file into a Pandas DataFrame and extracts specific information from it.
10. Implement a function that takes a Pandas DataFrame and returns the transpose of the DataFrame.

DATA VISUALIZATION WITH MATPLOTLIB AND SEABORN

Aim

Learn how to create data visualizations using Matplotlib and Seaborn in Python.

Resources

- Matplotlib Official Website: <https://matplotlib.org/>
- Seaborn Official Website: <https://seaborn.pydata.org/>

Example Questions

Q 1: Create a simple line plot using Matplotlib to visualize a series of data points.

Q 2: Generate a scatter plot using Matplotlib to visualize the relationship between two variables.

Q 3: Create a bar plot using Seaborn to compare the categories in a dataset.



Practice Questions

1. Given a Pandas DataFrame, create a line plot to visualize the trend of a specific column over time.
2. Implement a program that generates a histogram using Matplotlib to visualize the distribution of data.
3. Write a Python program that uses Seaborn to create a scatter plot matrix for multiple variables in a DataFrame.
4. Create a function that takes a Pandas DataFrame and generates a box plot to visualize the distribution of data.
5. Given a CSV file with sales data, use Matplotlib to create a bar plot to compare the sales of different products.
6. Implement a program that reads a JSON file into a Pandas DataFrame and uses Seaborn to create a violin plot.
7. Write a function that takes a Pandas DataFrame and generates a pair plot to visualize the relationships between variables.



Practice Questions



8. Given a Pandas DataFrame, create a pie chart using Matplotlib to visualize the distribution of data in a specific column.
9. Create a program that reads a CSV file into a Pandas DataFrame and uses Seaborn to create a swarm plot for data visualization.
10. Implement a function that takes a Pandas DataFrame and generates a heatmap using Seaborn to visualize the correlation between variables.

DATA CLEANING AND PREPROCESSING

Aim

Learn how to clean and preprocess data for analysis in Python.

Resources

- Data Cleaning with Python and Pandas: <https://realpython.com/python-data-cleaning-numpy-pandas/>
- Data Preprocessing with Scikit-Learn: <https://scikit-learn.org/stable/modules/preprocessing.html>

Example Questions

Q 1: Remove missing values from a Pandas DataFrame.

Q 2: Convert categorical variables into numerical values using Label Encoding.

Q 3: Scale numerical features using Min-Max Scaling.



Practice Questions

1. Given a Pandas DataFrame, remove duplicate rows and reset the index of the DataFrame.
2. Implement a program that reads a CSV file into a Pandas DataFrame and handles missing values using Imputation.
3. Create a function that takes a Pandas DataFrame and converts text data into numerical values using One-Hot Encoding.
4. Given a Pandas DataFrame, normalize the numerical features using Z-Score Normalization.
5. Write a Python program that uses Scikit-Learn to perform data standardization on a dataset.
6. Implement a program that reads a JSON file into a Pandas DataFrame and handles outliers using Winsorization.
7. Create a function that takes a Pandas DataFrame and removes irrelevant features using Feature Selection techniques.



Practice Questions

8. Given a CSV file with customer details, preprocess the data for further analysis (e.g., handle missing values, scale features).
9. Write a Python program that uses Scikit-Learn to perform data transformation using PCA (Principal Component Analysis).
10. Implement a function that takes a Pandas DataFrame and performs data discretization on a numerical feature.

MACHINE LEARNING BASICS

Aim

Introduction to the basics of Machine Learning in Python.

Resources

- Scikit-Learn Official Website: <https://scikit-learn.org/stable/>
- Machine Learning Basics: https://scikit-learn.org/stable/getting_started.html

Example Questions

Q 1: Split data into training and testing sets using Scikit-Learn.

Q 2: Train a Linear Regression model using Scikit-Learn.

Q 3: Evaluate the performance of a model using accuracy score.



Practice Questions

1. Given a CSV file with data about customers (features) and their churn status (target), split the data into training and testing sets.
2. Implement a program that uses Scikit-Learn to train a Decision Tree classifier on a dataset.
3. Write a Python program that uses Scikit-Learn to perform k-fold cross-validation on a dataset.
4. Create a function that takes a Pandas DataFrame and trains a Random Forest classifier on the data.
5. Given a CSV file with data about student scores (features) and their grades (target), split the data into training and testing sets.
6. Implement a program that uses Scikit-Learn to train a Support Vector Machine (SVM) classifier on a dataset.
7. Write a Python program that uses Scikit-Learn to perform hyperparameter tuning using Grid Search on a dataset.
8. Create a function that takes a Pandas DataFrame and trains a k-nearest neighbors (KNN) classifier on the data.
9. Given a CSV file with data about housing prices (features) and their labels (target), split the data into training and testing sets.
10. Implement a program that uses Scikit-Learn to train a Naive Bayes classifier on a dataset.