



AKCODING.COM

# Python programming interview questions and answers pdf

100 questions of Python programming interview questions and answers pdf, is a significant task, but I can certainly provide a substantial list to cover various aspects of Python programming. Here's a mix of questions ranging from basic to advanced levels along with their answers.

## Basics

- What is Python and why is it so popular?
- Python is a high-level, interpreted programming language known for its simplicity and readability. It's popular for web development, data analysis, artificial intelligence, and automation.
- How do you comment on a single line and multiple lines in Python?
- Single-line comment: `# This is a comment`
- Multiple-line comment: `''' This is a multi-line comment '''` or `""" This is a multi-line comment """`
- What is PEP 8?
- PEP 8 is a style guide for Python code, providing guidelines on how to format Python code for better readability and consistency.
- Explain the difference between Python 2 and Python 3.
- Python 3 is the latest version of Python, with improvements and new features compared to Python 2. Python 3 is not backward compatible with Python 2.
- How do you check the version of Python installed on your system?
- Open a terminal or command prompt and type `python --version` or `python -V`.

## Data Types and Operators

- What are the primitive data types in Python?
- Integers, floats, complex numbers, strings, and booleans.
- Explain the difference between `==` and `is` in Python.



- `==` checks for equality of values, while `is` checks for identity (whether two variables refer to the same object in memory).
- How do you concatenate two strings in Python?
- Strings can be concatenated using the `+` operator: `str1 + str2`.
- What is the difference between `and` and `&` in Python?
- `and` is a logical operator used for boolean expressions, while `&` is a bitwise operator used for performing bitwise AND operation.
- How do you convert a string to an integer in Python?
  - Use the `int()` function: `int("123")`.

## Control Flow

- Explain the `if`, `elif`, and `else` statements in Python.
  - `if` is used for conditional execution, `elif` is short for “else if,” and `else` is used for the final option if none of the previous conditions are true.
- What is a `for` loop and how does it work in Python?
  - A `for` loop is used for iterating over a sequence (such as a list, tuple, or string) and executing a block of code for each element in the sequence.
- How do you break out of a loop in Python?
  - You can use the `break` statement to exit a loop prematurely.
- What is the purpose of the `range()` function in Python?
  - The `range()` function generates a sequence of numbers, commonly used for looping a specific number of times in `for` loops.
- How do you iterate over a dictionary in Python?
  - You can use a `for` loop to iterate over the keys or values of a dictionary using the `keys()` or `values()` methods.

## Functions

- What is a function in Python?
  - A function is a block of reusable code that performs a specific task. It takes inputs, called parameters, and returns an output.
- How do you define a function in Python?
  - Use the `def` keyword followed by the function name and parameters, then a colon, and the function body.



- What is a default argument in Python?
  - A default argument is an argument that assumes a default value if a value is not provided in the function call.
- Explain the difference between `return` and `print` in Python.
  - `return` is used to exit a function and return a value to the caller, while `print` is used to display output to the console.
- How do you call a function recursively in Python?
  - A function can call itself recursively by using its own name inside its body.

## Data Structures

- What is a list in Python?
  - A list is a collection of elements that is ordered and mutable. Lists are defined by square brackets `[ ]`.
- How do you access elements of a list in Python?
  - Elements of a list can be accessed by index using square brackets:  
`my_list[index]`.
- Explain the difference between `append()` and `extend()` methods in Python.
  - The `append()` method adds an element to the end of a list, while the `extend()` method adds elements from an iterable to the end of a list.
- What is a tuple in Python?
  - A tuple is a collection of elements that is ordered and immutable. Tuples are defined by parentheses `( )`.
- How do you unpack a tuple in Python?
  - You can unpack a tuple by assigning its elements to individual variables:  
`a, b, c = my_tuple`.
- What is a dictionary in Python?
  - A dictionary is a collection of key-value pairs that is unordered and mutable. Dictionaries are defined by curly braces `{ }`.
- How do you access values in a dictionary in Python?
  - Values in a dictionary can be accessed by key using square brackets:  
`my_dict[key]`.
- Explain the difference between `keys()` and `values()` methods in Python dictionaries.



- The `keys()` method returns a view object containing the keys of the dictionary, while the `values()` method returns a view object containing the values of the dictionary.
- What is a set in Python?
  - A set is an unordered collection of unique elements. Sets are defined by curly braces `{ }`.
- How do you add elements to a set in Python?
  - You can add elements to a set using the `add()` method or by using the `update()` method to add elements from another set or iterable.

## Object-Oriented Programming

- What is a class in Python?
  - A class is a blueprint for creating objects. It defines properties (attributes) and behaviors (methods) that all objects of the class will have.
- How do you create an object of a class in Python?
  - You create an object of a class by calling the class name followed by parentheses: `my_object = MyClass()`.
- What is inheritance in Python?
  - Inheritance is the process by which a class can inherit attributes and methods from another class. It promotes code reusability and establishes a hierarchical relationship between classes.
- Explain the difference between an instance variable and a class variable in Python.
  - Instance variables belong to individual objects and are unique for each instance of a class. Class variables belong to the class itself and are shared among all instances of the class.
- What is method overriding in Python?
  - Method overriding occurs when a subclass provides a specific implementation of a method that is already defined in its superclass. The subclass method overrides the superclass method.
- What is method overloading in Python?
  - Python does not support method overloading in the same way as languages like Java or C++. However, you can achieve similar functionality by using default arguments
- or variable-length argument lists.

- What is encapsulation in Python?
  - Encapsulation is the bundling of data (attributes) and methods that operate on the data into a single unit, called a class. It hides the internal state of the object from the outside world.
- What is a constructor in Python?
  - A constructor is a special method in a class that is automatically called when an object of the class is created. In Python, the constructor method is named `__init__()`.
- What is a destructor in Python?
  - A destructor is a special method in a class that is automatically called when an object is about to be destroyed. In Python, the destructor method is named `__del__()`.
- What is a module in Python?
  - A module is a file containing Python code. It can define functions, classes, and variables, and can be imported and used in other Python scripts.

## File Handling

- How do you open a file in Python?
  - You can open a file using the `open()` function, specifying the file path and mode (read, write, append, etc.).
- What is the difference between “r”, “w”, and “a” file modes in Python?
  - “r” mode opens a file for reading, “w” mode opens a file for writing (creates a new file or overwrites an existing file), and “a” mode opens a file for appending (creates a new file or appends to an existing file).
- How do you close a file in Python?
  - You can close a file using the `close()` method or by using a context manager (`with` statement).
- How do you read the contents of a file in Python?
  - You can read the contents of a file using methods like `read()`, `readline()`, or `readlines()`.
- How do you write to a file in Python?
  - You can write to a file using the `write()` method or by using the `print()` function with the file argument.



## Exception Handling

- What is an exception in Python?
  - An exception is an error that occurs during the execution of a program. It disrupts the normal flow of the program and can be handled using exception handling mechanisms.
- How do you handle exceptions in Python?
  - Exceptions can be handled using `try`, `except`, `else`, and `finally` blocks. Code that may raise an exception is placed inside the `try` block, and the corresponding exception handling code is placed inside the `except` block.
- What is the purpose of the `finally` block in Python exception handling?
  - The `finally` block is used to execute cleanup code that should be run regardless of whether an exception occurs or not. It is often used to release resources like files or network connections.
- What is the difference between `except Exception as e` and `except` in Python?
  - `except Exception as e` catches and assigns the exception object to the variable `e`, allowing you to access information about the exception.
  - `except` without an exception type catches all exceptions.
- How do you raise an exception manually in Python?
  - You can raise an exception manually using the `raise` statement followed by the exception type and optional message.

## Advanced Topics

- What is a decorator in Python?
  - A decorator is a function that takes another function as an argument and extends its behavior without modifying it explicitly. Decorators are typically used to add functionality to existing functions.
- How do you define a decorator in Python?
  - You define a decorator by creating a function that takes another function as an argument, decorates it with additional functionality, and returns the modified function.
- What are lambda functions in Python?



AKCODING.COM

- Lambda functions, also known as anonymous functions, are small, inline functions defined using the `lambda` keyword. They are often used for short, one-time operations.
- How do you use lambda functions in Python?
  - Lambda functions are typically used in situations where a small, anonymous function is needed, such as when passing a function as an argument to higher-order functions like `map()`, `filter()`, or `sorted()`.
- What is the difference between `map()` and `filter()` functions in Python?
  - The `map()` function applies a given function to each item of an iterable and returns a list of the results. The `filter()` function applies a given function to each item of an iterable and returns a list of items for which the function returns `True`.
- What is list comprehension in Python?
  - List comprehension is a concise way to create lists in Python. It consists of an expression followed by a `for` clause, optionally followed by additional `for` or `if` clauses.
- How do you use list comprehension in Python?
  - List comprehension syntax: `[expression for item in iterable if condition]`.
- What is a generator in Python?
  - A generator is a special type of iterator that generates values lazily. It yields one value at a time using the `yield` keyword, allowing for efficient memory usage.
- How do you define a generator in Python?
  - Generators are defined using generator functions, which are regular functions that contain one or more `yield` statements. When called, a generator function returns a generator object.
- What is the purpose of the `yield` keyword in Python?
  - The `yield` keyword is used inside generator functions to yield values one at a time. It suspends the function's execution and returns a value to the caller, but retains the function's state, allowing it to resume where it left off.

## Concurrency and Parallelism



AKCODING.COM

- What is threading in Python?
  - Threading is a technique for concurrently executing multiple tasks within a single process. Python provides a `threading` module for creating and managing threads.
- How do you create a thread in Python?
  - You can create a thread by subclassing the `threading.Thread` class and implementing the `run()` method, or by passing a target function to the `Thread` constructor.
- What is the Global Interpreter Lock (GIL) in Python?
  - The Global Interpreter Lock (GIL) is a mutex that protects access to Python objects, preventing multiple native threads from executing Python bytecodes simultaneously. This can limit the performance of multi-threaded Python programs, especially on multi-core systems.
- How can you overcome the limitations of the GIL in Python?
  - You can use multiprocessing instead of threading to bypass the GIL and take advantage of multiple CPU cores. Alternatively, you can use asynchronous programming with libraries like `asyncio` or `twisted`.
- What is multiprocessing in Python?
  - Multiprocessing is a technique for parallel processing in Python, allowing multiple processes to run concurrently and utilize multiple CPU cores. Python provides a `multiprocessing` module for creating and managing processes.

## Regular Expressions

- What are regular expressions in Python?
  - Regular expressions are patterns used to match character combinations in strings. They provide a powerful and flexible way to search, match, and manipulate text.
- How do you use regular expressions in Python?
  - Regular expressions are supported in Python through the `re` module, which provides functions for working with regular expressions, such as `re.search()`, `re.match()`, `re.findall()`, etc.
- What is the purpose of the `re.search()` function in Python?





AKCODING.COM

- The `re.search()` function searches a string for a match with a regular expression pattern and returns the first match found, or `None` if no match is found.
- What is the difference between `re.match()` and `re.search()` in Python?
  - `re.match()` searches for a match only at the beginning of the string, while `re.search()` searches for a match anywhere in the string.
- How do you use groups in regular expressions in Python?
  - Groups in regular expressions are defined by enclosing parts of the pattern in parentheses. They can be accessed using the `group()` method of the match object.

## Networking

- What is socket programming in Python?
  - Socket programming is a way of connecting two nodes on a network to communicate with each other. Python provides a `socket` module for creating and managing network sockets.
- How do you create a TCP server in Python?
  - You can create a TCP server using the `socket` module by creating a socket, binding it to a specific address and port, and then listening for incoming connections.
- How do you create a TCP client in Python?
  - You can create a TCP client using the `socket` module by creating a socket, connecting it to a remote address and port, and then sending and receiving data.
- What is the difference between TCP and UDP in Python?
  - TCP (Transmission Control Protocol) is a connection-oriented protocol that provides reliable, ordered, and error-checked delivery of data. UDP (User Datagram Protocol) is a connectionless protocol that provides faster but less reliable delivery of data.
- How do you handle timeouts in socket programming in Python?
  - You can set a timeout on a socket using the `settimeout()` method, which specifies the maximum amount of time the socket will wait for an operation to complete before raising a timeout exception.



## Web Development

- What is Django?
  - Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It provides tools and features for building web applications quickly and efficiently.
- How do you install Django?
  - You can install Django using pip, the Python package manager, by running `pip install django` in the terminal or command prompt.
- What is Flask?
  - Flask is a lightweight Python web framework that provides tools and features for building web applications. It is known for its simplicity and flexibility, making it ideal for small to medium-sized projects.
- How do you install Flask?
  - You can install Flask using pip, the Python package manager, by running `pip install flask` in the terminal or command prompt.
- What is a virtual environment in Python?
  - A virtual environment is a self-contained directory that contains a Python interpreter and all the packages needed for a specific project. It allows you to isolate project dependencies and avoid conflicts between different projects.

## Data Science and Machine Learning

- What is NumPy?
  - NumPy is a Python library for numerical computing that provides support for multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.
- How do you install NumPy?
  - You can install NumPy using pip, the Python package manager, by running `pip install numpy` in the terminal or command prompt.
- What is Pandas?
  - Pandas is a Python library for data manipulation and analysis that provides data structures like DataFrames and Series, along with functions for reading, writing, and manipulating data.
- How do you install Pandas?



AKCODING.COM

- You can install Pandas using pip, the Python package manager, by running `pip install pandas` in the terminal or command prompt.
- What is Matplotlib?
  - Matplotlib is a Python library for creating static, animated, and interactive visualizations. It provides a MATLAB-like interface for creating plots and charts.
- How do you install Matplotlib?
  - You can install Matplotlib using pip, the Python package manager, by running `pip install matplotlib` in the terminal or command prompt.
- What is Scikit-learn?
  - Scikit-learn is a Python library for machine learning that provides tools and algorithms for tasks such as classification, regression, clustering, dimensionality reduction, and model selection.
- How do you install Scikit-learn?
  - You can install Scikit-learn using pip, the Python package manager, by running `pip install scikit-learn` in the terminal or command prompt.
- What is TensorFlow?
  - TensorFlow is an open-source machine learning framework developed by Google that provides tools and APIs for building and training machine learning models, including deep neural networks.
- How do you install TensorFlow?
  - You can install TensorFlow using pip, the Python package manager, by running `pip install tensorflow` in the terminal or command prompt.

## Testing

- What is unit testing?
  - Unit testing is a software testing technique where individual units or components of a software application are tested in isolation to ensure they work correctly.
- What is the `unittest` module in Python?



- The `unittest` module is a built-in Python module that provides a framework for writing and running unit tests. It allows you to define test cases, test suites, and test fixtures.
- How do you write a unit test in Python?
  - You write a unit test in Python by creating a subclass of `unittest.TestCase` and defining test methods that begin with the word `test`.
- What is test-driven development (TDD)?
  - Test-driven development (TDD) is a software development process where tests are written before the code. The developer writes a failing test case, writes the minimum amount of code to pass the test, and then refactors the code as needed.
- What is mocking in unit testing?
  - Mocking is a technique used in unit testing to replace parts of the code under test with mock objects. Mock objects simulate the behavior of real objects and allow you to control the inputs and outputs of the code being tested.

## Deployment

- What is Docker?
  - Docker is a platform for developing, shipping, and running applications in containers. Containers are lightweight, portable, and self-contained environments that contain everything needed to run an application, including code, runtime, system tools, and libraries.
- How do you create a Docker container for a Python application?
  - You create a Docker container for a Python application by writing a `Dockerfile`, which contains instructions for building the container, and then running the `docker build` command to build the container image.
- What is Kubernetes?
  - Kubernetes is an open-source container orchestration platform for automating the deployment, scaling, and management of containerized applications. It provides tools and APIs for managing clusters of containers across multiple hosts.
- How do you deploy a Python application to Kubernetes?



AKCODING.COM

- You deploy a Python application to Kubernetes by creating Kubernetes manifests (YAML files) that describe the desired state of the application, such as deployments, services, and ingresses, and then applying these manifests using the `kubectl apply` command.
- What is a serverless architecture?
  - Serverless architecture is a cloud computing model where the cloud provider manages the infrastructure and automatically scales resources up or down based on demand. Developers write
  - and deploy code as functions, which are executed in response to events triggered by external triggers or requests.

These questions cover a wide range of topics and should give you a good foundation for preparing for Python programming interviews.