1. How can you add a new key-value pair to a dictionary in Python?
2. del dictionary[key]

OR

My\_Dict = {1: "Mathew", 2: "Joe", 3: "Lizzy", 4: "Amos"}

data = My\_Dict.pop(1)

print(data)

print(My\_Dict)

OR

ADD ITEM in Dictionary

thisdict = {  
  "brand": "Ford",  
  "model": "Mustang",  
  "year": 1964  
}  
thisdict["color"] = "red"  
print(thisdict)

Update Dictionary

The update() method will update the dictionary with the items from a given argument. If the item does not exist, the item will be added.

The argument must be a dictionary, or an iterable object with key:value pairs.

Example

Add a color item to the dictionary by using the update() method:

thisdict = {  
  "brand": "Ford",  
  "model": "Mustang",  
  "year": 1964  
}  
thisdict.update({"color": "red"})

You can use the clear() method to remove all key-value pairs from a dictionary. The syntax is as follows:

dictionary.clear()

1. How do you raise an exception in Python?
2. What is the role of the cursor() method in Python's MySQL database connectivity?
3. What is the Django admin console used for in a Django project?
4. What is the role of HTTP methods in RESTful APIs?
5. Describe the purpose of the del keyword in Python.
6. Example
7. Delete an object:
8. class MyClass:  
     name = "John"  
     
   del MyClass  
     
   print(MyClass)
9. Definition and Usage
10. The del keyword is used to delete objects. In Python everything is an object, so the del keyword can also be used to delete variables, lists, or parts of a list etc.
11. More Examples
12. Example
13. Delete a variable:
14. x = "hello"  
      
    del x  
      
    print(x)
15. Example
16. Delete the first item in a list:
17. x = ["apple", "banana", "cherry"]  
      
    del x[0]  
      
    print(x)
18. How do you open a file in Python using the open() function?

f = open("demofile.txt", "r")  
print(f.read())

1. What is the role of views in Django and how are they mapped to URLs?
2. What is the primary purpose of resource representations in RESTful APIs?
3. What do HTTP status codes indicate in RESTful API responses?

HTML Error Messages

When a browser requests a service from a web server, an error might occur, and the server might return an error code like "404 Not Found".

It is common to name these errors HTML error messages.

But these messages are something called HTTP status messages. In fact, the server always returns a message for every request. The most common message is 200 OK.

Below is a list of HTTP status messages that might be returned:

1xx: Information

|  |  |
| --- | --- |
| **Message:** | **Description:** |
| 100 Continue | The server has received the request headers, and the client should proceed to send the request body |
| 101 Switching Protocols | The requester has asked the server to switch protocols |
| 103 Early Hints | Used with the Link header to allow the browser to start preloading resources while the server prepares a response |

2xx: Successful

|  |  |
| --- | --- |
| **Message:** | **Description:** |
| 200 OK | The request is OK (this is the standard response for successful HTTP requests) |
| 201 Created | The request has been fulfilled, and a new resource is created |
| 202 Accepted | The request has been accepted for processing, but the processing has not been completed |
| 203 Non-Authoritative Information | The request has been successfully processed, but is returning information that may be from another source |
| 204 No Content | The request has been successfully processed, but is not returning any content |
| 205 Reset Content | The request has been successfully processed, but is not returning any content, and requires that the requester reset the document view |
| 206 Partial Content | The server is delivering only part of the resource due to a range header sent by the client |

3xx: Redirection

|  |  |
| --- | --- |
| **Message:** | **Description:** |
| 300 Multiple Choices | A link list. The user can select a link and go to that location. Maximum five addresses |
| 301 Moved Permanently | The requested page has moved to a new URL |
| 302 Found | The requested page has moved temporarily to a new URL |
| 303 See Other | The requested page can be found under a different URL |
| 304 Not Modified | Indicates the requested page has not been modified since last requested |
| 307 Temporary Redirect | The requested page has moved temporarily to a new URL |
| 308 Permanent Redirect | The requested page has moved permanently to a new URL |

ADVERTISEMENT

4xx: Client Error

|  |  |
| --- | --- |
| **Message:** | **Description:** |
| 400 Bad Request | The request cannot be fulfilled due to bad syntax |
| 401 Unauthorized | The request was a legal request, but the server is refusing to respond to it. For use when authentication is possible but has failed or not yet been provided |
| 402 Payment Required | *Reserved for future use* |
| 403 Forbidden | The request was a legal request, but the server is refusing to respond to it |
| 404 Not Found | The requested page could not be found but may be available again in the future |
| 405 Method Not Allowed | A request was made of a page using a request method not supported by that page |
| 406 Not Acceptable | The server can only generate a response that is not accepted by the client |
| 407 Proxy Authentication Required | The client must first authenticate itself with the proxy |
| 408 Request Timeout | The server timed out waiting for the request |
| 409 Conflict | The request could not be completed because of a conflict in the request |
| 410 Gone | The requested page is no longer available |
| 411 Length Required | The "Content-Length" is not defined. The server will not accept the request without it |
| 412 Precondition Failed | The precondition given in the request evaluated to false by the server |
| 413 Request Too Large | The server will not accept the request, because the request entity is too large |
| 414 Request-URI Too Long | The server will not accept the request, because the URI is too long. Occurs when you convert a POST request to a GET request with a long query information |
| 415 Unsupported Media Type | The server will not accept the request, because the media type is not supported |
| 416 Range Not Satisfiable | The client has asked for a portion of the file, but the server cannot supply that portion |
| 417 Expectation Failed | The server cannot meet the requirements of the Expect request-header field |

5xx: Server Error

|  |  |
| --- | --- |
| **Message:** | **Description:** |
| 500 Internal Server Error | A generic error message, given when no more specific message is suitable |
| 501 Not Implemented | The server either does not recognize the request method, or it lacks the ability to fulfill the request |
| 502 Bad Gateway | The server was acting as a gateway or proxy and received an invalid response from the upstream server |
| 503 Service Unavailable | The server is currently unavailable (overloaded or down) |
| 504 Gateway Timeout | The server was acting as a gateway or proxy and did not receive a timely response from the upstream server |
| 505 HTTP Version Not Supported | The server does not support the HTTP protocol version used in the request |
| 511 Network Authentication Required | The client needs to authenticate to gain network access |

1. How can you check if a value exists in a set in Python?

thisset = {"apple", "banana", "cherry"}  
  
if "apple" in thisset:  
  print("Yes, 'apple' is in this set")

1. What is the purpose of abstraction in object-oriented programming?
2. What is the purpose of Python libraries like pymysql and pymongo?
3. How do URLs contribute to RESTful API design?
4. What function is used to convert a string to lowercase in Python?
5. message = 'PYTHON IS FUN'
6. # convert message to lowercase
7. print(message.lower())
8. # Output: python is fun
9. How is the super() function utilized in Python?
10. What data structures in Python are similar to lists but are immutable?
11. What is the term for the process of bundling data and methods together within a class?
12. Which method is used to insert a document into a MongoDB collection using PyMongo?
13. For what purpose does the Django admin console provide a user interface?
14. Which Python library is commonly used for consuming RESTful APIs?
15. Which keyword is used to exit a loop prematurely in Python?
16. In MySQL database connectivity with Python, which method is used to execute SQL queries?
17. What administrative tasks can be performed using the Django admin console?
18. Explain the difference between a set and a dictionary in Python.
19. How do you handle a ZeroDivisionError exception in Python?
20. Explain the structure of a typical Django app.
21. What is Routing in Flask?
22. What is the primary purpose of resource representations in RESTful APIs?
23. What is the purpose of the in keyword in Python? Provide an example.
24. The in keyword has two purposes:
25. The in keyword is used to check if a value is present in a sequence (list, range, string etc.).
26. The in keyword is also used to iterate through a sequence in a for loop:
27. Example
28. Loop through a list and print the items:
29. fruits = ["apple", "banana", "cherry"]  
      
    for x in fruits:  
      print(x)
30. Describe how you would handle a FileNotFoundError in Python.
31. Explain the structure of a typical Django app.
32. Explain the role of HTTP methods GET, POST, PUT, and DELETE in RESTful APIs.

What is REST architecture?

REST stands for REpresentational State Transfer. REST is web standards based architecture and uses HTTP Protocol.

In REST architecture, a REST Server simply provides access to resources and REST client accesses and modifies/updates the resources. Here each resource is identified by URIs

REST uses various representation to represent a resource like text, JSON, XML. JSON is the most popular one.

HTTP methods

Following four HTTP methods are commonly used in REST based architecture.

* **GET** − Provides a read only access to a resource. SELECT QERUY
* **POST** − Used to create a new resource. INSERT QUERY
* **DELETE** − Used to remove a resource. DELETE QUERY
* **PUT** − Used to update a existing resource or create a new resource. UPDATE / INSERT QUERY

Introduction to RESTFul web services

A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. This interoperability (e.g., between Java and Python, or Windows and Linux applications) is due to the use of open standards.

Web services based on REST Architecture are known as RESTful web services. These webservices uses HTTP methods to implement the concept of REST architecture. A RESTful web service usually defines a URI, Uniform Resource Identifier a service, provides resource representation such as JSON and set of HTTP Methods.

Creating RESTFul Webservice

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.No.** | **URI** | **HTTP Method** | **POST body** | **Result** |
| 1 | /UserService/users | GET | empty | Show list of all the users. |
| 2 | /UserService/addUser | POST | JSON String | Add details of new user. |
| 3 | /UserService/getUser/:id | GET | empty | Show details of a user. |

What is HTTP?

The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.

HTTP works as a request-response protocol between a client and server.

Example: A client (browser) sends an HTTP request to the server; then the server returns a response to the client. The response contains status information about the request and may also contain the requested content.

HTTP Methods

* **GET**
* **POST**
* **PUT**
* **HEAD**
* **DELETE**
* **PATCH**
* **OPTIONS**
* **CONNECT**
* **TRACE**

The two most common HTTP methods are: GET and POST.

The GET Method

GET is used to request data from a specified resource.

Note that the query string (name/value pairs) is sent in the URL of a GET request:

/test/demo\_form.php?name1=value1&name2=value2

**Some notes on GET requests:**

* GET requests can be cached
* GET requests remain in the browser history
* GET requests can be bookmarked
* GET requests should never be used when dealing with sensitive data
* GET requests have length restrictions
* GET requests are only used to request data (not modify)

The POST Method

POST is used to send data to a server to create/update a resource.

The data sent to the server with POST is stored in the request body of the HTTP request:

POST /test/demo\_form.php HTTP/1.1  
Host: amit.com  
  
name1=value1&name2=value2

**Some notes on POST requests:**

* POST requests are never cached
* POST requests do not remain in the browser history
* POST requests cannot be bookmarked
* POST requests have no restrictions on data length

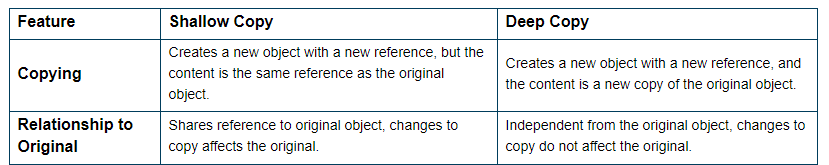
The PUT Method

PUT is used to send data to a server to create/update a resource.

The DELETE Method

The DELETE method deletes the specified resource.

1. How do you establish a connection to a MySQL database in Python using the pymysql library?
2. Explain the difference between a shallow copy and a deep copy of a list in Python.



SHALLOW

import copy

*# Shallow copy example*

original\_list = [[1, 2, 3], [4, 5, 6]]

shallow\_copy = copy.copy(original\_list)

*# Modify the first element of the shallow copy*

shallow\_copy[0][0] = 9

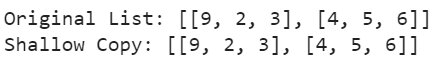
*# Both the original and shallow copy will reflect the change*

print("Original List:", original\_list)

print("Shallow Copy:", shallow\_copy)

**Copy code**

**Output:**



DEEP

import copy

# Deepcopy example

original\_list = [[1, 2, 3], [4, 5, 6]]

deep\_copy = copy.deepcopy(original\_list)

# Modify the first element of the deep copy

deep\_copy[0][0] = 9

# Only the deep copy will reflect the change

print("Original List:", original\_list)

print("Deep Copy:", deep\_copy)

Copy code

Output:

2023\_06\_4-1.jpg

1. How do you implement method overriding in Python?
2. How do you define and register models in a Django project?
3. What are the basic CRUD operations, and how are they performed in PyMongo for MongoDB?
4. Describe how you would handle a FileNotFoundError in Python.
5. Discuss the role of templates in Django and how they are rendered.
6. A program that generates a random password of a specified length.
7. Describe the process of creating a new app within a Django project.
8. A program that checks if a given string is a palindrome.
9. Write a program to demonstrate how to insert a row into the table using MySQL and Python.
10. A program that sorts a list of numbers in ascending or descending order.



1. A program that creates a web application that allows users to register and login.
2. Select query using MySQL and Python
3. Delete query using MySQL and Python