

[By OnlineInterviewQuestions.com](http://OnlineInterviewQuestions.com)

[Python Flask Interview Questions](#)

What is Python Flask?

Flask is one of the Python frameworks, which is based on Werkzeug and JINJA 2 and is inspired by the Sinatra Ruby framework that is available under the BSD license. Armin Ronacher developed a flask at POCCO.

Though Flask is younger as compared to other [Python frameworks](#), it holds a great future and is seen to have gained popularity among web developers of Python. Flask also called a micro-framework for Python was designed to be simple to use and to extend. The main goal behind Flask was to build a firm foundation for web applications of different needs and complexities. Afterward, users are free to use any extensions they need and want to use. Also, one is free to build their own modules and Flask is great for all such kinds of projects.

It is exceptionally good for prototyping and Flask depends on two external libraries namely the Jinja2 template engine and the other being the Werkzeug WSGI toolkit. Flask is one of the most stylish and loaded with features of micro frameworks available in Python. Still a very new framework, Flask can be looked upon as a thriving community, an elegant API, and first-class extensions. Flask has all the benefits, which should be in fast templates and has strong WSGI features; in addition, it has tough unit testability at the web application as well as library level, and extensive documentation.

Finally, practice here the best **18+ Python Flask Interview Questions and Answers** to check your final preparation for your job interviews.

Q1. [What are the features of Flask Python?](#)

Flask Python is one of the newest frameworks of Python and is used for designing web applications for the following features:

- Flask comes with a built-in development server as well as fast debugger
- It also contains the integrated support required for unit testing
- It has the feature of restful request dispatching
- Comes with the Jinja2 templating technique
- Flask supports secure cookies i.e. client-side sessions
- Also has the WSGI 1.0 compliant feature.
- It is based on Unicode.
- Python Flask is extensively documented.

Q2. [What are the advantages of Flask Python?](#)

Flask Python comes with all the advantages of Python and some additional pros of it are:

- Flask's design is lightweight and modular. Therefore, it is easy to transform it into the web applications or framework when one needs very few extensions without weighing much.
- Flask is ORM-agnostic: i.e. user can plug in their favorite ORM like SQLAlchemy
- The basic foundation of API is very nicely shaped and made coherent.
- Documentation of flask is very comprehensive, filled with lots of examples and are well structured. Users can even try out some sample applications to really get the real feel of Flask.
- It is very easy to deploy Flask in production as Flask comes with 100% WSGI 1.0 compliant
- Flask can handle HTTP request easily with help of its functionalities
- It is highly flexible. Its configuration is even more flexible than that of Django, which gives its users plenty of solutions for every product they need.

Q3. What is Flask-WTF and what are their features?

Flask-WTF is featured to offer simple integration with WTForms. The Features include for Flask WTF are:

- Provides Integration with web forms
- Is very secure form as it comes with CSRF token
- Provides global CSRF protection
- Comes with internationalization integration
- Also features Recaptcha supporting
- Contains File upload that closely works with Flask Uploads

Q4. How long can an identifier be in Flask Python?

In Flask Python, an identifier can be of any length. Also, there are certain rules that the users must follow to name an identifier

- It should begin with a character or an underscore or from A-Z or a-z.
- The rest of the name of identifier can contain anything from the following: A-Z or a-z or 0-9 or _.

Flask Python is case-sensitive so it will treat upper case and lower case letters differently.

There are certain words, which are reserved in Python called keywords and they cannot be used as identifiers. Some of them are listed below:

and, def, false, import, not, true, as, del, finally, in, or, try, assert, elif, for, is, pass, while, break, else, from, lambda, print, with, class, except, global, none, raise, yield, continue, exec, if, nonlocal, return

Q5. What are the HTTP methods provided by Python Flask?

HTTP methods are used to retrieve data from an URL:

- **GET:** The GET is the method that sends data to the server unencrypted.
- **HEAD:** HEAD is similar to GET, but that it has no response body.
- **POST:** The POST server does not cache the HTML form data that it sends
- **PUT:** It is the method in which the uploaded content replaces current data representations of the target resources.
- **DELETE:** This method removes the current representations of the target resource that is suggested by a URL.

Q6. What do you mean by template engines in Flask Python?

When users built a website they often face the problem to keep the style of the website consistent. Sometimes the users have had to write multiple times the same text when they ever try to change the style of such websites. If the website contains only a few pages, changing its style will take the users only some time which is doable. However, if they have a lot of pages (for example the list of items sold in a mall), this task becomes monotonous and hectic.

Using templates the users may set a basic layout for all their pages and provide which element needs to be changed frequently. Using this way the users can define their header once and keep it consistent in all the pages of their website, and if they need to change their header, they will only have to update it at one place. Making use a template engine will save the users a whole lot of time not only while they create their application but also when they are updating and maintaining it.

Q7. What are the major differences between Pyramid and Flask?

Flask can be stated as a micro framework, which is solely built for a small application, which has simpler requirements. In flask, the users have to use external libraries. Flask is always ready to use.

Pyramid, on the other hand, is built for larger application as it provides flexibility and allows the developer use the right features for their project. The developer can choose the database, templating style URL structure and more. Pyramid is therefore heavy configurable.

Q8. Explain how can one-request database connections in Flask?

Flask framework allows to its users to request database in three ways. They are as follows:

- **before_request()-**These connections are called before making a request and no arguments are passed
- **after_request() :** These connections are called after making a request and response is passed that will be sent to the client.
- **teardown_request():** These connections are called in a situation where an exception is raised and the response are not sure to get. They are also called after the construction of response. These are not allowed

to change the request, and their values can be ignored.

Q9. Explain can a request context be created in Flask Python?

A request profile in Flask Python can be created in two ways:

- On its own or auto when the application receives a request from the system
- Manually that is by calling `app.test_request_context`

Q10. Mention how one can enable debugging in Flask Python ?

There are two ways by which users can enable debugging in Flask. They are as follows:

- By setting the flag on the applications object
- By passing the flag as a parameter to run. If the user is enabling to debug support, the server will reload it when the code will change and the user doesn't have to restart after each change made in the code.

Q11. What do you mean by the Thread-Local object in Flask Python?

Flask Python makes use of thread-local objects internally so that the user doesn't have to pass objects around from one function to another function within a request so as to stay thread safe. This approach is quite useful, but it requires a pure request context for dependency injection or while attempting to reuse code, which uses a value indulged in the requests.

Flask Python supports all kinds of database-powered applications like RDBS. Such systems require the creation of a schema, which further requires connecting the schema.sql file to a sqlite3 command. So users need to install the sqlite3 command if they want to create or start the database in Flask Python.

Q12. How can one-access sessions in Flask Python? State whether Flask Python is an MVC model.

A session in Flask Python is a feature that allows one to remember the information from one request to another. In a flask program, it makes use of a signed cookie so that the user can look at the contents of the session and modify them. The user can also modify the sessions if and only if it has the secret key called the `Flask.secret_key`. Flask is a small form of Python framework, which behaves the same as the MVC framework. So MVC is a perfect match for Flask.

Q13. What do you mean by a decorator? Name some PDB commands and their uses.

A decorator is defined as a function that adds functionality to another function without changing it. It wraps the function to add some functionality to it. Some PDB commands include

- : It adds a breakpoint
- <c>: It resumes the execution
- <s>: It debugs step by step
- <n>: It moves to next line
- <l>: It lists the source code
- <p>: It prints an expression

Q14. What do you mean by NumPy? Is it better than a list?

NumPy is one of the Flask Python packages which have made its identity in the era of scientific computing. It deals with large data sizes, and also contains a powerful N dimensional array object along with a set of advanced functions.

A NumPy array is much better than a list. Here are the ways:

- NumPy is more compact than list.
- NumPy is more convenient than the list.
- Numpy is more efficient than List.
- It is simpler to read and write items with NumPy.

Q15. What do you mean by pickling and unpickling?

To make a portable and serialized representations of Python objects, we have the module known as pickle which accepts a Python object (basically everything in Python is an object) and then converts it into a string type, and after that uses the dump () function to dump it into a file. We term this as pickling.

On the contrary, retrieving objects from the stored string forms is called as unpickling.

Q16. How is memory managed in Flask Python?

Flask Python is a collection of private heap spaces, which holds all objects and data structures together. Programmers cannot access it. It is the task of the interpreter to manage it. But in the core API, users can access some of the tools. The Flask Python memory manager controls its allocation. Also, an inbuilt garbage collector is present which recycles all unused memory so it is made available for the heap space.

Q17. Explain how can you structure a large Flask application ?

To structure a large flask application, one needs to follow these steps:

1. Connect to the functions and then move them to various files, as long as users are assured that it will get imported as the applications start
2. Then use blueprints to assign the views to various categories such as auth, backend, profile, etc.
3. Then use the hidden Werkzeug URL map and then register functions on a central URL map.

Q18. What are the requirements to create the database in Flask Python?

Flask Python supports all kinds of database-powered applications like RDBS. Such systems require creating of a schema, which further requires connecting the schema.sql file to a sqlite3 command. So users need to install sqlite3 command if they want to create or start the database in Flask Python.

Please Visit [OnlineInterviewquestions.com](https://www.onlineinterviewquestions.com) to download more pdfs