|  |
| --- |
|  |
| Flask |
|  |

|  |
| --- |
| Soubika BISOO  17/01/2022 |

Table des matières

[I) Qu’est-ce que Flask ? 2](#_Toc93306692)

[II) Qu’est-ce que Django ? 2](#_Toc93306693)

[III) Flask vs Django 3](#_Toc93306694)

[1) Flask vs Django : Bases de données 3](#_Toc93306695)

[2) Flask vs Django : Taille du projet 3](#_Toc93306696)

[3) Flask vs Django : Performances 3](#_Toc93306697)

[4) Flask vs Django : Sécurité 3](#_Toc93306698)

[5) Flask vs Django : Système de routage et vues 4](#_Toc93306699)

[6) Flask vs Django : Autres 4](#_Toc93306700)

[7) Flask vs Django : Quel framework choisir ? 4](#_Toc93306701)

[Annexe 5](#_Toc93306702)

De nombreux frameworks web basés sur Python permettent aux développeurs de créer rapidement des applications évolutives. Qu’ils s’agissent de sites web simples ou complexe, ces frameworks peuvent tout faire. Parmi les nombreux choix populaires, Django et Flask sont ceux qui font le plus parler d’eux, principalement parce qu’ils sont similaires à certains égards et différents à d’autres !

Django d’une part est un framework web complet, tandis que flask est un framework léger et extensible.

# Qu’est-ce que Flask ?

Flask est un micro framework offrant les fonctionnalités de base d’une application web. Ce framework ne dépend d’aucune bibliothèque externe. Le framework offre des extensions pour la validation de formulaires, des mappeurs objets-relationnel, des systèmes d’authentification ouverts, un mécanisme de téléchargement et plusieurs autres outils.

Les principales caractéristiques de Flask :

* Un support intégré pour les tests unitaires
* Serveur de développement et débogueur rapide
* Moteur de modèles Jinja2
* Werkzeug, qui offre une prise en charge robuste de l’interface passerelle pour serveur web (WSGI), et Jinja2, qui offre un puissant moteur de modèles web
* Prise en charge des cookies sécurisés dans les sessions côté client
* Documentation détaillée
* APIs claires et soignées
* Déploiement de production simple
* Moins de 10 000 lignes de code source

# Qu’est-ce que Django ?

Django est un framework de développement Web pour [Python](https://mobiskill.fr/blog/conseils-emploi-tech/quest-ce-que-le-python-et-quelle-est-son-utilisation/). Ce framework offre une méthode standard pour le développement rapide et efficace de sites Web. Il t’aide à créer et à maintenir des applications Web de qualité. Il te permet de rendre le processus de développement fluide et de gagner du temps.

Il s’agit d’un framework web de haut niveau qui permet d’effectuer un développement rapide. L’objectif principal de ce framework web est de créer des sites web complexes pilotés par des bases de données.

Les principales caractéristiques de Django :

* Architecture Modèle-Vue-Contrôleur (MVC)
* Bibliothèques prédéfinies pour les graphiques, l’imagerie, les calculs scientifiques, etc.
* Prise en charge de plusieurs bases de données
* Système d’exploitation multiplateforme
* Optimisation du site sur des serveurs spécialisés
* Prise en charge des outils frontaux tels que Ajax, Pyjamas, jQuery, etc.
* Support multi-langues et multi-caractères
* Il comporte environ 24 000 lignes de code source

# Flask vs Django

## Flask vs Django : Bases de données

Flask ne prend pas en charge le framework ORM (Mapping objet-relationnel) et les développeurs sont donc libres de choisir l’ORM qui convient à leur application. En revanche, Django fournit un ORM puissant et un outil de gestion de la migration qui facilite le démarrage.

Django prend en charge un certain nombre de bases de données relationnelles telles qu’Oracle, MySQL, SQLite, MariaDB et PostgreSQL. Flask prend en charge non seulement les bases de données relationnelles mais aussi les bases de données non relationnelles comme MongoDB.

L’ORM de Django permet de créer facilement des modèles, des vues, des formulaires, etc. qui sont basés sur des modèles de données, mais Flask ne prend pas en charge les modèles de données.

## Flask vs Django : Taille du projet

Si tu optes pour une application web de petite taille, Flask sera un choix optimal car il est bien adapté aux petites applications web non compliquées et à forte charge. D’un autre côté, Django est un excellent choix pour les projets de plus grande envergure puisqu’il est conçu pour les grandes applications compliquées et à fort trafic. Étant donné que Flask est modulaire et très flexible, son extension peut se traduire par un code complexe et sujet aux erreurs. Avec Django, il est relativement facile de passer à l’échelle supérieure grâce aux fonctionnalités complètes qu’il offre.

## Flask vs Django : Performances

Flask surpasse définitivement Django en termes de performances et est nettement plus rapide que Django, car il comporte moins de fonctionnalités, moins de couches et est un framework léger. Django fournit beaucoup de fonctionnalités qui ne sont pas présentes dans Flask pour un développement rapide et a donc une baisse de performance par rapport à son homologue, Flask.

## Flask vs Django : Sécurité

Flask et Django fournissent des mécanismes similaires pour prévenir les attaques d’applications web et autres menaces. Django dispose d’un système intégré pour prévenir les attaques courantes telles que le Cross-Site Request Forgery (CSRF), le Cross-Site Scripting (XSS), l’injection SQL, etc. Cela permet d’empêcher ou d’atténuer les attaques qui permettent d’exécuter des scripts dans le navigateur, de se connecter en utilisant les informations d’identification d’autres utilisateurs et d’exécuter un code SQL arbitraire dans la base de données. Comme les applications Flask sont beaucoup plus petites, elles sont plus faciles à sécuriser, car il y a moins de surface disponible pour les attaques. Mais comme Flask repose sur des extensions de sécurité externes, il se peut que si l’extension elle-même est compromise, l’application web Flask le soit aussi.

## Flask vs Django : Système de routage et vues

Le routage est une partie importante de toute application web et ces deux frameworks prennent en charge le routage et les vues basées sur les classes. Il crée des localisateurs de ressources universelles (URL), puis les associe à leurs fonctions respectives qui, à leur tour, déterminent ce qui doit être fait lorsque l’URL est chargée. Dans Django, les URL et les vues sont définies dans urls.py et views.py. Pour accéder à l’objet de requête dans Django, tu devras le passer explicitement, alors que, dans Flask, l’objet de requête est global et donc facilement accessible.

## Flask vs Django : Autres

La documentation de Flask est étendue et très accueillante pour les débutants qui veulent mettre la main sur Flask. Elle implique une explication détaillée et couvre tout, de l’installation au déploiement, et elle a une courbe d’apprentissage plus faible que celle de Django. En revanche, la documentation de Django est beaucoup plus complète que celle de Flask, car il s’agit d’un framework complet et non d’un framework minimaliste. En outre, il existe un système d’administration prêt à l’emploi qui n’est pas présent dans Flask.

## Flask vs Django : Quel framework choisir ?

Alors que Django est considéré comme meilleur pour un projet plus important avec un développement rapide et offre plus de fonctionnalités, Flask semble être plus facile à utiliser pour commencer. Les deux frameworks sont très utiles pour le développement d’applications Web et doivent être choisis en fonction des exigences, des besoins et de la taille du projet.

# Annexe

<https://www.kicklox.com/blog-client/flask-vs-django-framework-python>

<https://mobiskill.fr/blog/conseils-emploi-tech/flask-vs-django-quel-framework-choisir/>

<https://waytolearnx.com/2019/04/difference-entre-django-et-flask.html>

https://easypartner.fr/blog/flask-vs-django/