

Flask - Historique

IFT₃₂₂₅ – Technologie de l'Internet

Jamel Eddine Jridi

Steve Lévesque



Table des Matières

- Les origines de Flask
- Information utile
- Approche et fonctionnement
- Points positifs
- Inconvénients
- Comparaison

Les origines de Flask

- Son but de départ était d'être une simple couche pour Werkzeug et Jinja (des projets du même groupe « The Pallet Projects »).
- Cela à pris un autre tour et Flask est maintenant un framework web Python assez populaire.
- Créé par Armin Ronacher, la première version sort le 1er avril 2010.
- Cette première version de Flask est sortie 5 ans après Django.
- La dernière version en date de l'été 2021 est le 21 mai 2021.

Information utile

- 600+ contributeurs et source libre.
- Utilisé par 772k personnes.
- 55.9k + étoiles (2.2k+ regards et 14.4k+ « forks ») sur GitHub.
- Léger et permet un apprentissage rapide et facile.
- Un front-end React peut être utilisé pour la partie page web.
- Adhère à l'intégration continue.
- Multiplateforme pour le développement.

*Informations en date de l'été 2021

Approche et fonctionnement

- Flask est un framework Web fonctionnant avec Python.
- L'interface web traditionnel est le HTML et CSS.
- React ou un framework front-end peut être utilisé pour compléter.
- Flask est un micro web framework n'ayant pas besoin strictement d'outils et de librairies pour son bon fonctionnement. (bien que des extensions pour des fonctions avancées doivent être ajoutées)
- Intégration de base de données facile. Il n'y a pas de couche d'accès ou un ORM en dépendance.

Approche et fonctionnement

- Page simple avec Flask.

```
# save this as app.py
from flask import Flask, escape, request

app = Flask(__name__)

@app.route('/')
def hello():
    name = request.args.get("name", "World")
    return f'Hello, {escape(name)}!'
```

```
$ flask run
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Hello, World!

Points positifs

- Le « Routing » est inclus, et ce pour une librairie assez légère.
- Facile à débiter avec une base en Python et en Web.
- La couche de données peut être modifiée aux besoins pour supporter des types et une étendue de sortes de BD (ex : SQLite, MySQL (relationnel), MongoDB (document), Redis (No SQL))

Inconvénients

- Des extensions pour des tâches web assez importantes de nos jours doivent être importées.
- Approche de développement libre, besoin d'une direction bien établie.

Comparaison

- Différences entre les compétiteurs populaires.

django

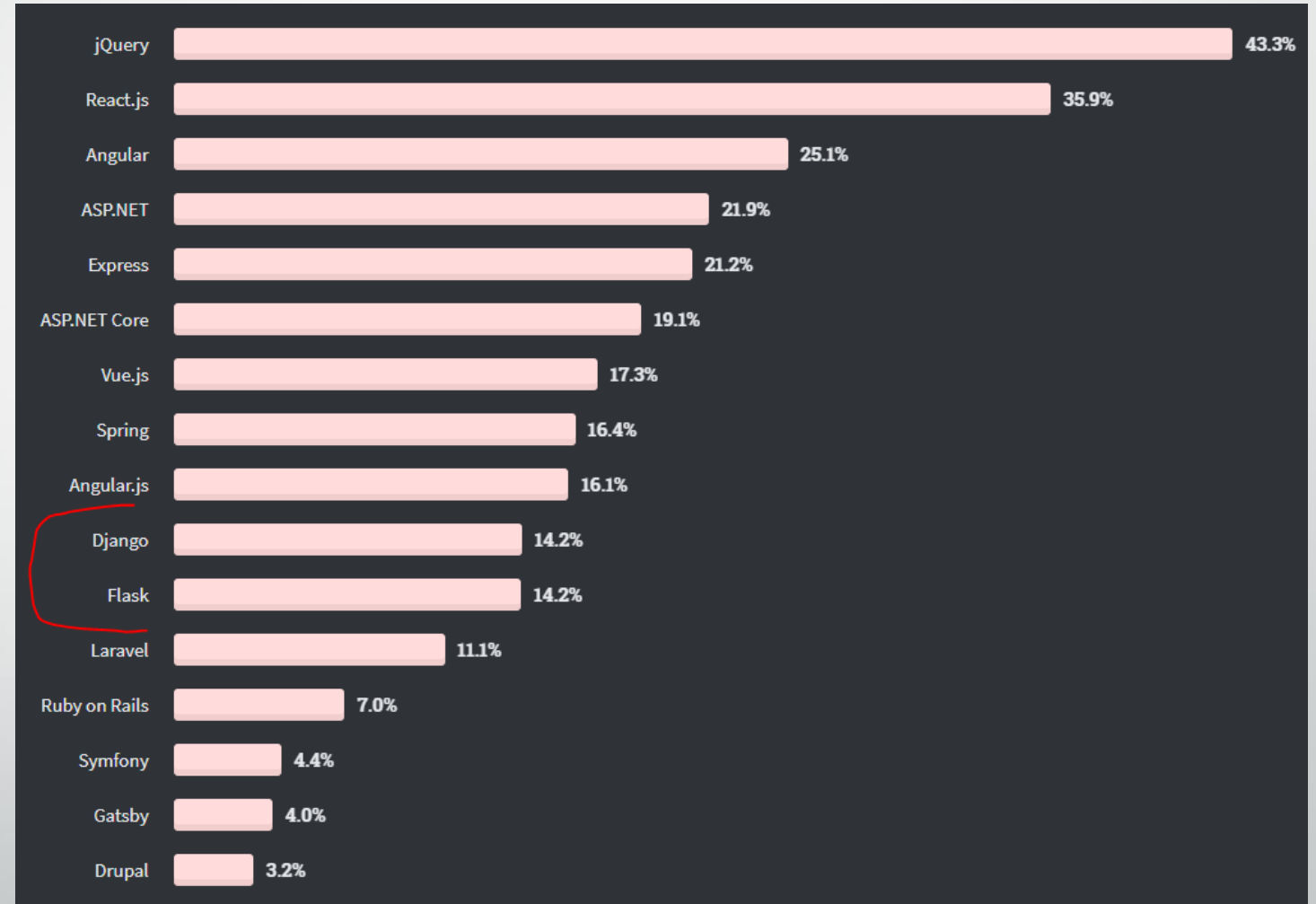
vs



Comparaison

- Partage de l'utilisation.

<https://insights.stackoverflow.com/survey/2020#technology-web-frameworks-all-respondents2>



Comparaison



	Flask	Django
Open source MIT	~ Similaire (BSD 3-Clause revised)	~ Similaire (Attribution-ShareAlike)
Contributeurs	600+	2000+
Support officiel	The Pallets organization	Django Software Foundation
Courbe d'apprentissage	Facile	Intermédiaire
Vitesse développement	Rapide	Moyen
« Data-binding »	Un sens	Deux sens
Support MVC	MVC	MVT (controller en charge par le framework)
Rendue	-	-
DOM	-	-
Potentiel de proportion	Faible	Haut
Développement natif	-, (Kivy pour Python)	

Bibliographie

- <https://palletsprojects.com/p/flask/>
- Données pour la comparaison :
 - <https://github.com/pallets/flask>
 - <https://github.com/django/django>
- Diagramme popularité 2020 (stackoverflow) :
 - <https://insights.stackoverflow.com/survey/2020#technology-web-frameworks-all-respondents2>