

# Veille Flask 2025 - Jupyter Notebook

=====

## 1. Introduction à Flask

=====

""" Flask est un micro-framework open-source pour le développement web en Python.

- Minimalisme et Extensibilité : noyau léger, extensions possibles pour ajouter authentification, bases de données, etc.
- Composants Clés : Werkzeug (WSGI) et Jinja2 (moteur de template HTML).
- Accessibilité : idéal pour débutants, démarrage rapide.

Origine et Statut :

- Créateur : Armin Ronacher
- Lancement : 1er avril 2010
- Popularité : >68k étoiles sur GitHub en 2025
- Version actuelle : 3.1.2 (19 août 2025) """

=====

## 2. Comparaison avec d'autres frameworks

=====

```
In [1]: import pandas as pd

frameworks = {
    "Caractéristique": ["Philosophie", "Performance", "Courbe d'apprentissage",
    "Flask": ["Micro-framework, flexible", "Moyenne (WSGI)", "Facile pour débuta
    "Django": ["Full-stack, batteries incluses", "Plus lent (ORM lourd)", "Abrup
    "FastAPI": ["Moderne, typage Pydantic, auto-docs", "Très rapide (ASGI asynch
}

df = pd.DataFrame(frameworks)
df
```

Out[1]:

|   | Caractéristique        | Flask   | Django                           | FastAPI                               |
|---|------------------------|---|----------------------------------|---------------------------------------|
| 0 | Philosophie            | Micro-framework, flexible                       | Full-stack, batteries incluses   | Moderne, typage Pydantic, auto-docs   |
| 1 | Performance            | Moyenne (WSGI)                                  | Plus lent (ORM lourd)            | Très rapide (ASGI asynchrone)         |
| 2 | Courbe d'apprentissage | Facile pour débutants                           | Abrupte                          | Excellente expérience dev             |
| 3 | Cas d'usage            | Prototypage rapide, APIs simples, microservices | Sites complexes, dashboards, CMS | Microservices, APIs temps réel, IA/ML |

=====

### 3. Capacités, Requêtes et Extensions

=====

"" Types de requêtes HTTP : GET, POST, PUT, DELETE via @app.route() Requêtes SQL / ORM SQLAlchemy 2.0 : select() + Session.scalars() Vues asynchrones : async def + await

Extensions courantes :

- Bases de données : Flask-SQLAlchemy, Flask-Migrate
- APIs : Flask-RESTful
- Sécurité : Flask-Login, Flask-Security, Flask-WTF
- Déploiement : Flask-Caching, Docker ""

=====

### 4. Environnements d'utilisation

=====

""

- Développement local : prototypage rapide, test de fonctionnalités
- Production : microservices, applications web légères, APIs REST
- Conteneurisation : Docker + docker-compose pour cohérence et isolation
- Contextes pédagogiques : idéal pour l'apprentissage du web Python ""

=====

## 5. Installation et commandes de base

=====

"""

### Installer Flask

```
pip install flask
```

### Vérifier la version

```
flask --version
```

### Créer un projet Flask (exemple minimal)

```
mkdir mon_projet_flask cd mon_projet_flask
```

### Créer app.py

```
echo "from flask import Flask\napp = Flask(name)\n\n@app.route('/')\ndef hello():\n    return 'Hello, Flask!'" > app.py
```

### Lancer le serveur en développement

```
export FLASK_APP=app.py # Linux/Mac set FLASK_APP=app.py # Windows flask run ""
```

=====

## 6. Vulnérabilités et bonnes pratiques

=====

""""

- Breaking changes : vérifier compatibilité des extensions (ex: Flask-Login)
- CVE-2025-47278 : SECRET\_KEY\_FALLBACKS corrigé en 3.1.1
- Sécurité : HTTPS, secrets via variables d'environnement
- Logging et gestion des erreurs
- Docker : gérer dépendances et mots de passe """"

=====

## 7. Ressources et Liens

=====

```
In [2]: ressources = [  
    {"Nom": "Documentation Flask", "Lien": "https://flask.palletsprojects.com"},  
    {"Nom": "Flask-SQLAlchemy Queries", "Lien": "https://flask-sqlalchemy.palletsprojects.com/en/latest/queries/"},  
    {"Nom": "Flask 3.1.1 Release Notes", "Lien": "https://github.com/pallets/flask/releases/tag/3.1.1"},  
    {"Nom": "Flask-RESTful", "Lien": "https://flask-restful.readthedocs.io/en/latest/"},  
    {"Nom": "Flask HTTPAuth", "Lien": "https://flask-httpauth.readthedocs.io/en/latest/"},  
    {"Nom": "Docker Docs", "Lien": "https://docs.docker.com/get-started/"},  
    {"Nom": "Jinja2 Documentation", "Lien": "https://jinja.palletsprojects.com/en/latest/"},  
    {"Nom": "FastAPI Documentation", "Lien": "https://fastapi.tiangolo.com/"}  
]  
  
pd.DataFrame(ressources)
```

Out[2]:

|   | Nom                       | Lien  |
|---|---------------------------|---|
| 0 | Documentation Flask       | https://flask.palletsprojects.com                               |
| 1 | Flask-SQLAlchemy Queries  | https://flask-sqlalchemy.palletsprojects.com/en/latest/queries/ |
| 2 | Flask 3.1.1 Release Notes | https://github.com/pallets/flask/releases/tag/3.1.1             |
| 3 | Flask-RESTful             | https://flask-restful.readthedocs.io/en/latest/                 |
| 4 | Flask HTTPAuth            | https://flask-httpauth.readthedocs.io/en/latest/                |
| 5 | Docker Docs               | https://docs.docker.com/get-started/                            |
| 6 | Jinja2 Documentation      | https://jinja.palletsprojects.com/en/latest/                    |
| 7 | FastAPI Documentation     | https://fastapi.tiangolo.com/                                   |