

# Comparison of Django and Bottle

## Group: The Beauties

Group leader: Wang Jiayu

Members: Wang Shuyi, Wang Xiaoxuan, Wang Yiran, Wang Yuqi, Yang Wenzheng,

Wang Xingzhu, Shi Rongqing, Liu Xuehan, Hu Peiyan, Fan Xiaoying

### Introduction



Django is an open source Python web framework that encourages rapid development. It was originally developed to manage some **news-focused websites** under the Lawrence Publishing Group, which is CMS (Content Management System) software. It is based **model-template-view (MTV)** design pattern. It follows a **"batteries included"** philosophy and ships with many tools that are needed by application developers such as ORM framework, admin panel, directory structure and more.



Bottle is a fast, simple and lightweight WSGI micro web-framework for Python. It is distributed as a single file module and has no dependencies other than the Python Standard Library. The Bottle framework can be roughly divided into the following parts: **Routing, Templates, Utilities, Server.**

### Disadvantages

#### Django:

- **High integration**  
Django also includes some functional modules that are not needed for lightweight applications. For lightweight development process, it is not as light as flask.
- **Over encapsulation**  
Django encapsulates many classes and methods. It can't see the internal code directly. It's easy to use directly, but it's difficult to change.
- **Performance disadvantage**  
Compared with the performance of C and C + +, Django has low performance. Of course, this is the impact of Python language. Other Python frameworks will have the same problem when the traffic increases to a large enough size.
- **Template problem**  
Django's template realizes complete separation of code and style, and does not allow Python code in the template, which reduces the modifiability of the template. For some programmers, flexibility may not be enough.

#### Bottle:

- **Lack of functions**  
One consequence of bottle minimalism is that some functions don't exist at all. Form validation, including CSRF protection, is not supported. If you want to build web applications that support a high level of user interaction, you need to add them yourself.

### Advantages

#### Django

- Django takes care of much of the hassle of Web development.It's free and open source.
- **Ridiculously fast**  
Django was designed to help developers take applications from concept to completion as quickly as possible.
- **Fully loaded**  
Django includes dozens of extras you can use to handle common Web development tasks.
- **Reassuringly secure**  
Django takes security seriously and helps developers avoid many common security mistakes.
- **Exceedingly scalable**  
Some of the busiest sites on the planet use Django's ability to quickly and flexibly scale to meet the heaviest traffic demands.
- **Incredibly versatile**  
Django can be used to build all sorts of things — from content management systems to social networks to scientific computing platforms.

#### Bottle

- **compact and concise**  
Bottle can be considered as a mini flask because it is more compact and concise than other micro web-framework.
- **fast and lightweight**  
It is distributed as a single file module and has no dependencies other than the Python Standard Library. Bottle does not depend on any external libraries.We can just download bottle.py int our project directory and start coding.
- **flexible and supports a wide range of types**  
In pure WSGI, the range of types you may return form your application is very limited. But Bottle is much more flexible and supports a wide range of types. It even adds a Content-Length header if possible and encodes Unicode automatically

### Comparison

Compare of Django and Bottle

#### Django

Python all-inclusive, megaframework for building web application.

Adrian Holovaty and Simon Willison.

Django expects things to be done in a certain way which have no opinion on how developers structure things.

ORM is one of the best features of Django, loved by developers.

REST development is supported via the popular and active Django REST Framework project.

When vulnerabilities are discovered, the Django team has an excellent security policy and fixes are released quickly.

Very many users, very popular.

Not as fast as compared to bare-bones Flask or other microframeworks, but for many real-world use cases, the difference is negligible.

Has a learning curve especially for those who are not familiar with other web frameworks.

#### Bottle

Python microframework for building Web applications..

Marcel Hellkam p.

Very flexible and extremely simple. Doesn't force anything on developers.Just a .py file.

Bottle has no built-in ORM framework. It leaves it up to developers to choose a library like SQLAlchemy or extensions such as Macaron.

Bottle has no built-in support but can be implemented easily.

Bottle has no built-in protection. Bottle is a minimalist framework. It must be handled by developers themselves or by using 3rd party extensions.

In terms of quantity, the number of projects using Bottle is smaller than that using Django.

The performance of the Bottle is very fast.But extensions could have an adverse impact on the performance.But due to its minimal footprint, bottle is ideal for inclusion in other projects or for small projects such as rapid delivery of rest APIs.

Straightforward and easy to learn. It has a good amount of tutorials online.

### Summary



VS



In fact, bottle is lighter, smaller, easier to get started and accepted by beginners. Bottle does not depend on any third-party python module and is easy to deploy. It has high performance and built-in basic needs for web development request, etc.). In contrast, Django is more suitable for larger projects, which require a lot of back-end and front-end support, or when time plays a key role,for Django provides a large number of ready-made components.

### References

<https://blog.csdn.net/happyteafriends/article/details/42552093>  
<https://www.cnblogs.com/wupeiqi/articles/5341480.html>  
[https://blog.csdn.net/zandaoguang/article/details/77387358?utm\\_medium=distribute.pc\\_relevant.none-task-blog-BlogCommendFromBaidu-2.nonecase&depth\\_1-utm\\_source=distribute.pc\\_relevant.none-task-blog-BlogCommendFromBaidu-2.nonecase](https://blog.csdn.net/zandaoguang/article/details/77387358?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-2.nonecase&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-2.nonecase)

<http://www.bottlepy.org/docs/dev/tutorial.html#request-routing>  
<https://www.jianshu.com/p/8590665a5eb5>  
<https://blog.csdn.net/JackLiu16/article/details/81611979>  
<https://blog.csdn.net/tyfbhlxd/article/details/72123854>