django

# DJANGO

# FRAMWORK

(Programming Language)

Prepared By

Abdedaiem Amin

16 APRIL 2021

# Meet Django !

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

# Aim and Objectives:

In This Section We Are going to cover the why’s and how’s of Django being the valid option to Maintain in your next or Current project … !

1. Ridiculously fast:

   Django was designed to help developers take applications from concept to completion as quickly as possible.

2. Reassuringly secure:

   Django takes security seriously and helps developers avoid many common security mistakes.

3. Exceedingly scalable:

   Some of the busiest sites on the Web leverage Django’s ability to quickly and flexibly scale.

# Why Django?

With Django, you can take Web applications from concept to launch in a matter of hours. Django takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

## Ridiculously fast:

Django was designed to help developers take applications from concept to completion as quickly as possible.

**Wanna see how fast ? ? ?**

## Quick install guide:

Before you can use Django, you'll need to get it installed. this guide will guide you to a minimal installation that'll work while you walk through the introduction.

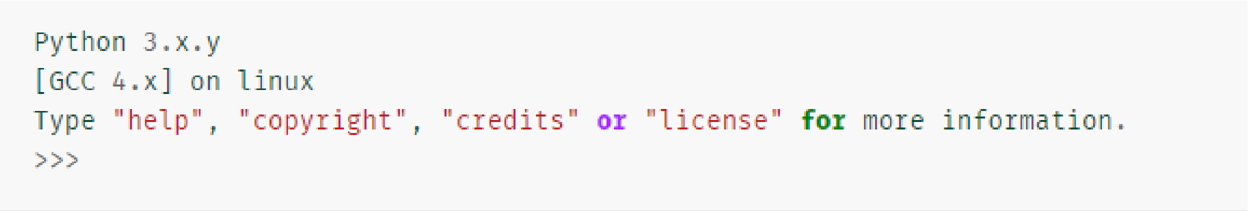
## Install Python

Being a Python Web framework, Django requires Python. Python includes a lightweight database called SQLite so you won't need to set up a database just yet.

Get the latest version of Python at https://www.python.org/downloads/ or with your operating system's package manager.(Unix based OS's users)

You can verify that Python is installed by typing **python** from your shell (CMD for windows users); you should see something like:

Python 3.x.y



```
Python 3.x.y
[GCC 4.x] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

MariaDB, MySQL, or Oracle. To install such a database, consult the

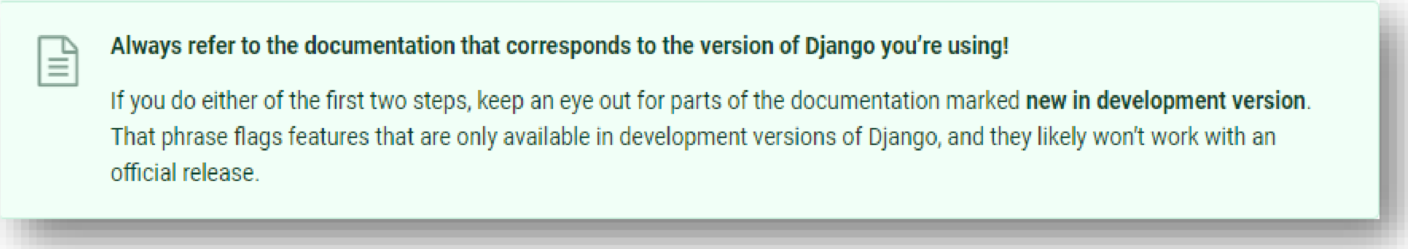
## Install Django

You've got three options to install Django:

- Install an official release. This is the best approach for most users.
- Install a version of Django provided by your operating system distribution.
- Install the latest development version. This option is for enthusiasts who want the latest-and-greatest features and aren't afraid of running brand new code. You might encounter new bugs in the development version, but reporting them helps the development of

Django. Also, releases of third-party packages are less likely to be compatible with the development version than with the latest stable release.

**NOTE!**

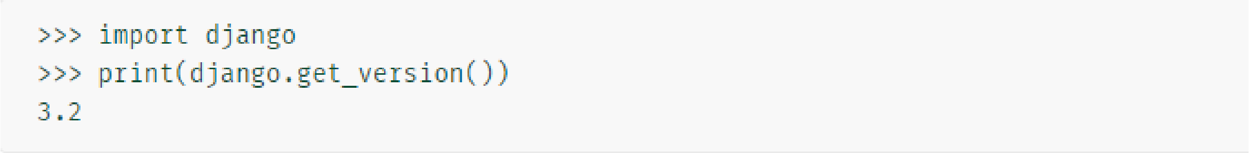


**Always refer to the documentation that corresponds to the version of Django you're using!**

If you do either of the first two steps, keep an eye out for parts of the documentation marked **new in development version**. That phrase flags features that are only available in development versions of Django, and they likely won't work with an official release.

## Verifying

To verify that Django can be seen by Python, type python from your shell. Then at the Python prompt, try to import Django:



```
>>> import django
>>> print(django.get_version())
3.2
```

## That's it!

## Fully loaded:

Django includes dozens of extras you can use to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds, and many more tasks — right out of the box.

## Reassuringly secure:

Django takes security seriously and helps developers avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery and clickjacking. Its user authentication system provides a secure way to manage user accounts and passwords.

## 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.

### Sites Using Django

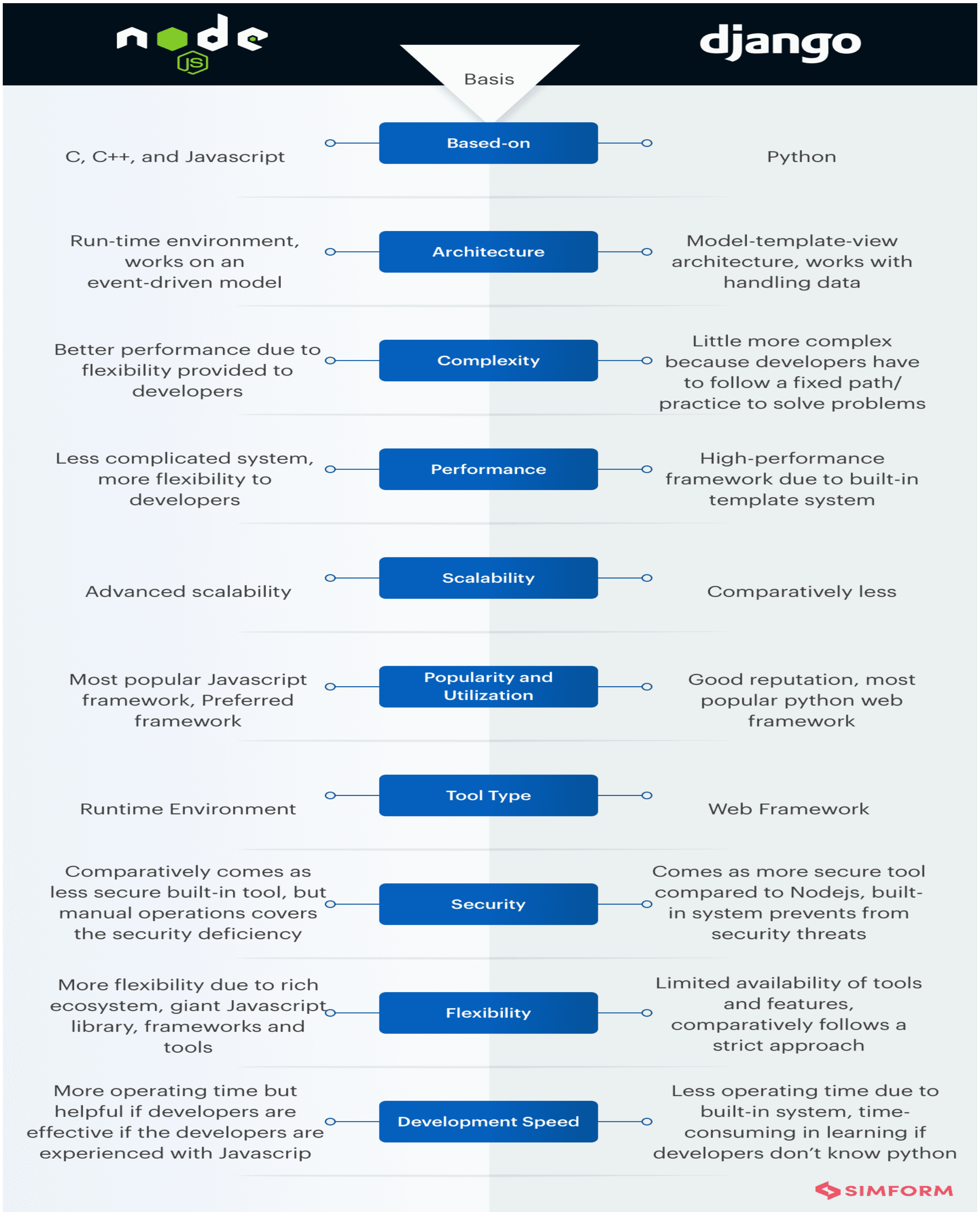
- Disqus
- Instagram
- Knight Foundation
- MacArthur Foundation
- Mozilla
- National Geographic
- Open Knowledge Foundation
- Pinterest
- Open Stack

## Incredibly versatile:

Companies, organizations and governments have used Django to build all sorts of things — from content management systems to social networks to scientific computing platforms.

# DJANGO vs NODE.JS(JavaScript)

Before we compare the two technologies, you must understand the key features that set them apart.



| node JS | Basis | django |
|---|---|---|
| C, C++, and Javascript | Based-on | Python |
| Run-time environment, works on an event-driven model | Architecture | Model-template-view architecture, works with handling data |
| Better performance due to flexibility provided to developers | Complexity | Little more complex because developers have to follow a fixed path/ practice to solve problems |
| Less complicated system, more flexibility to developers | Performance | High-performance framework due to built-in template system |
| Advanced scalability | Scalability | Comparatively less |
| Most popular Javascript framework, Preferred framework | Popularity and Utilization | Good reputation, most popular python web framework |
| Runtime Environment | Tool Type | Web Framework |
| Comparatively comes as less secure built-in tool, but manual operations covers the security deficiency | Security | Comes as more secure tool compared to Nodejs, built-in system prevents from security threats |
| More flexibility due to rich ecosystem, giant Javascript library, frameworks and tools | Flexibility | Limited availability of tools and features, comparatively follows a strict approach |
| More operating time but helpful if developers are effective if the developers are experienced with Javascrip | Development Speed | Less operating time due to built-in system, time-consuming in learning if developers don't know python |

SIMFORM

# Django

Much like Node.js, Django can be learned quickly. Moreover, if you're already familiar with Python, this framework will only come naturally to you. However, to work with Django, you'll need to understand the methods and features that come out-of-the-box and the framework's MTV architecture.

The framework follows a strict approach to following them throughout the development. Additionally, it allows you to do more work with less code, saving you a lot of time. Django offers excellent documentation and a community to help you through the hiccups.

# Node.js

Node.js architecture is distinct among other server-side platforms; its single-thread event loop architecture helps build high-end applications. This server-side runtime environment offers robustness to the applications built with it, making them available for processing requests without delays because of its non-blocking input/output and asynchronous request handling architecture.

# *Pros & Cons of Django Framework*

## ➢ PROS

- ### Django accelerates the development of large projects

Django demonstrates certain flexibility in large projects, allowing you to simultaneously create parallel modules, thereby speeding up the development process several times

- ### Django focuses on safety

Django becomes an excellent framework for large projects that intend to work with large amounts of sensitive data. A project built on Django remains secure even after receiving a massive influx of traffic.

- ### Django provides easy API creation

When it is impossible to use an API created by someone else, the developer must create their own interfaces. With Django, this disadvantage quickly turns into an advantage. For easy API creation, you can use the Django Rest framework, which allows you to utilize powerful API functionality

- ### Django is ideal for machine learning projects

Machine learning technologies are at their peak, and Django is excellent for building ML-powered platforms. For such projects, Django simplifies the development process thanks to its built-in computation and mathematical and statistical capabilities , apart from it being built with python

- ### Django provides smooth work with databases

As we have said, Django has high-end computations and statistical capabilities. And working with databases is one more situation where Django is incredibly useful. What is more, there are significant advantages to using Django forms when working with databases. These forms allow the developer to easily transfer the data from a form to a database table.

## ➢ Cons

- ### Thorny study

  Despite the simplicity of Python, the developers say that Django is quite tricky to master. Although Django was created for Python web projects, Python's knowledge is not synonymous with Django.

- ### Django is not suitable for small projects

  Can this be considered one of the significant Django drawbacks? Most likely, this is just a feature that you need to take into account. Django is not the best choice for small projects, as its primary goal is to speed up the development of large projects that will scale in the future.

- ### Django is monolithic

  This framework is a huge piece of software with the same huge number of dependencies. That is why this framework's adherents have already created many models suitable for reuse in different projects, thereby rendering the creators of Django a disservice

# Conclusion

As you can see, Django's advantages prevail over its disadvantages. It is an ideal framework for creating large projects that work with large amounts of data, for example, applications based on machine learning, regression analysis, and predictive analytics. What's more, Django's built-in security features keep the project secure from fraudulent attempts after launch, and its excellent scaling capabilities allow a vast number of users to use your solution without compromising their experience.

## ***LIST OF SOURCES :***

https://www.simform.com/nodejs-vs-django/

https://codecondo.com/pros-cons-of-django-framework/

https://www.djangoproject.com

https://canvas.hull.ac.uk/courses/367/pages/report-structure

**THANK YOU!**