

PYTHON CASE STUDY



Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

Cognitive Convergence is Subject Matter Expert in Office 365, Dynamics 365, SharePoint, Project Server, Artificial Intelligence Practices, Machine Learning practices, Power Platform: Power Apps-Power BI-Power Automate-Power Virtual Agents.

Our Python Development, Consulting, Add-in Development, Customization, Integration services and solutions are dedicated to help companies maximize their business performance, overcoming market challenges, achieving profitability, and providing best customer service.

Contents

OBJECTIVE	1
WHAT IS PYTHON?	1
<i>History of Python.....</i>	<i>1</i>
PYTHON FEATURES.....	1
TYPES OF PYTHON FRAMEWORKS.....	2
1. Full-Stack Framework.....	2
2. Microframework.....	3
3. Asynchronous Framework.....	3
Best Python Frameworks	3
1. AIOHTTP	3
2. Bottle.....	3
3. CherryPy	4
4. CubicWeb	4
5. Dash.....	4
6. Django.....	5
7. Falcon	5
8. Flask	6
9. Giotto	6
10. Growler.....	7
11. Hug	8
12. MorePath	8
13. Pycnic.....	8
14. Pylons Framework	9
15. Pyramid	9
16. Sanic.....	9
17. Tornado	10
18. TurboGears.....	10
19. Web2Py	11
PROS OF PYTHON FRAMEWORK FOR BUSINESS APPLICATION DEVELOPMENT	11
1. Python Encourages Faster Project Development	12
2. It Provides an Ideal Platform for Web Development.....	12
3. Develop Portable Codes Using Python	12
4. Python Supports Prototype and MVP Development.....	13
5. Keep Your Data Safe and Secure	13
6. Availability of Multiple Frameworks for Better UI Experience.....	13
7. Develop Your IoT Visions with Python	14
8. Save Money with Python.....	14
9. Leverage Test-Driven Development.....	14
10. Best Community Support System	15
CONS OF PYTHON FRAMEWORK FOR BUSINESS APPLICATION DEVELOPMENT	15
1. Poor Memory Efficiency.....	15
2. Slow Speed	15
3. Database Access.....	15
4. Weak in Mobile Computing.....	15
5. Runtime Errors.....	15
BUILDING CUSTOM MIDDLEWARE	15
DEPLOYMENT OF PYTHON PROJECT	16

<i>No-impact applications.</i>	16
<i>Private components by default.</i>	16
<i>Controlled code sharing.</i>	16
<i>Side-by-side versioning.</i>	16
<i>XCOPY deployment and replication.</i>	16
<i>On-the-fly updates.</i>	16
<i>Integration with the Windows Installer.</i>	17
<i>Enterprise deployment.</i>	17
<i>Downloading and caching.</i>	17
<i>Partially trusted code.</i>	17
IMPLEMENTATIONS	18
<i>Reference implementation</i>	18
<i>Other implementations</i>	18
<i>Unsupported implementations.</i>	18
<i>Cross-compilers to other languages</i>	19
<i>Performance.</i>	19
BEST FRONT-END FRAMEWORKS	19
React	19
<i>Pros.</i>	20
<i>Cons.</i>	20
<i>When to use:</i>	20
<i>When not to use:</i>	20
Angular	20
<i>Pros.</i>	21
<i>Cons.</i>	21
<i>When to use:</i>	21
<i>When not to use:</i>	21
Vue.js	21
<i>Pros.</i>	21
<i>Cons.</i>	22
<i>When to use:</i>	22
<i>When not to use:</i>	22
Ember.js	22
<i>Pros.</i>	23
<i>Cons.</i>	23
<i>When to use:</i>	23
<i>When not to use:</i>	23
jQuery	23
<i>Pros.</i>	24
<i>Cons.</i>	24
<i>When to use:</i>	24
<i>When not to use:</i>	24
Semantic-UI	24
<i>Pros.</i>	24
<i>Cons.</i>	24
<i>When to use:</i>	25
<i>When not to use:</i>	25
Backbone.js	25
<i>Pros.</i>	25

Cons.....	25
When to use:	25
When not to use:	25
Preact.....	26
Pros.....	26
Cons.....	26
When to use:	26
When not to use:	26
Svelte.....	26
Pros.....	26
Cons.....	27
When to use:	27
When not to use:	27
Foundation	27
Pros.....	27
Cons.....	27
When to use:	28
When not to use:	28
LIBRARIES	28
PYTHON DEVELOPMENT ENVIRONMENT.....	29
Text Editors.....	29
Vim	29
Python-mode.....	29
SuperTab	29
Emacs	29
TextMate.....	29
Sublime Text.....	30
Atom.....	30
IDEs	30
PyCharm / IntelliJ IDEA.....	30
Python (on Visual Studio Code)	30
Enthought Canopy.....	30
Eclipse.....	30
Komodo IDE.....	30
Spyder.....	30
WingIDE.....	31
NINJA-IDE	31
Eric (The Eric Python IDE)	31
Interpreter Tools.....	31
Virtual Environments.....	31
pyenv.....	31
Other Tools	32
IDLE	32
IPython	32
BPython	32
ptpython.....	32
DATABASE WITH PYTHON	33
Microsoft SQL Server	33
Oracle	33

<i>MySQL</i>	34
<i>PostgreSQL</i>	34
<i>SQLite</i>	34
<i>Firebird</i>	35
USE CASES OF PYTHON	35
1. <i>Web applications</i>	35
2. <i>Software development</i>	35
3. <i>Data science</i>	35
4. <i>AI applications</i>	36
5. <i>Game development</i>	36
INDUSTRIES USE PYTHON IN THEIR BUSINESS	36
<i>Google:</i>	36
<i>Facebook:</i>	36
<i>Uber:</i>	37
<i>Spotify:</i>	37
<i>Instagram:</i>	38
<i>Dropbox:</i>	38
<i>Quora:</i>	38
<i>Netflix:</i>	39
WHY CHOOSE PYTHON?	39
CONCLUSION	40

OBJECTIVE

The objective of this document is to describe Python and its different modules. We will elaborate on the use of Python in various industries and how we will provide you more advance, secure web applications based on your requirement.

WHAT IS PYTHON?

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.



- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

- Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.
- Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.
- Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).
- Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

PYTHON FEATURES

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.

- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Leverage our expert Python services to harness
the potential of futuristic technology

Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

TYPES OF PYTHON FRAMEWORKS

1. Full-Stack Framework

Such frameworks are a one-stop solution for all developer requirements. Form generators, form validation, and template layouts are usually available with a typical full-stack framework.

2. Microframework

These are lightweight frameworks that don't offer additional functionalities and features, such as database abstraction layer, form validation, and specific tools and libraries. Developers using a microframework need to add a lot of code and additional requirements manually.

3. Asynchronous Framework

Gaining popularity recently, any asynchronous framework is a microframework that allows for handling a large set of concurrent connections. Typically, an asynchronous framework built for Python uses the programming language's `asyncio` library.

Best Python Frameworks

1. AIOHTTP

Type – Asynchronous framework AIOHTTP is a Python framework that relies heavily on Python 3.5+ features, such as `async` & `await`s. The Python framework makes use of Python's `asyncio` library, and is hence an asynchronous framework. In addition to being a server web framework, AIOHTTP can also serve as a client framework. It provides a request object and router to enable the redirection of queries to functions developed to handle the same.

Key highlights: -

- Allows effectively building the views
- Middlewares support
- Pluggable routing
- Signals
- Supports both Client WebSockets and Server WebSockets without the Callback Hell



2. Bottle

Type – Microframework Bottle creates a single source file for every application developed using it. It is one of the best python web framework. The microframework for Python was originally developed for building APIs. Other than the Python Standard Library, Bottle has no dependencies required for crafting small web applications. One of the most important advantages of using Bottle is that it allows developers to work closer to the hardware. In addition to building simplistic personal-use apps, Bottle is an apt fit for learning the organization of web frameworks and prototyping.



Key highlights: -

- Adapter support for 3rd-party template engines and WSGI/HTTP servers
- Allows simple access from cookies, data, file uploads, and other HTTP-related metadata
- Built-in HTTP server
- Plugin support for different databases
- Provides request-dispatching routes having URL-parameter support

3. CherryPy

Type – Microframework CherryPy is a popular open-source, object-oriented Python framework that follows a minimalistic approach. The micro-framework is one of the oldest Python frameworks, debuting in the June of 2002.

Any CherryPy-powered web application is a standalone Python application with its own embedded multi-threaded web server and runs on any OS with support for Python. Such an app can be deployed anywhere an ordinary Python app is able to run.

There is no need for an Apache server for running apps developed using CherryPy. The micro-framework allows the developer(s) to make use of any type of technology for data access, templating, et cetera.



Key highlights: -

- A number of out-of-the-box tools for authentication, caching, encoding, sessions, static content, and much more
- A flexible built-in plugin system
- HTTP/1.1-compliant WSGI thread-pooled web server
- Inbuilt support for coverage, profiling, and testing
- Offers simplicity for running multiple HTTP servers simultaneously
- Powerful configuration system
- Runs on Android

4. CubicWeb

Type – Full-stack framework Developed and curated by Logilab, CubicWeb is a free-to-use, semantic, open-source, Python-based web framework. Based on the data model, CubicWeb requires to have the same defined in order to develop a functional application.



Unlike other popular Python frameworks that use separate views and models, CubicWeb makes use of cube. Multiple cubes are then joined together for creating an instance with the help of a database, a web server, and some configuration files.

Key highlights: -

- OWL (Web Ontology Language) and RDF (Resource Description Framework) support
- Reusable components
- Security workflows
- Simplifies data-related queries with RQL (Relational Query Language) embedding
- Support for multiple databases

5. Dash

Type – Microframework Dash is an open-source Python-based framework for building analytical web applications. It is an ideal Python framework for data scientists that aren't much into the mechanics of web development.

Applications designed with Dash are web servers that run Flask and communicate with JSON packets via HTTP



requests. Their frontend renders components using ReactJS. Flask plugins can be used for extending the capabilities of Dash.

Because Dash apps are rendered in the web browser and can be deployed to servers, they are cross-platform and mobile-ready. Dash developers are provided access to the underlying Flask instance as well as all of its configurable properties.

Key highlights: -

- Dash apps demand very little boilerplate code for getting started
- Error handling (Dash Deployment Server)
- High-degree of customization
- LDAP integration (Dash Deployment Server)
- Plugin support
- Simple interface for tying UI controls, including dropdowns, graphs, and sliders
- URL routing (Dash Deployment Server)

6. Django

Type – Full-stack framework Django is one of the most beloved web development frameworks for developing Python applications. In fact, it ended up being one of the best web development frameworks. Django framework follows the DRY (Don't Repeat Yourself) principle.

Unlike others, the free-to-use and open-source full-stack Python framework include a grand number of built-in features rather than offering them as individual libraries. Django makes use of its ORM for mapping objects to database tables.

This results in allowing the code to work across different databases as well as making it easier to migrate from one database to the other. Though Django has inherent support for MySQL, PostgreSQL, SQLite, and Oracle Database, it can support other databases via third-party drivers.

Key highlights: -

- A plethora of ready-to-use libraries
- Authentication support
- Database schema migrations
- Object-relational mapper (ORM)
- Support for web servers
- Template engine
- URL routing



7. Falcon

Type – Microframework Aimed at rapidly building web APIs, Falcon is another widely used Python framework. Unlike other Python frameworks that require loading a lot of dependencies for building HTTP APIs, Falcon allows developers to build a cleaner design that enables HTTP and REST architectures.



As per the benchmark test conducted by Sanic, Falcon is able to handle most requests with the same hardware than all its contemporaries. The Python framework aims to have 100% code coverage. Falcon is used by big players like LinkedIn, OpenStack, and RackSpace.

Key highlights: -

- An extensible, highly-optimized code base
- DRY request processing through middleware components and hooks
- Ease of access for headers and bodies via request and response classes
- Extra speed boost with Cython support
- Idiomatic HTTP error responses
- REST-inspired resource classes and URI templates offer intuitive routing
- Unit testing via WSGI helpers and mocks
- Upfront exception handling

8. Flask

Type – Microframework Available under the BSD license, Flask is another popular Python framework. Inspired by the Sinatra Ruby framework, the microframework requires Jinja2 template and Werkzeug WSGI toolkit. Thanks to its lightweight and modular design, Flask is readily adaptable.



Flask allows the developers to build a solid web application foundation from where it is possible to use any kind of extensions required. The micro-framework is compatible with Google App Engine.

Key highlights: -

- Built-in fast debugger
- HTTP request handling
- Inbuilt development server
- Jinja2 templating
- RESTful request dispatching
- Support for plugging in any ORM
- Supports secure cookies to establish client-side sessions
- Unicode-based
- Unit testing support
- WSGI 1.0 compliance

9. Giotto

Type – Full-stack framework Based on the Model View Controller pattern, Giotto is an application framework for Python. In order to allow web designers, web developers, and system admins to work independently, Giotto separates Model, View, and Controller elements in order.



Giotto includes controller modules that enable users to create apps on top of the web, IRC (Internet Relay Chat), and command line.

Key highlights: -

- Automatic URL routing
- Database persistence with SQLAlchemy
- Extremely succinct code
- Functional CRUD patterns
- Generic models and views
- Inbuilt cache with support for Memcache and Redis (Available API for extending support for other engines)
- Jinja2 for HTML templates (API available for supporting other template engines)
- Multiple pluggable controllers
- RESTful interface with normal “browser POST” CRUD site

10. Growler

Type – Asynchronous framework Inspired by the NodeJS and the Express/Connect frameworks, Growler is a micro web framework written atop the Python’s asyncio library.

Unlike other conventional Python frameworks, requests in Growler aren’t handled in the framework but by passing through middleware technology.

A top choice among Python frameworks for easily and quickly implementing complex applications, Growler was originally developed by its author to simply learn how to use asyncio library at its lowest levels.



Key highlights: -

- Easy to see program flow due to lack of required callbacks and proper try/except blocks
- Support for a multitude of open-source packages
- Use decorators for writing clean, reusable code
- Ziapp module allows zipping an entire application into a single executable file

Our Python development company has a huge team of proficient testers who provide seamless code execution for your apps.

Cognitive Convergence

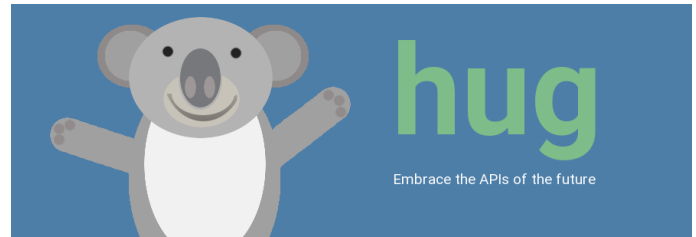
<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

11. Hug

Type – Microframework HUG The Hug is designed to allow Python developers to develop an API once and then use it anywhere they wish. The Python framework simplifies API development by means of offering multiple interfaces. It is labeled as the fastest web framework for Python 3.



Whether you are doing local development or over HTTP or via the CLI, Hug lets you complete application development quickly and easily. In order to take performance to the next level, Hug consumes resources only when required and uses Cython for compilation.

Key highlights: -

- Annotation-powered validation
- Follows write once, use anywhere philosophy
- Inbuilt version management
- Supports automatic documentation

12. MorePath

Type – Microframework Labeled as the “Super Powered Python Web Framework,” MorePath ensures minimal setup footprint. It is designed specifically for getting most of the typical use cases up and running ASAP, including the common Python data structures being induced into RESTful Web Services.

The microframework, MorePath, is a very flexible model-driven web framework. A unique feature available to the Python framework is that it automatically turns the paths defined in the framework into links in creating apps with the cleanest URLs.



Key highlights: -

- All views are generic. Reuse views in views
- Comes with all the necessary tools to develop RESTful web services
- Creating generic user interfaces is as simple as subclassing
- Extensible with a simple, coherent, and universal extension and override mechanism
- Flexible, simple, and powerful permissions

13. Pycnic

Type – Microframework Pycnic is an object-oriented microframework believed to be the fastest for building JSON-based APIs. The small, standalone, and optimized for JSON-based APIs framework is able to hold its ground well among big players. Because Pycnic creates just the Web APIs, it has a minimal footprint and therefore, it is fast.



Key highlights: -

- Built-in error handling
- Capable of handling JSON-based requests
- Handles routing
- Support for cookies

14. Pylons Framework

Type – Full-stack framework Pylons Framework is an open-source Python-based framework that focuses on the rapid development of applications. The framework is designed by incorporating some of the best elements and properties of languages including Perl, Python, and Ruby.



Although available in maintenance mode, several developers still use the Pylons framework due to its ability to offer a highly flexible structure for web development. In order to promote reusability, the full-stack framework makes extensive use of WSGI (Web Server Gateway Interface).

Key highlights: -

- HTML form validation and generation
- Routes
- Text-based templating
- URL dispatch
- URL mapping based on Routes configuration via WebHelpers

15. Pyramid

Type – Full-stack framework The Pyramid is another leading Python framework. The primary goal of the open-source Python-based web development framework is to achieve as much as with minimalistic complexity.



The most desirable aspect of the Pyramid is the ability to work equally well with small as well as full-scale applications. Moreover, there is a multitude of resources available to extend its abilities as per the requirements call for.

Key highlights: -

- Flexible authentication and authorization
- Function decorators
- Predicates
- Renderers
- Single-file applications
- Templating and asset specifications
- URL generation

16. Sanic

Type – Asynchronous framework Built on top of the uvloop, Sanic is a simple and open-source Python framework developed especially for offering fast HTTP responses via asynchronous request handling. Hence, it is an asynchronous framework for Python.



Sanic supports asynchronous request handlers, making it compatible with Python 3.5's `async/await` functions. It results in enhancing speed as well as offering non-blocking capabilities.

During a benchmark test with one process and 100 connections, Sanic was able to handle as much as 33,342 requests in a single second.

Key highlights: -

- Able to read and write cookies
- Allows different types of logging, such as access log and error log
- Class-based views
- Handlers with easy to apply decorator's support
- Plugin support
- Supports blueprints for sub-routing within an application
- The configuration object can be modified either by using dot-notation or like a dictionary

17. Tornado

Type – Asynchronous framework The Tornado is an open-source Python framework and an asynchronous networking library. In addition to solving the C10k issue (which simply means to handle 10k connections at any given time), the asynchronous framework uses a non-blocking network I/O.

The Python framework was originally developed for a company called FriendFeed, which was acquired by Facebook in 2009. The Tornado is an ideal tool for building apps asking for high performance and several thousand concurrent users.



Key highlights: -

- Allows implementation of 3rd-party authentication and authorization schemes
- Offers high-quality output
- Real-time services
- Supports translation and localization
- User authentication support
- Web templating

18. TurboGears

Type – Full-stack TurboGears is a data-driven, open-source, full-stack web application framework for Python. The framework allows developers to rapidly develop extensible data-driven web apps.

In addition to supporting a flexible and powerful ORM, TurboGears come with intelligible templating. The full-stack framework makes use of components such as Genshi, Repoze, SQLAlchemy, and WebOb to easily and quickly develop apps requiring database connectivity.



Key highlights: -

- All features are implemented as function decorators
- Available command-line tools
- MochiKit JavaScript library integration

- Multi-database support
- MVC-style architecture
- PasteScript templates
- ToscaWidgets for simplifying coordination of frontend design and server deployment
- Uses Pylons as a web server
- Validation support with FormEncode

19. Web2Py

Type – Full-stack framework For Python developers looking for a scalable full-stack framework, Web2Py might be the answer. The open-source Python framework comes with its own web-based IDE, which includes a code editor, debugger, and one-click deployment.

Though Web2Py allow users to create dynamic web content in Python, it doesn't provide support for Python 3. The ticketing system is one of the most important features of the Python framework. The system issues a ticket to the user whenever an error occurs.



Key highlights: -

- Ability to run on any web hosting platform that provides support for either Python or Java and Python
- Backward compatibility
- Built-in data security for preventing several common vulnerabilities, including cross-site scripting, injection flaws, and malicious file execution
- Devoid of installation and configuration requirements
- Follows MVC-pattern
- Provides support for internationalization
- Readability of multiple protocols

Our Python experts are skilled in developing highly secure backend platforms for your application.

Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

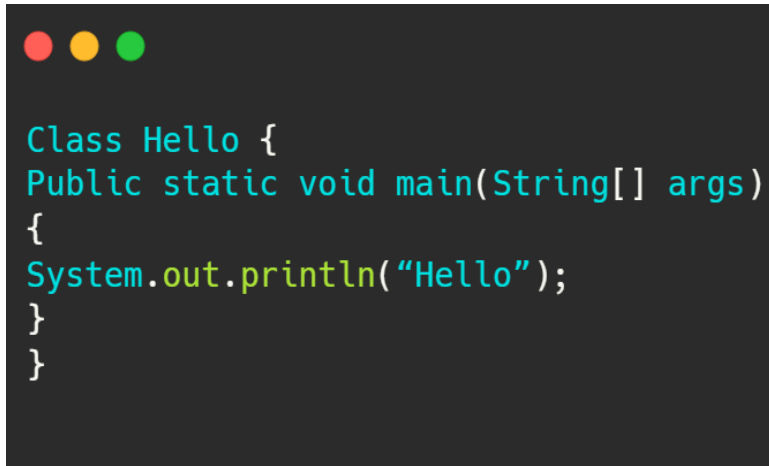
PROS OF PYTHON FRAMEWORK FOR BUSINESS APPLICATION DEVELOPMENT

Versatility is one of the major aspects that modern-day businesses focus on. And when we talk about a programming language, there is hardly any other language as versatile as Python. A few aspects that entice businesses to choose Python development services for product development are as follows:

1. Python Encourages Faster Project Development

There are multiple built-in dictionary data structures in Python that contribute to the faster runtime of the written code. Moreover, the length of code written in Python is way shorter than in any other language.

For instance, if you want to code a simple program to write 'Hello' on your screen. In Java, the code will be written as:



```
Class Hello {  
Public static void main(String[] args)  
{  
System.out.println("Hello");  
}  
}
```

However, in Python, the code can be directly written as:

print 'Hello'

Apart from that, Python has strong process control and text processing capabilities along with an efficient unit testing framework. These abilities account for higher speed and productivity, which in turn helps in the completion of projects more swiftly.

2. It Provides an Ideal Platform for Web Development

In the realm of web development, Python is considered the most preferred language by developers. It is so because the language offers the developers a set of numerous valuable and unique features such as:

- A wide selection of pre-built libraries.
- Built-in framework for unit testing.
- Easy integration capabilities.
- Fast to implement codes.
- Multiple functional frameworks.

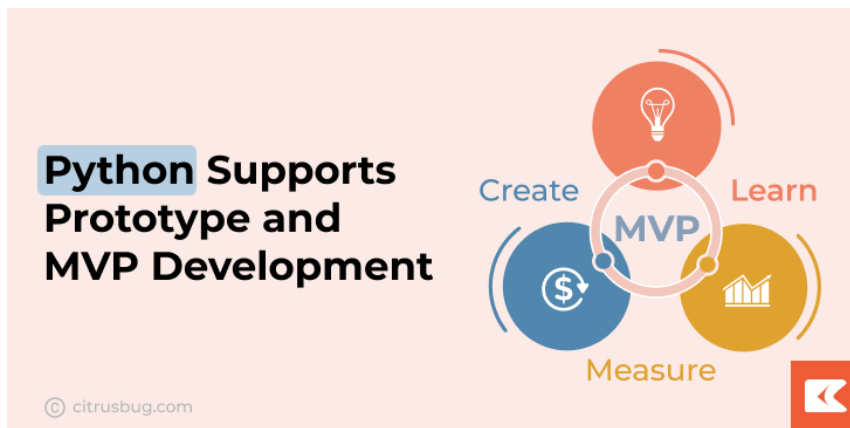
Python helps develop flawless mobile and web applications that accelerate the ROI of commercial projects, making it an ideal language for developing web applications.

3. Develop Portable Codes Using Python

Another reason why Python is suitable for developing projects for business operations is because of its portability. The code written in Python is entirely portable and can be used on different operating systems such as Linux, Unix, Mac, and Windows.

This feature of the language is useful to businesses as you can seamlessly switch to different platforms if required without reinvesting in code regeneration. It also saves immense time for your organization.

4. Python Supports Prototype and MVP Development



Development of prototypes and MVP (Minimum Viable Products) can be highly effective for business operations. They help to check the workability of the developed project and make any changes if required with minimal effort.

With Python development services, building MVP and prototypes of larger projects are extremely simple. Therefore, businesses must consider Python while building their new projects.

5. Keep Your Data Safe and Secure

Data and its security is considered to be one of the highest values for any business. It is estimated that nearly 64% of businesses have experienced web-based attacks to date.

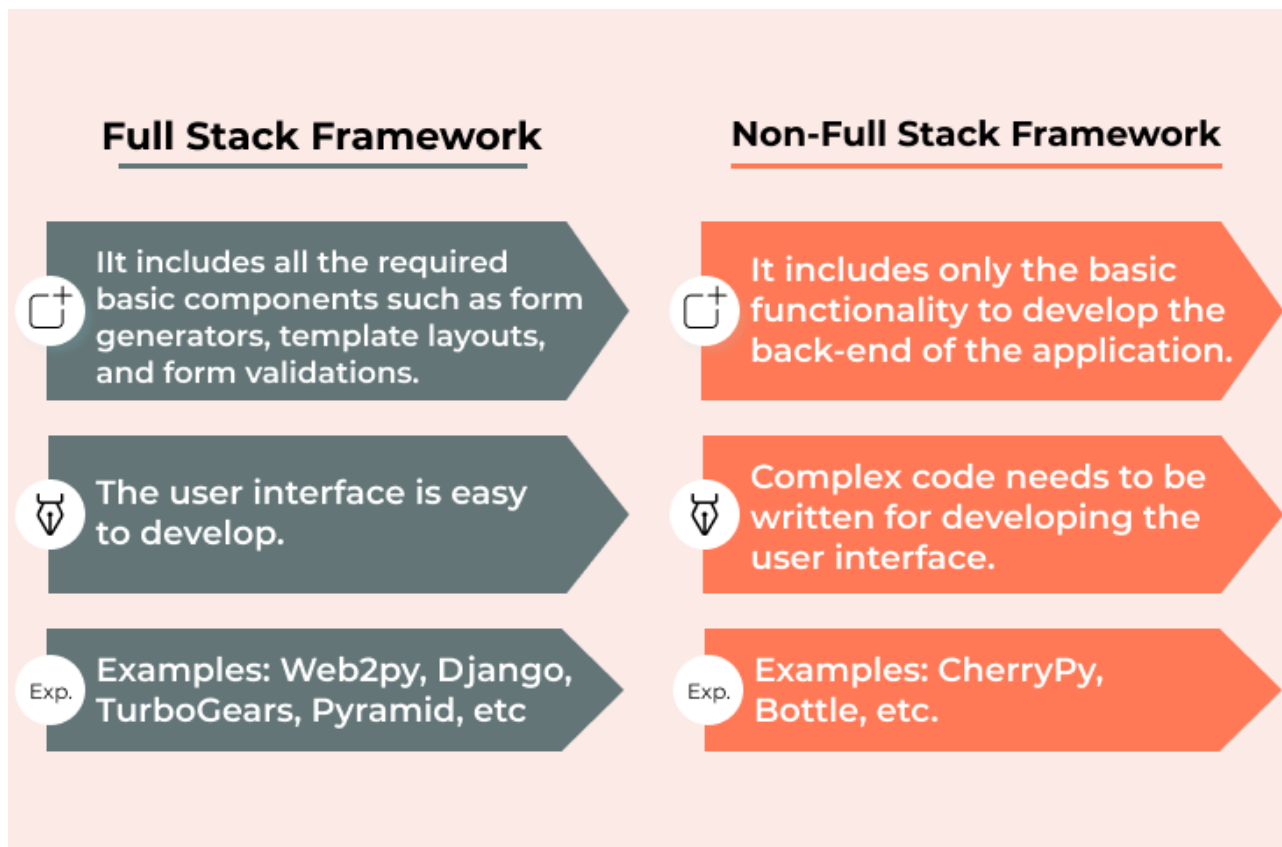
What makes Python safe is that it hides the source code of your program from direct viewing, which makes it harder for hackers to breach the data. This invariably provides enhanced security to your applications and keeps them safe from external attacks.

6. Availability of Multiple Frameworks for Better UI Experience

Python provides a wide scope of frameworks to its developers. Fundamentally, Python has two types of frameworks:

- Full Stack Framework
- Non-Full Stack Framework

Here is the difference between the two:



The multiple frameworks used in Python make the development process easier. It enables the developer to become more efficient and creative in their work. Also, the business client gets a wide range of framework options to select for their project and provide users with greater experience to their application.

7. Develop Your IoT Visions with Python

There is no doubt that the future of technology lies in IoT (Internet of Things). IoT is spreading like wildfire all across the globe because of its countless benefits. Python platforms like Raspberry Pi enable developers to build their own digital devices with ease.

Leveraging the advanced concepts of Python development, connecting with real-world markets for small and independent businesses has become extremely simple.

8. Save Money with Python

Though budget may not be a constraint for large enterprises, for small and medium-sized businesses, every penny counts. As mentioned above, Python uses efficient coding techniques that reduce the time to develop the code. Hence it provides a lower time to market.

The cost of any project is highly dependent on the working hours of a programmer. So if you are a startup with a stringent budget, you can hire Python programmer to get your project built at a reasonable price and get faster results.

9. Leverage Test-Driven Development

Another factor that outshines Python from other programming languages is its test-driven development approach. In Python, the developers can plan and write the test study case before they start coding. This comes into use when the process of code development starts.

The written test cases can be used to test the code and provide results simultaneously, saving plenty of time. Also, the pre-needed set of materials based on the code can be tested using this approach. Various Python frameworks support unit testing, which helps keep bugs out of code and develop robust and efficient code for the program.

10. Best Community Support System

Python is an open-source language. Anybody can learn, develop, and use codes written in Python worldwide. There are over a million programmers who actively work regularly to improve Python's core features and functionalities.

It is one of the biggest reasons for businesses to choose Python. However complex the code may be, the extensive network of the Python community is always there to support developers and provide a feasible solution.

CONS OF PYTHON FRAMEWORK FOR BUSINESS APPLICATION DEVELOPMENT

1. Poor Memory Efficiency

To make it simple for the developer, Python needs a lot of memory space; this can be a tad problematic if you want to develop apps where you need to optimize memory.

2. Slow Speed

After the high memory usage, its lack of speed is one of the biggest disadvantages of Python.

As it executes the code one line at a time, the speed of execution often is hampered. Where speed is important for the project, Python cannot be used for coding.

3. Database Access

Though it is easy to program with Python, the database access layer is underdeveloped compared to other technologies like ODBC.

For enterprise apps, it's imperative that there is hassle-free interaction of complicated legacy data, and hence it is not an ideal language for such apps.

4. Weak in Mobile Computing

Python is used for backed programming; due its high memory usage and slow speed, it is generally not used for frontend programming or mobile app development.

5. Runtime Errors

The data types of variables in Python can change suddenly, as it is a dynamically typed language. A variable holding a string may contain an integer later, and this can lead to runtime errors.

BUILDING CUSTOM MIDDLEWARE

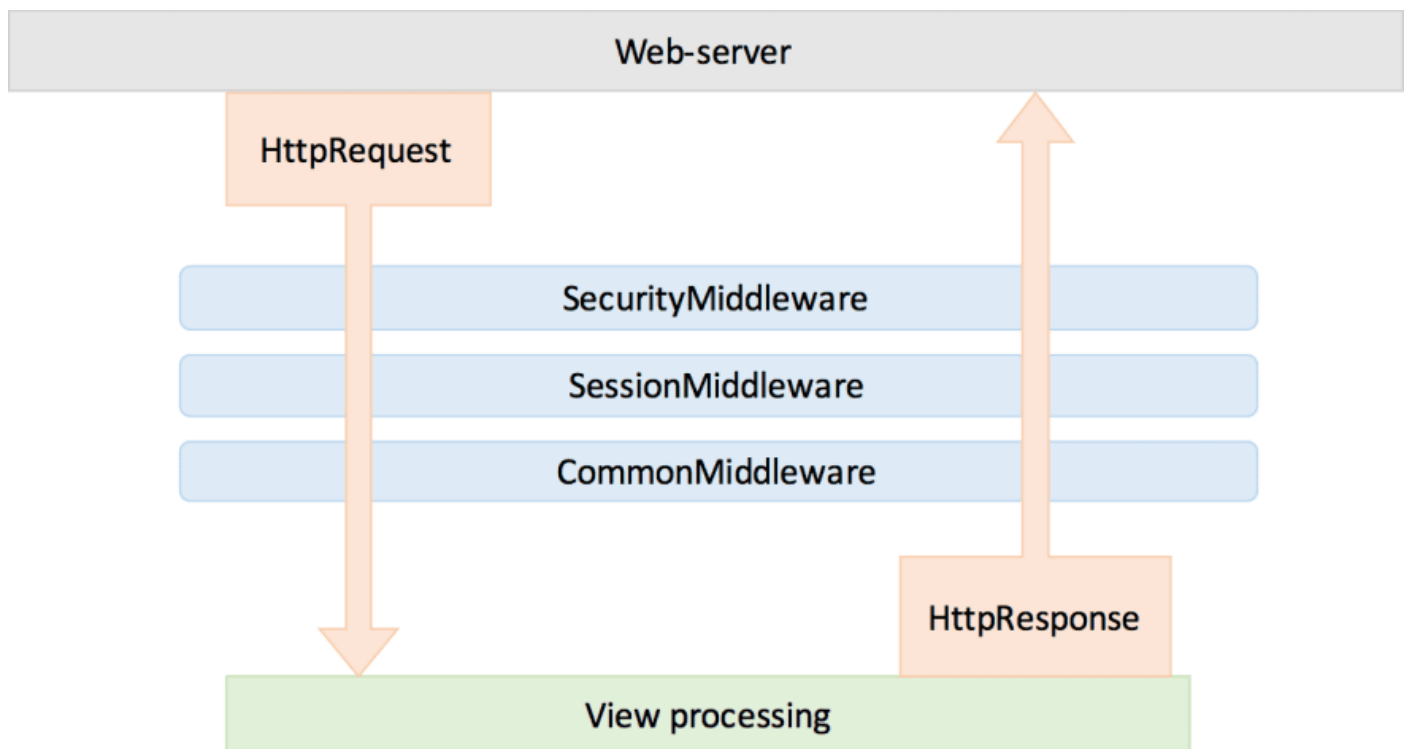
Middleware is a way to hook some functionality into request/response processing of Django Framework. Each middleware is meant to perform some process either before request is processed or/and after request is processed. This is similar to Filter concept in Java EE.

Here are the steps to create a custom middleware that is used to add copyright information at the end of all responses.

Create function called `footer()` in module `footer_middleware.py`. The function `footer()` is supposed to return a function (middleware () in this case) that is to be hooked in request pipeline.

The function returned by `footer()` must itself return an object of type `Response`.

It is also calling `get_response()` method to get access to response after view is processed.



DEPLOYMENT OF PYTHON PROJECT

The python Framework provides several basic features that make it easier to deploy the applications:

No-impact applications.

This feature provides application isolation and eliminates DLL conflicts. By default, components do not affect other applications.

Private components by default.

By default, components are deployed to the application directory and are visible only to the containing application.

Controlled code sharing.

Code sharing requires to explicitly make code available for sharing instead of being the default behavior.

Side-by-side versioning.

Multiple versions of a component or application can coexist, choose which versions to use, and the common language runtime enforces versioning policy.

XCOPY deployment and replication.

Self-described and self-contained components and applications can be deployed without registry entries or dependencies.

On-the-fly updates.

Administrators can use hosts, such as ASP.NET, to update program DLLs, even on remote computers.

Integration with the Windows Installer.

Advertisement, publishing, repair, and install-on-demand are all available when deploying the application.

Enterprise deployment.

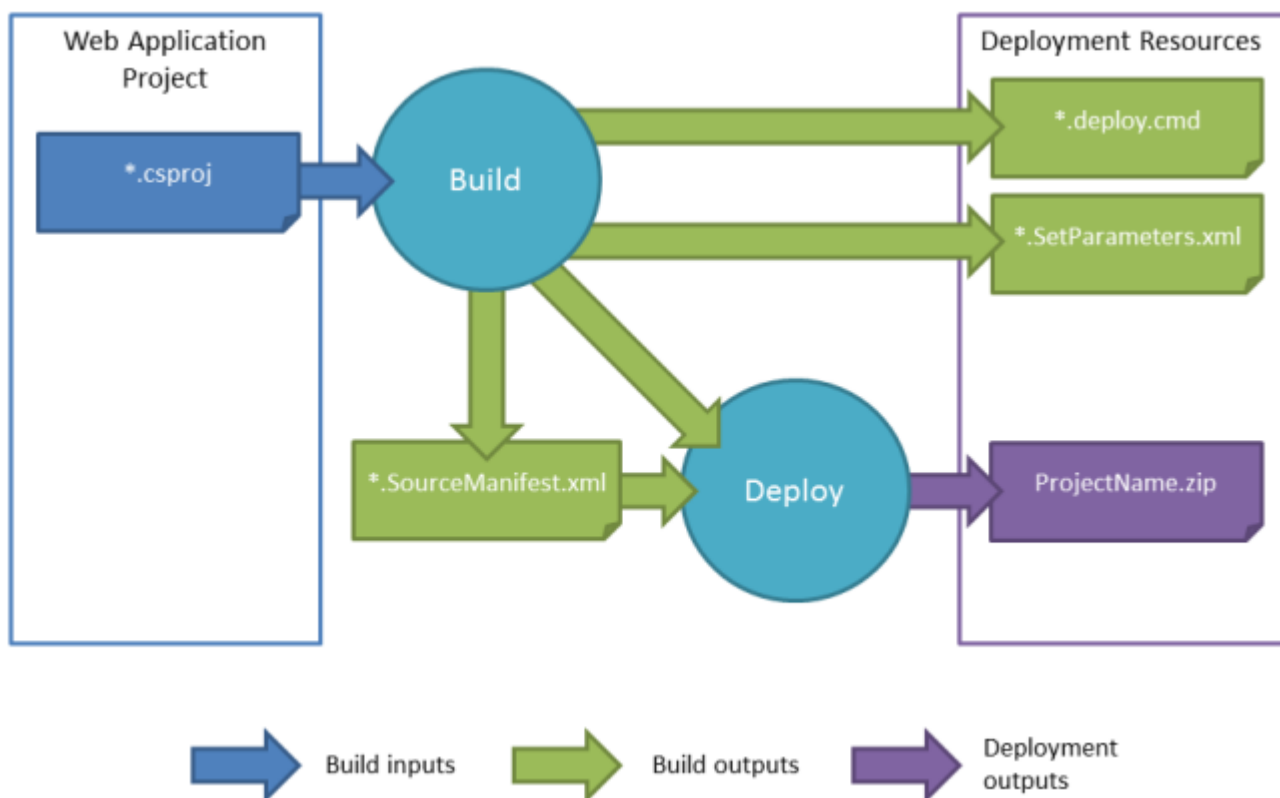
This feature provides easy software distribution, including using Active Directory.

Downloading and caching.

Incremental downloads keep downloads smaller, and components can be isolated for use only by the application for low-impact deployment.

Partially trusted code.

Identity is based on the code instead of the user, and no certificate dialog boxes appear.



Our Python app development company experts efficiently and swiftly create prototypes of the idea you have in your mind.

Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

IMPLEMENTATIONS

Reference implementation

CPython is the reference implementation of Python. It is written in C, meeting the C89 standard (Python 3.11 uses C11) with several select C99 features (With later C versions out, it is considered outdated. CPython includes its own C extensions, but third-party extensions are not limited to older C versions—e.g. they can be implemented with C11 or C++.) It compiles Python programs into an intermediate bytecode which is then executed by its virtual machine. CPython is distributed with a large standard library written in a mixture of C and native Python, and is available for many platforms, including Windows (starting with Python 3.9, the Python installer deliberately fails to install on Windows 7 and 8; Windows XP was supported until Python 3.5) and most modern Unix-like systems, including macOS (and Apple M1 Macs, since Python 3.9.1, with experimental installer) and unofficial support for e.g. VMS. Platform portability was one of its earliest priorities. (During Python 1 and 2 development, even OS/2 and Solaris were supported, but support has since been dropped for many platforms.)

Other implementations

- PyPy is a fast, compliant interpreter of Python 2.7 and 3.8 Its just-in-time compiler brings a significant speed improvement over CPython but some libraries written in C cannot be used with it.
- Stackless Python is a significant fork of CPython that implements microthreads; it does not use the call stack in the same way, thus allowing massively concurrent programs. PyPy also has a stackless version.
- MicroPython and CircuitPython are Python 3 variants optimized for microcontrollers, including Lego Mindstorms EV3.
- Pyston is a variant of the Python runtime that uses just-in-time compilation to speed up the execution of Python programs.
- Cinder is a performance-oriented fork of CPython 3.8 that contains a number of optimizations including bytecode inline caching, eager evaluation of coroutines, a method-at-a-time JIT and an experimental bytecode compiler.

Unsupported implementations

Other just-in-time Python compilers have been developed, but are now unsupported:

- Google began a project named Unladen Swallow in 2009, with the aim of speeding up the Python interpreter fivefold by using the LLVM, and of improving its multithreading ability to scale to thousands of cores,[143] while ordinary implementations suffer from the global interpreter lock.
- Psyco is a discontinued just-in-time specializing compiler that integrates with CPython and transforms bytecode to machine code at runtime. The emitted code is specialized for certain data types and is faster than the standard Python code. Psyco does not support Python 2.7 or later.
- PyS60 was a Python 2 interpreter for Series 60 mobile phones released by Nokia in 2005. It implemented many of the modules from the standard library and some additional modules for integrating with the Symbian

operating system. The Nokia N900 also supports Python with GTK widget libraries, enabling programs to be written and run on the target device

Cross-compilers to other languages

There are several compilers to high-level object languages, with either unrestricted Python, a restricted subset of Python, or a language similar to Python as the source language:

- Brython, Transcrypt and Pyjs (latest release in 2012) compile Python to JavaScript.
- Cython compiles (a superset of) Python 2.7 to C (while the resulting code is also usable with Python 3 and also e.g. C++).
- Nuitka compiles Python into C.
- Numba uses LLVM to compile a subset of Python to machine code.
- Pythran compiles a subset of Python 3 to C++ (C++11)
- RPython can be compiled to C, and is used to build the PyPy interpreter of Python.

Specialized:

- MyHDL is a Python-based hardware description language (HDL), that converts MyHDL code to Verilog or VHDL code.

Older projects (or not to be used with Python 3.x and latest syntax):

- Google's Grumpy (latest release in 2017) transpiles Python 2 to Go.
- IronPython allows running Python 2.7 programs (and an alpha, released in 2021, is also available for "Python 3.4, although features and behaviors from later versions may be included") on the .NET Common Language Runtime.
- Jython compiles Python 2.7 to Java bytecode, allowing the use of the Java libraries from a Python program.
- Pyrex (latest release in 2010) and Shed Skin (latest release in 2013) compile to C and C++ respectively.

Performance

Performance comparison of various Python implementations on a non-numerical (combinatorial) workload was presented at EuroSciPy '13. Python's performance compared to other programming languages is also benchmarked by The Computer Language Benchmarks Game.

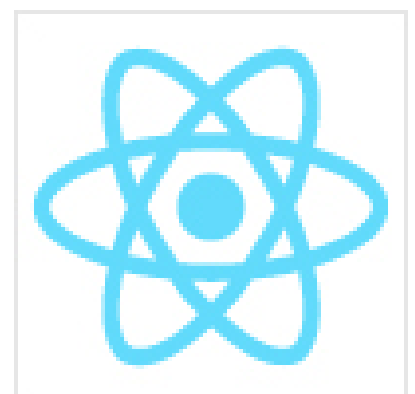
BEST FRONT-END FRAMEWORKS

React

React is the open-source framework developed and created by Facebook. This framework is the best UI framework of 2022, used by a majority of the frontend developers according to Stack Overflow Developer's survey 2021.

Its team was concerned with procuring excellent performance by forming a useful UI. The primary intention was to fix code maintainability problems due to the constant addition of features in the app.

A front-end framework, React stands apart because of its virtual Document Object Model (DOM), which presents its excellent functionality. It is a perfect framework for those who expect high traffic and require a steady platform to manage it.



Furthermore, this framework is user-friendly for new developers – the guides also aid to deal with any complications that happen during the learning process.

Pros

- Saving time while re-utilizing components
- Virtual DOM enhances both the experience of the users and the work of the developer
- An open-source library with a diversity of tools
- The steady code is supplied by one direction data movement

Cons

- Absence of documentation due to significant pace development
- The comparatively long learning curve
- The developers find it challenging to comprehend the complications of JSX

When to use:

React is utilized for developing rich user interface, particularly when you need to build single-page apps. It is the most robust front-end framework when you require to make an interactive interface with less time since it supports reusable components.

When not to use:

When you do not have hands-on knowledge with JavaScript, React is not the suggested alternative. Similarly, for inexperienced developers, the JSX learning curve is severe.

Angular

The list of best front end development frameworks is incomplete without mentioning Angular. It is a simple and straightforward framework that is based on TypeScript.

Formally released in 2016, the Angular framework was established by Google to link the gap between the mounting demands of technology and conventional notions that displayed the results.

In contrast to React, Angular is exclusive with its two-way data binding trait. It means that there is actual time synchronization between view and model, where any alteration in the model replicates promptly on view and vice versa.

If your scheme contains developing web or mobile apps, Angular is a perfect choice. As an Angular development agency in Canada, our web developers build progressive web applications and single page applications with the framework.

Companies like Xbox, BMW, Blender, and Forbes are some of the best applications developed with Angular.

In comparison of Angular vs React, Angular is not simple to learn. However, there is countless documentation accessible; they are either too composite or perplexing to read.



Pros

- Making the coding procedure easier due to its refactoring services and enhanced navigation
- The component-based pattern of Angular sanctions forms a user interface with single components
- Large Ecosystem
- Angular Material reorganizes Material Design interface production
- High Performance

Cons

- Angular complication
- Relocating legacy schemes from AngularJS to Angular
- The CLI documentation is not fairly defined
- The learning effort

When to use:

Angular enlarges the execution of browser-based apps by vigorously modernizing the contents in less time since it exercises two-way data binding. For an enterprise based and active web app, utilizing Angular is apt.

When not to use:

Angular is a comprehensive solution as a front-end framework. If you need to develop apps with inadequate scopes, you will not be capable of employing the resources that Angular furnishes.

Vue.js

Nowadays, one of the simplest frameworks is Vue.js. It is worthy of eliminating the intricacies that Angular software developers face.

It has a small size and presents two main benefits – visual DOM and component-based. It also employs two-way data binding. This frontend framework is versatile, and it aids you with various tasks when building web applications.

From constructing web and mobile applications to progressive web apps (PWA), it can manage both dynamic and simple developments with simplicity.

The difference between Vue and React is that Vue is a JS framework while React is a JS library. So Vue is more suitable for large projects. Although Vue is created to tackle complexities and enhance app performance, it is not very popular amongst the industry's giants yet. Similarly, comparing Angular vs Vue.js, Vue improves the performance and usability of Angular.

Nevertheless, Xiaomi, Alibaba, Reuters, and 9Gag are the users of this framework. Vue.js continues to propagate in terms of adoptions regardless of fewer takers from Silicon Valley.



Pros

- Extensive and detailed documentation
- Simplicity and clarity

- Browser dev tools extensions
- Code reusability and simple integration

Cons

- Reduced developers' community
- Flexibility leads to code irregularities

When to use:

Vue.js is one of the most popular frontend frameworks today owing to its simplicity and flexibility. It lets you scheme the whole thing from scratch and is efficacious in building massive projects as well. Apt for progressive web apps, dynamic web apps, and complex projects with a scalable and efficient architecture.

When not to use:

If you consider the support community would be there to respond to the intricacies, Vue.js is not the right track to go. Likewise, the applications necessitating steady components are not appropriate to be fabricated with Vue since the framework has presented difficulties with the firmness of parts.

Our Python experts are skilled in developing highly secure backend platforms for your application.

Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

Ember.js

Ember.js is a component-based framework developed in 2011. It presents two-way data binding, which is similar to Angular. It is intended to manage the growing demand for contemporary technologies impeccably.



You can build multifaceted web and mobile applications with Ember.js and expect its effective design to handle the concerns.

Being comparatively new and unexplored, its developer community is trivial. Anyone with no restraint in devoting the time for acquiring its knowledge can learn it.

Pros

- Fastest development framework
- Well-managed
- Appropriate documentation
- Two-way data binding

Cons

- Slow updates and challenging syntax
- A minor community and less popular
- Substantial framework for small apps
- Difficult learning curve

When to use:

If you need to develop contemporary applications like LinkedIn with a responsive user interface, Ember.js is the right frontend framework. It comes with every mechanical front-end enablement, like seeing a more comprehensive range of applications because of outstanding routing delivered by Ember.js.

This framework positions as the complete frontend solution for an extensive project because it supplies good data binding, an equipped configuration, and custom properties to deliver the page as required.

When not to use:

Ember.js is probably not an apt choice with a smaller development team since the framework needs business logic and experience to resolve the difficulties. The preliminary cost may be more significant with Ember.js.

Similarly, for scripting simple Ajax functionalities and building user interfaces, the framework may not be suitable.

jQuery

jQuery is one of the initial frontend frameworks introduced in 2006. Regardless of its release date, what makes it stand apart is its significance in today's tech ecosystem.

Not only does jQuery present ease of use, but it also diminishes the requirement to script wide-ranging JavaScript codes.

Basically, jQuery is utilized to operate DOM and CSS manipulation on the frontend and to augment the interactivity and functionality of a website.



Though you are not able to build mobile apps with jQuery, current developments in jQuery Mobile have shown that the usage borders have been extended. Moreover, the latest development in this framework allows the developers to create native mobile apps with its HTML5 based UI scheme – jQuery Mobile.

Further, jQuery framework is browser-friendly and supports the browsers you aim to utilize.

Pros

- Conveying HTTP requests is streamlined
- Document Object Model (DOM) is flexible for accumulating or eliminating the components
- HTTP requests are made easy
- Simplifies dynamic content

Cons

- Numerous progressive substitutes are accessible other than jQuery
- APIs of the Document Object Model (DOM) is outdated
- Reasonably sluggish working capability

When to use:

This web development framework is utilized in building desktop-based JavaScript apps. This framework retains the code crisp and fairly simple. It is employed in managing the events and executing animations.

When not to use:

When you are building a large-scale app, it is not feasible to utilize jQuery as it makes your application hefty by placing extra JavaScript code. This framework is not proficient in challenging the contemporary frameworks with the progressive enablement of JavaScript, fewer coding, and reusability of elements.

Semantic-UI

Semantic UI was invented by Jack Lukic and it got released in 2014. It is a new frontend framework that is supported by LESS and jQuery.

The objective of Semantic lies in authorizing the designers and developers by creating a language for sharing UI. It uses natural language that makes the entire code self-explanatory.

Fresher can simply make command as it permits a unified development procedure with incorporation to several third-party libraries.



Pros

- Has superior functionality
- Amongst the newest front-end frameworks
- Rich UI modules and receptiveness
- Less complex

Cons

- Not appropriate for learners with no knowledge of JavaScript
- Have to be skilled enough to build custom necessities

- Larger packages

When to use:

Semantic-UI is used for a super lightweight practice and smooth design of interactive user interfaces.

When not to use:

When you have a squad of beginners and are less knowledgeable with JavaScript, Semantic-UI framework is not suggested because it needs the aptitude to form customizations in the app without relying on the handy functions.

Backbone.js

Backbone.js is one of the easiest frameworks which permits you to build single-page applications quickly. It is a framework that is based on the MVC pattern. Parallel to a Controller, MVC architecture View sanctions the execution of the component logic.

Consequently, irrespective of whether you aim to implement Backbone.js for backend or frontend, this is a perfect choice as its REST API compatibility delivers smooth synchronization amid the two.



Pros

- Fastest JavaScript frameworks
- Much easier to learn
- Have a lightweight framework

Cons

- Delivers uncomplicated tools to produce the app construction.
- The framework does not offer a handy structure
- Have to transcribe boilerplate code

When to use:

Backbone.js is utilized for vibrant applications like Trello. It permits developers to develop a client-side model, quicker updates, and reuse the coding. Henceforth, it is proficient in vigorously maintaining the client, conducting the updates, and continuous synchronization with the server.

When not to use:

Backbone.js offers a minimalist prerequisite for developing a web app as compared to the rest of the MVC client-side frameworks. Nevertheless, you can expand the functionalities with the provision of extensions and plugins.

Consequently, the team targeting for a comprehensive solution in one framework should not intend for Backbone.js.

Preact

Preact is a JavaScript library that defines itself as the quickest 3KB substitute to that of React containing a similar ES6 API. Preact presents the smallest possible virtual DOM abstraction in addition to DOM.

It develops on steady platform features and functions well with other available frontend and UI libraries. Preact has a small size but does not compromise in speed, and enables developing complex dynamic web applications.



Pros

- Utilize the advantages of React-style elements in the old sites that are founded on the Backbone and jQuery
- Preact increases performance while building an app
- All these characteristics are executed in the React community

Cons

- It does not support the React prototype
- Preact does not deliver any support for synthetic event employment of React
- It still does not offer support for the context

When to use:

As it is known, Preact is a lightweight form of React so when it is about utilizing a lightweight framework then you should opt for Preact and not React.

When not to use:

Preact does not offer support to displaced functional elements. So when you are in such a requirement, then you must not use Preact.

Svelte

Svelte is one of the modern web frameworks for frontend development. This framework has made a change by placing the work into an accumulated step instead of tapping them in the browser, unlike the frameworks like Vue and React.

It transcribes the code to update the Document Object Model (DOM) in synchronization with the condition of the application.



Pros

- Scalable framework
- Improved reactivity
- It is lightweight, simple, and uses the prevailing JavaScript libraries

- Newest of all
- Quicker than any other framework, namely, React or Angular

Cons

- Lack of support
- It is not very popular yet
- Minor community
- Lack of tooling

When to use:

This framework is most suitable for small application projects having a small team. Because it does not have a larger support community, it is better if you do not use it for multifarious projects.

When not to use:

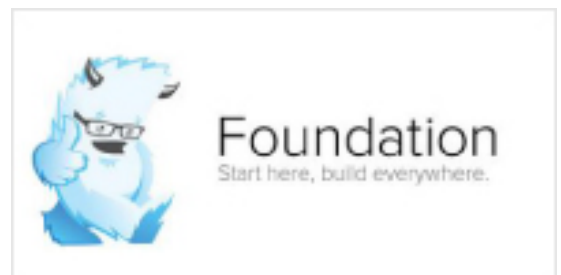
It is better if you do not use the Svelte framework for huge projects presently because the frameworks lack sufficient community as well as tooling.

It is difficult to search for answers to the questions or bugs that may be visible later in the development process because of the minor community.

Foundation

Invented by Zurb, Foundation is majorly intended for enterprise-level development of agile and responsive websites. It is advanced as well as challenging for web developers to begin developing frontend applications utilizing Foundation.

It presents GPU acceleration for swift mobile rendering features, smooth animations, and data-interchange attributes where it loads lightweight segments for heavy and mobile sections for bigger devices.



Pros

- It lets you form exquisite looking websites
- Flexible grids
- Personalized user experience for dissimilar devices and media
- HTML5 form authentication library

Cons

- Fewer community forums and support platforms
- Contender framework like Twitter, Bootstrap is more prevalent than Foundation
- Reasonably hard to learn for beginners

When to use:

If you are aiming for styled open-source, CSS components, and mobile-friendly front end framework, Foundation is better than the rest of the other options.

When not to use:

Not preferable for beginners since it is difficult to alter the code and increases the complexity because of its customization competence.

Transform your old business applications with Python into new innovative solutions for efficient and scalable operational workflows.

Contact Us**Cognitive Convergence**

<http://www.cognitiveconvergence.com>

+1 4242530744 info@cognitiveconvergence.com

LIBRARIES

Python's large standard library provides tools suited to many tasks, and is commonly cited as one of its greatest strengths. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary-precision decimals, manipulating regular expressions, and unit testing.

Some parts of the standard library are covered by specifications—for example, the Web Server Gateway Interface (WSGI) implementation `wsgiref` follows PEP 333—but most are specified by their code, internal documentation, and test suites. However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of June 2022, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 380,000 packages with a wide range of functionality, including:

- Automation
- Data analytics
- Databases
- Documentation
- Graphical user interfaces
- Image processing
- Machine learning
- Mobile apps
- Multimedia
- Computer networking
- Scientific computing

- System administration
- Test frameworks
- Text processing
- Web frameworks
- Web scraping

PYTHON DEVELOPMENT ENVIRONMENT

Text Editors

Just about anything that can edit plain text will work for writing Python code; however, using a more powerful editor may make your life a bit easier.

Vim

Vim is a text editor which uses keyboard shortcuts for editing instead of menus or icons. There are a couple of plugins and settings for the Vim editor to aid Python development. If you only develop in Python, a good start is to set the default settings for indentation and line-wrapping to values compliant with PEP 8. \

Python-mode

Python-mode is a complex solution for working with Python code in Vim. It has:

- Asynchronous Python code checking (pylint, pyflakes, pycodestyle, mccabe) in any combination
- Code refactoring and autocompletion with Rope
- Fast Python folding
- Virtualenv support
- Search through Python documentation and run Python code
- Auto pycodestyle error fixes

SuperTab

SuperTab is a small Vim plugin that makes code completion more convenient by using <Tab> key or any other customized keys.

Emacs

Emacs is another powerful text editor. It is fully programmable (Lisp), but it can be some work to wire up correctly.

A good start if you're already an Emacs user is Python Programming in Emacs at EmacsWiki.

1. Emacs itself comes with a Python mode.

TextMate

TextMate brings Apple's approach to operating systems into the world of text editors. By bridging Unix underpinnings and GUI, TextMate cherry-picks the best of both worlds to the benefit of expert scripters and novice users alike.

Sublime Text

Sublime Text is a sophisticated text editor for code, markup, and prose. You'll love the slick user interface, extraordinary features, and amazing performance.

Sublime Text has excellent support for editing Python code and uses Python for its plugin API. It also has a diverse variety of plugins, some of which allow for in-editor PEP8 checking and code "linting".

Atom

Atom is a hackable text editor for the 21st century, built on atom-shell, and based on everything we love about our favorite editors.

Atom is web native (HTML, CSS, JS), focusing on modular design and easy plugin development. It comes with native package control and a plethora of packages. Recommended for Python development is Linter combined with linter-flake8.

IDEs

PyCharm / IntelliJ IDEA

PyCharm is developed by JetBrains, also known for IntelliJ IDEA. Both share the same code base and most of PyCharm's features can be brought to IntelliJ with the free Python Plug-In. There are two versions of PyCharm: Professional Edition (Free 30-day trial) and Community Edition (Apache 2.0 License) with fewer features.

Python (on Visual Studio Code)

Python for Visual Studio is an extension for the Visual Studio Code IDE. This is a free, lightweight, open-source IDE, with support for Mac, Windows, and Linux. Built using open-source technologies such as Node.js and Python, with compelling features such as Intellisense (autocompletion), local and remote debugging, linting, and the like. MIT licensed.

Enthought Canopy

Enthought Canopy is a Python IDE which is focused towards Scientists and Engineers as it provides pre installed libraries for data analysis.

Eclipse

The most popular Eclipse plugin for Python development is Aptana's PyDev.

Komodo IDE

Komodo IDE is developed by ActiveState and is a commercial IDE for Windows, Mac, and Linux. KomodoEdit is the open-source alternative.

Spyder

Spyder is an IDE specifically geared toward working with scientific Python libraries (namely SciPy). It includes integration with pyflakes, pylint and rope.

Spyder is open source (free), offers code completion, syntax highlighting, a class and function browser, and object inspection.

WingIDE

WingIDE is a Python specific IDE. It runs on Linux, Windows, and Mac (as an X11 application, which frustrates some Mac users).

WingIDE offers code completion, syntax highlighting, source browser, graphical debugger and support for version control systems.

NINJA-IDE

NINJA-IDE (from the recursive acronym: “Ninja-IDE Is Not Just Another IDE”) is a cross-platform IDE, specially designed to build Python applications, and runs on Linux/X11, Mac OS X, and Windows desktop operating systems.

NINJA-IDE is open-source software (GPLv3 licence) and is developed in Python and Qt. The source files can be downloaded from GitHub.

Eric (The Eric Python IDE)

Eric is a full featured Python IDE offering source code autocompletion, syntax highlighting, support for version control systems, Python 3 support, integrated web browser, python shell, integrated debugger, and a flexible plug-in system.

Written in Python, it is based on the Qt GUI toolkit, integrating the Scintilla editor control. Eric is an open-source software project (GPLv3 licence) with more than ten years of active development.

Interpreter Tools

Virtual Environments

Virtual Environments provide a powerful way to isolate project package dependencies. This means that you can use packages particular to a Python project without installing them system wide and thus avoiding potential version conflicts.

pyenv

pyenv is a tool to allow multiple versions of the Python interpreter to be installed at the same time. This solves the problem of having different projects requiring different versions of Python. For example, it becomes very easy to install Python 2.7 for compatibility in one project, while still using Python 3.4 as the default interpreter. pyenv isn't just limited to the CPython versions – it will also install PyPy, Anaconda, miniconda, stackless, Jython, and IronPython interpreters.

pyenv works by filling a shims directory with fake versions of the Python interpreter (plus other tools like pip and 2to3). When the system looks for a program named python, it looks inside the shims directory first, and uses the fake version, which in turn passes the command on to pyenv. pyenv then works out which version of Python should be run based on environment variables, python-version files, and the global default.

Other Tools

IDLE

IDLE is an integrated development environment that is part of the Python standard distribution. It is completely written in Python and uses the Tkinter GUI toolkit. Though IDLE is not suited for full-blown development using Python, it is quite helpful to try out small Python snippets and experiment with different features in Python.

It provides the following features:

- Python Shell Window (interpreter)
- Multi window text editor that colorizes Python code
- Minimal debugging facility

IPython

IPython provides a rich toolkit to help you make the most out of using Python interactively. Its main components are:

- Powerful Python shells (terminal- and Qt-based)
- A web-based notebook with the same core features but support for rich media, text, code, mathematical expressions and inline plots
- Support for interactive data visualization and use of GUI toolkits
- Flexible, embeddable interpreters to load into your own projects
- Tools for high level and interactive parallel computing

BPython

bpython is an alternative interface to the Python interpreter for Unix-like operating systems. It has the following features:

- In-line syntax highlighting
- Readline-like autocomplete with suggestions displayed as you type
- Expected parameter list for any Python function
- “Rewind” function to pop the last line of code from memory and re-evaluate
- Send entered code off to a pastebin
- Save entered code to a file
- Auto-indentation
- Python 3 support

ptpython

ptpython is a REPL build on top of the prompt_toolkit library. It is considered to be an alternative to BPython. Features include:

- Syntax highlighting
- Autocompletion
- Multiline editing
- Emacs and Vim Modes
- Embedding REPL inside of your code
- Syntax validation

- Tab pages
- Support for integrating with IPython's shell, by installing IPython (pip install ipython) and running ptipython.

The apps are designed to derive powerful insights from collected company data that help make effective business decisions.

Contact Us

Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744 info@cognitiveconvergence.com

DATABASE WITH PYTHON

The following list includes some of the more popular database systems that can be used in Visual Studio projects. The list is not exhaustive. For a list of third-party vendors that offer ADO.NET data providers that enable deep integration with Visual Studio tooling, see ADO.NET Data Providers.

Microsoft SQL Server

SQL Server is the Microsoft flagship database offering. SQL Server 2016 delivers breakthrough performance, advanced security, and rich, integrated reporting and analytics. It ships in various editions that are designed for different uses: from highly scalable, high-performance business analytics, to use on a single computer. SQL Server Express is a full-featured edition of SQL Server that is tailored for redistribution and embedding. LocalDB is a simplified edition of SQL Server Express that requires no configuration and runs in your application's process. You can download either or both products from the SQL Server Express download page.



Oracle

You can download a paid or free edition of the Oracle database from the Oracle technology network page. For design-time support for Entity Framework and TableAdapters, you will need the Oracle Developer tools for Visual Studio. Other official Oracle products, including the Oracle Instant Client, are available through the NuGet Package Manager. You can download Oracle sample schemas by following the instructions in the Oracle online documentation.

The Oracle logo, consisting of the word "ORACLE" in a bold, red, sans-serif font.

MySQL

MySQL is a popular open-source database system that is widely used in enterprises and websites. Downloads for MySQL, MySQL for Visual Studio, and related products are at [MySQL on Windows](#). Third parties offer various Visual Studio extensions and stand-alone management applications for MySQL. You can browse the offerings in the NuGet Package Manager (Tools > NuGet Package Manager > Manage NuGet Packages for Solution).



PostgreSQL

PostgreSQL is a free, open-source object relational database system. To install it on Windows, you can download it from the [PostgreSQL download page](#). You can also build PostgreSQL from the source code. The PostgreSQL core system includes a C language interface. Many third parties provide NuGet packages for using PostgreSQL from .NET applications. You can browse the offerings in the NuGet Package Manager (Tools > NuGet Package Manager > Manage NuGet Packages for Solution). Perhaps, the most popular package is provided by [npgsql.org](#).



SQLite

SQLite is an embedded SQL database engine that runs in the application's own process. You can download it from the [SQLite download page](#). Many third-party NuGet packages for SQLite are also available. You can browse the offerings in the NuGet Package Manager (Tools > NuGet Package Manager > Manage NuGet Packages for Solution).



Firebird

Firebird is an open-source SQL database system. You can download it from the Firebird download page. An ADO.NET data provider is available through the NuGet Package Manager.



USE CASES OF PYTHON

1. Web applications

Python streamlines web application development with its built-in frameworks:

- Django
- Flask
- Falcon
- Tornado
- Pyramid
- Masonite
- FastAPI

Ultimately, these frameworks enable speedy application builds with minimal development time. You can read more about their individual nuances [here](#).

2. Software development

Python is used to develop many different applications and platforms across industries. Notable examples include Instagram and Spotify.

The FinTech industry, in particular, favours Python over other coding languages to create competitive applications (such as Venmo). This is due to Python's ability to handle a range of complex mathematical tasks, of which the financial sector has many.

3. Data science

Data scientists use Python to help them extract, process and manipulate data via various Python libraries, such as Pandas and SciPy.

The coding language is relatively easy to learn when used in conjunction with data. As such, Python skills are increasingly in demand across both big and small data-driven organizations.

4. AI applications

The number of businesses adopting artificial intelligence grew by 270 percent in four years. It's no wonder, then, that AI and machine learning project development is a top priority for many organizations.

Python can easily handle the computations necessary to create machine learning models and is responsible for creating many advanced applications, such as facial recognition software.

5. Game development

Though not the most popular language for game development, Python is often used by developers for building tree-based algorithms and designing game levels.

In addition to this, some developers use Python for scripting games or automating repetitive tasks (such as batching game textures)

INDUSTRIES USE PYTHON IN THEIR BUSINESS

Google:

- Ideally, Google needs practically zero introduction. Truth be told, it's genuinely safe to say that you're utilizing a result of Google, whether that is your web crawler, internet browser, or both.
- Python is one of Google's true server-side programming languages, alongside C++, Java, and Go. Peter Norvig, the head of examination at Google and previous overseer of search quality, keeps up with that, "Python has been a significant component of Google from the start, and remains so as the technology develops and advances."
- Google uses Python for YouTube, the world's most popular video sharing site.



Facebook:

- Facebook production engineers are outstandingly enthusiastic about Python, making it the third most famous language at the web-based entertainment company (simply behind C++ and their restrictive PHP language, Hack).
- By and large, there are more than 5,000 commits on utilities and services at Facebook, monitoring framework, binary dispersion, equipment imaging, and functional automation.
- The simplicity of utilizing Python libraries implies that the production engineers don't need to compose or maintain as much code, permitting them to focus on getting updates live.

- Hiring a Python development company in India will offer enterprises & startups the same advantages that Facebook gets from using Python.



Uber:

- Uber, a global organization that allows you to book rides to your destinations, utilizes Python alongside Node.js, Java, and Go at the lower levels. The vast majority of the services for its users are as yet fueled by Python, including the business rationale and every one of the estimations that occur.
- This includes the middle and high levels, for example, working out ETA, ride passages, ascertaining the geo-areas, and the demand and supply.
- Uber professes to utilize Tornado with Python yet frequently trades that with Go to accomplish better throughput concerning concurrency.



Spotify:

- This music streaming company is an immense defender of Python, utilizing the language basically for data examination and back-end services. The explanation that the services are written in Python is on the grounds that Spotify likes how quick the improvement pipeline is while composing and coding in Python.
- To give ideas and suggestions to its users, Spotify depends on an enormous volume of examination. To decipher these, Spotify utilizes Luigi, a Python module that synchronizes with Hadoop.
- Altogether, Spotify utilizes more than 6000 individual Python processes that cooperate over the nodes of the Hadoop bunch.



Instagram:

- Instagram is an image and video-sharing online entertainment platform offered by Facebook, Inc. It is one of the most well-known organizations that use Python language in relationship with Django, and lately, Instagram has embraced Python.
- The excellent justification for Instagram to decide on Python is its productivity, straightforwardness, and common sense. For any flourishing organization, these details are outstandingly important.
- It is an 'amicable' language to engineers, which allows them to zero in on suitable elements. Instagram considered changing to PHP, yet after a few contemplations, they detected that Python would serve them best.



Dropbox:

- Dropbox is a cloud-based data storing service to store all your photographs, recordings, archives, and documents, and its whole tech stack is composed of Python.
- In 2012 Dropbox persuaded Rossum, the innovator of Python, to join Dropbox, who then directed them working there as a specialist and assisted the Python developers with distributing datastores. It can be imparted to the individuals from the Dropbox people team.
- Since a ton of Dropbox's libraries aren't open-source, it is challenging to examine their degree of reliance on Python yet it is noted from large numbers of the meetings with their developers that a greater part of their code on the server-side is written in Python.



Quora:

- This immense publicly supported Q & A stage considered what language they needed to use to execute their idea. The pioneers behind Quora took their lead from Google, deciding to utilize Python where they could, on account of its simplicity of writing code and meaningfulness, and executed C++ for the presentation of basic areas.

- They got around Python's absence of type checking by creating unit tests that achieve a lot of exactly the same thing. One more key thought for utilizing Python was the presence of a few decent frameworks at the time, including Django and Pylons.



Netflix:

- Netflix implements Python very much like Spotify, depending on the language to control its data analysis on the server-side. It doesn't stop there, nonetheless. Netflix allows its developers to pick what language to code in and has seen a huge upsurge in the quantity of Python applications.
- When overviewed, Netflix engineers refer to the standard library, the very dynamic Python community, and the rich assortment of outsider libraries accessible to tackle almost any given issue.
- Also, on the grounds that Python is so natural to write, it has turned into a key part in a large number of Netflix's different services.



WHY CHOOSE PYTHON?

Python is a leading technology and we've listed its main benefits:

- 1) Easy to Learn and Use
- 2) Mature and Supportive Python Community
- 3) Support from Renowned Corporate Sponsors
- 4) Hundreds of Python Libraries and Frameworks
- 5) Versatility, Efficiency, Reliability, and Speed
- 6) Big data, Machine Learning and Cloud Computing
- 7) First-choice Language
- 8) The Flexibility of Python Language
- 9) Use of python in academics
- 10) Automation

CONCLUSION

As we have seen several advantages of the python framework for application development. The python development framework is a reliable, safe, and beneficial platform using which you can build ultimate business applications. It allows python developers to apply, expand, and scale excellent software, for both economic development and general business prosperity.

Python development services provide endless benefits for different problems such as security, memory management, phenomenal handling, etc. The python framework is an exciting and future dominating technology, appropriate for building AI/ML applications. Since most organizations search for a unique business solution to assist consumers in the best possible way, you can contact cognitive convergence.

Contact Us
Cognitive Convergence
<http://www.cognitiveconvergence.com>
+1 4242530744
info@cognitiveconvergence.com