

अजगर

By Aarsh Garg

AJGAR



AARSH GARG

AJGAR

अजगार

Project AJGAR

Project AJGAR aims to make Python programmable in Hindi!

```
छापो("Namaste")
naam = पूछो("Aapka naam kya hai? ")
one = संख्या("1")
pi = दशमलव(22 / 7)
छापो(one, pi)
अगर naam == "Aarsh":
    छापो("Namaste, Aarsh Garg")
वरना:
    छापो("Namaste", naam)
केलिए i अंदर श्रेणी(1, 100):
    छापो(i)
loop = सत्य
जबतक loop:
    छापो("loop")
    loop = असत्य
निर्धारित addition(a: संख्या, b: संख्या):
    वापसी a + b
छापो(addition(1, 20))
छापो("Dhanyavad!")
```

But **WHY** Ajgar?

According to the Census of India, 2011, nearly 49%* of India's total population couldn't speak English, that is, 590 Million people of India, could only speak their Mother Tongue.

We can only think and guess, how many people could be in that 590 Million, who had Hindi-Medium schooling, good at academics, but never got any opportunities because they couldn't speak English!

This contributed to unemployment, and one's household was ruined.

Until NEP 2023 was introduced to promote multilingualism.

*Source - Census Digital Library.

Multilingualism, in Programming?

NEP 2023 has contributed to the growth of Multilingualism.

Multiple career paths opened their doors to multilingual employees:

- Bilingual Instructional Designer
- AI Localization Engineer / Natural Language Processing (NLP) Specialist
- Multilingual Product Manager
- Vernacular UX/UI Designer
- Technical Content Translator / Academic Adapter
- Vocational Training Coordinator
- Regional Market Expansion Specialist
- Bilingual Customer Success Manager
- ECCE (Early Childhood Care and Education) Educator
- Public Sector Undertaking (PSU) Technical Officer
- Community Relations Manager / Corporate Social Responsibility (CSR) Lead
- Regional Content Creator / Digital Copywriter

But where is Programming?

Attempts have been made before...

In India, there have been made several attempts to make native-language programs, like:

- Kalaam - [Kalaam Lang](#).
- OM Lang - [OM Lang Official Site](#)
- Bhailang - [Bhailang Web Play](#)
- Hindawi Programming System - [Hindawi 2020 GitHub](#)

The problem is, no software firm is ready to accept them.

This is because majority of the software made in the world, is purely English. Linux, Windows, take your pick - it can be any type of program. And so, languages like these won't be able to pick up the pace.

But what if a transpiler is made such that native-language programs turn into normal, English programs that are feasible?

That's where **Ajgar** comes in.

Ajgar is a transpiler that lets people program Python in Hindi, then process and run/save that code like normal Python.

Project Ajgar's aim is simple - Abide with NEP 2023, make a version of Python in Hindi, that is feasible for modern software too. The software is available as a [PyPI Package](#). Ajgar also fulfils NEP 2023's spirit for introducing coding and vocational training starting from Class 6 in local media. Ajgar lets it's users to program in pure, Devanagari Hindi. Then, the file can be transpiled and run/saved using Ajgar's Command Line Interface (Ajgar CLI).

Ajgar is the perfect tool for those who don't know English, but want to be a multilingual programmer and possibly even make a living.

Why can **Ajgar** be **accepted by firms**?

Unlike other attempts to make native-language programs like Kalam, OM Lang etc. Ajgar is different!

Instead of being a completely separate language, Ajgar is a Hindi-Python to Python transpiler, which means the programming language works the same way, only the interpreter is different.

This solves problems other programs have a low chance of overcoming:

- The English Bottleneck: Ajgar transpiles Hindi Python into English Python, which is how it runs the code. It can also save the English code.
- Team Collaboration & Scalability: Since Ajgar can transpile into English code, it will be easier for others to review Hindi Python after transpiling it.

Ajgar is also open-source: many versions of it can be created for other languages. View the GitHub Repository of Ajgar [here](#).

How does **Ajgar** transpile **Hindi-Python**?

Ajgar is able to transpile Hindi-Python to normal Python, mainly using these 3 features:

- Python supports Devanagari Scripts
- JSON Dictionary Mapping
- Python Tokenizer

In basic words, Ajgar takes a file in which Hindi-Python is there, reads the whole file, tokenizes it, searches for the tokens in a JSON, finds the values for those tokens, replaces the file's tokens with their values, and it does this sequence for the whole file. Every keyword, syntax, function, data type, everything.*

It then takes the modified file content, and executes that instead of what was originally in the file.

Ajgar can also save the modified, normal Python to a file of choice if needed.

All this can be done using the CLI.

*Ajgar v1.0.0 supports almost all Python syntax, but not Python modules yet.

The **Ajgar CLI** – Defaulted to **Hindi**.

The Ajgar software is available as a PyPI Package.

Normally, when a PyPI Package has a CLI, it uses a Python module called 'argparse' to desing the CLI's arguments, flags, etc. But in the case of Ajgar, that would simply defeat the whole point of Ajgar.

Argparse defaults its errors, messages, help menus and warnings in English, nullifying the main goal of Ajgar to keep the whole software accessible and usable in Hindi.

And so, the CLI for Ajgar is specially designed, not using Argparse.

However, an English CLI for Ajgar has also been kept alongwith the default Hindi CLI.

A tutorial for the Ajgar CLI is available [here](#)

An example **Ajgar** transpilation.

Here is an example Hindi-Python script and it's transpilation.

```
छापो("Namaste")
naam = पूछो("Aapka naam kya hai? ")
one = संख्या("1")
pi = दशमलव(22 / 7)
छापो(one, pi)
अगर naam == "Aarsh":
    छापो("Namaste, Aarsh Garg")
वरना:
    छापो("Namaste", naam)
केलिए i अंदर श्रेणी(1, 100):
    छापो(i)
loop = सत्य
जबतक loop:
    छापो("loop")
    loop = असत्य
निर्धारित addition(a: संख्या, b: संख्या):
    वापसी a + b
छापो(addition(1, 20))
छापो("Dhanyavad!")
```



```
print("Namaste")
naam = input("Aapka naam kya hai? ")
one = int("1")
pi = float(22 / 7)
print(one, pi)
if naam == "Aarsh":
    print("Namaste, Aarsh Garg")
else:
    print("Namaste", naam)
for i in range(1, 100):
    print(i)
loop = True
while loop:
    print("loop")
    loop = False
def addition(a: int, b: int):
    return a + b
print(addition(1, 20))
print("Dhanyavad!")
```

I'm not the **first to try to make this real.**

Developers on GitHub have tried to make the same software in other forms. For example, someone named Shravani Nimbolkar made the same software in early February 2022. Although her software was flawless and working properly, it had problems that even Ajgar v1.0.0 has, it seemed to be a discontinued project (no commits since 2022) and it was a EXE that was not easily accessible.

My software is a PyPI Package, easily accessible and being actively developed on.

Links and Referrals.

Ajgar v1.0.0+

- Ajgar PyPI - <https://pypi.org/project/ajgar>
- Ajgar GitHub - <https://github.com/AarshGarg2011Code/ajgar-v1.0.0>
- Ajgar Tutorial - <https://ag2cdev.pages.dev/ajgar/tutorials>

My Links

- Website - <https://ag2cdev.pages.dev>
- PyPI - <https://pypi.org/user/aarshgargpython>
- GitHub - <https://github.com/AarshGarg2011Code>
- Email - <mailto:aarshg.13@gmail.com>
- Phone - <tel:+918356949225>





THANK
YOU

AARSH GARG