

# Understanding Pythonic Idioms via Visualization

Jacquelyn Yapenare  
PNGUoT, PNG  
21302387jaya@student.pnguot.ac.pg

Kazumasa Shimari  
NAIST, Japan  
k.shimari@is.naist.jp

Brittany Reid  
NAIST, Japan  
brittany.reid@naist.ac.jp

Abuzo Sankwi  
PNGUoT, PNG  
sankwi.abuzo@pnguot.ac.pg

Benson Mirou  
PNGUoT, PNG  
benson.mirou@pnguot.ac.pg

Raula Gaikovina Kula  
Osaka University, Japan  
benson.mirou@pnguot.ac.pg

## Introduction

**Pythonic idioms** improve code **efficiency** and **readability** by leveraging Python's concise syntax and unique features.

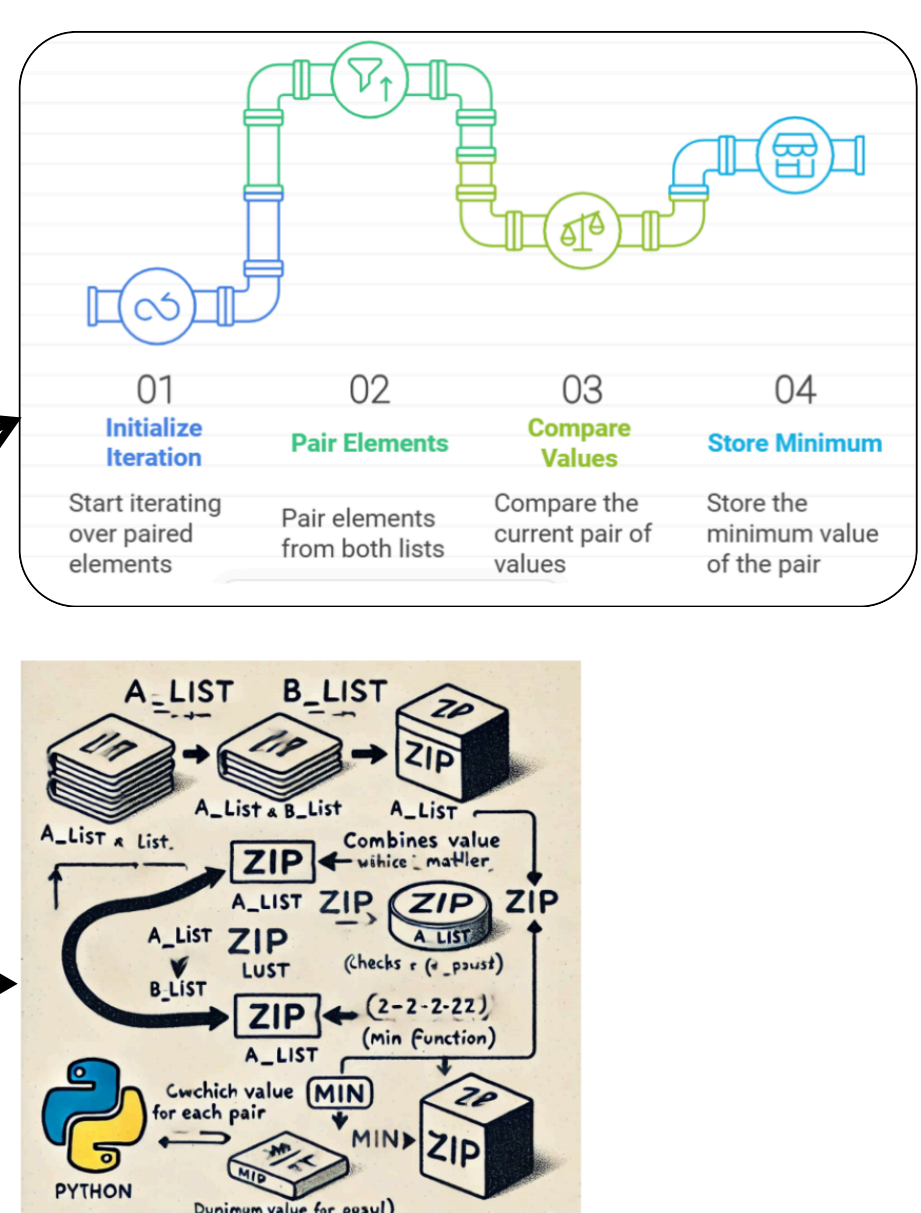
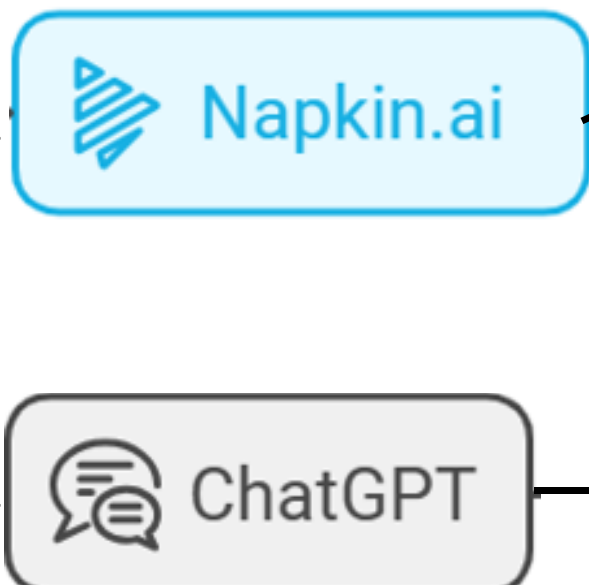
Their **compact** and **abstract nature** makes Pythonic idioms challenging for beginners and developers who are unfamiliar with or unsure of how to use them effectively.

## Objective

Explore **AI-powered tools**, such as **Napkin.ai** and **ChatGPT**, to determine if they can generate **visualizations** of Pythonic idioms that are **intuitive**.

**Pythonic Code Snippet**

```
Pythonic idiom: zip  
  
for a, b in zip(a_list ,  
              b_list):  
    result = min(a, b)
```



## Methodology

**Nine Pythonic idiom** snippets were input into **ChatGPT** and **Napkin.ai** to generate visualizations, which were then **evaluated based on their comprehensibility** to a novice Python programmer.

## Preliminary Analysis

Pythonic Idiom	ChatGPT	Napkin.ai
List Comprehension	2	4
Dict Comprehension	2	4
Generator expression	2	3
yield	2	3
lambda	2	5
collections.defaultdict	1	3
collections.dequeue	1	4
zip	2	4
itertools	1	5

**Rating Key**

1 = Do not understand at all  
2 = Barely understand  
3 = Somewhat understand  
4 = Mostly understand  
5 = Completely understand

The ratings of understandability show that AI tools can assist with comprehension, with **Napkin.ai proving more efficient** due to its ability to **generate clear** and more **intuitive visualizations** without needing a prompt.

## Future Work

Future work will involve generating more visualizations and **analyzing** their **accuracy in capturing Pythonic syntax**, alongside a **study testing developer responses** to visualizations for understanding Pythonic idioms.