

# ความสำคัญและหลักการพื้นฐานของไลบรารี NumPy

ผศ. ทิพย์วรรณฟูเฟื่อง  
สาขาเทคโนโลยีสารสนเทศ คณะวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ  
มหาวิทยาลัยราชภัฏรำไพพรรณี

ในปัจจุบัน การคำนวณเชิงตัวเลขและการประมวลผลข้อมูลปริมาณมหาศาลถือเป็นหัวใจสำคัญของหลายสาขา ไม่ว่าจะเป็น วิทยาศาสตร์ข้อมูล (Data Science) การเรียนรู้ของเครื่อง (Machine Learning) หรือแม้แต่งานวิจัยด้านวิทยาศาสตร์กายภาพและชีววิทยา สิ่งเหล่านี้ต้องอาศัยเครื่องมือที่มีประสิทธิภาพสูงเพื่อจัดการกับข้อมูลที่ซับซ้อนและปริมาณมาก ในระบบนิเวศของภาษา Python ไม่มีไลบรารีใดที่จะมีบทบาทโดดเด่นไปกว่า NumPy (Numerical Python) ซึ่งเปรียบเสมือน “กระดูกสันหลัง” ของงานคำนวณเชิงวิทยาศาสตร์

## NumPy คืออะไร

NumPy คือไลบรารีมาตรฐานสำหรับการคำนวณเชิงวิทยาศาสตร์ในภาษา Python หัวใจของ NumPy อยู่ที่โครงสร้างข้อมูล ndarray (N-dimensional array) ซึ่งเป็นอาร์เรย์หลายมิติที่ถูกออกแบบมาเพื่อให้

- ประสิทธิภาพสูง (High Performance) การประมวลผลเร็วกว่า Python list หลายเท่า เนื่องจากเก็บข้อมูลแบบต่อเนื่องในหน่วยความจำ
- ความเป็นเนื้อเดียวกัน (Homogeneity): ข้อมูลใน ndarray ต้องมีชนิดเดียวกัน เช่น ทั้งหมดเป็น integer หรือ float ซึ่งช่วยเพิ่มประสิทธิภาพด้านการประมวลผล
- ฟังก์ชันในตัวที่หลากหลาย: เช่น การคำนวณทางสถิติ (Mean, Variance, Standard Deviation), การคำนวณพีชคณิตเชิงเส้น (Linear Algebra) และการสร้างเมทริกซ์

แม้ Python จะใช้งานง่าย แต่การจัดการข้อมูลเชิงตัวเลขจำนวนมากโดยใช้ list นั้นไม่มีประสิทธิภาพ ด้วยเหตุผลหลักสองประการ

### 1. การจัดเก็บหน่วยความจำ (Memory Management)

Python list เป็นอาร์เรย์ของตัวชี้ (Pointers) ที่ชี้ไปยังวัตถุต่างๆ ซึ่งอาจกระจายอยู่ในหน่วยความจำ ทำให้การเข้าถึงข้อมูลช้าลง ในทางกลับกัน NumPy ndarray จะเก็บข้อมูลชนิดเดียวกันไว้ในพื้นที่หน่วยความจำที่

ต่อเนื่องกัน (Contiguous Memory Block) ทำให้ซีพียู (CPU) สามารถเข้าถึงและประมวลผลข้อมูลได้อย่างรวดเร็ว

## 2. การประมวลผลแบบ Vectorization

การดำเนินการทางคณิตศาสตร์กับ list จำเป็นต้องใช้การวนซ้ำ (Loop) ที่ละสมาชิก แต่ NumPy ใช้หลักการที่เรียกว่า Vectorization ซึ่งเป็นการประมวลผลข้อมูลทั้งอาเรย์ในคราวเดียวผ่านโค้ดเบื้องหลังที่เขียนด้วยภาษา C หรือ Fortran ซึ่งเร็วกว่าการทำงานในระดับ Python Interpreter หลายเท่าตัว

ด้วยเหตุนี้ NumPy จึงถูกสร้างขึ้นเพื่อเป็นพื้นฐานให้กับไลบรารีอื่น ๆ เช่น Pandas SciPy Matplotlib Scikit-learn และ TensorFlow ซึ่งทั้งหมดพึ่งพา ndarray ของ NumPy เป็นหลัก

### การเริ่มต้นใช้งาน NumPy

การใช้งาน NumPy สามารถเริ่มต้นได้ง่ายๆ ผ่านขั้นตอนต่อไปนี้

การติดตั้ง (Installation)

```
pip install numpy
```

การนำเข้า (Import)

```
import numpy as np
```

การสร้าง NumPy Array

```
# 1 มิติ
list_a = [1, 2, 3, 4, 5]
np_a = np.array(list_a)

# 2 มิติ
list_b = [[1, 2, 3], [4, 5, 6]]
np_b = np.array(list_b)
```

ผลลัพธ์

```
อาเรย์ 1 มิติ: [1 2 3 4 5]
อาเรย์ 2 มิติ:
[[1 2 3]
 [4 5 6]]
```

การตรวจสอบคุณสมบัติของอาเรย์

```
print(np_b.ndim)    # จำนวนมิติ
print(np_b.shape)   # รูปร่าง (แถว, คอลัมน์)
print(np_b.dtype)   # ชนิดข้อมูล
print(np_b.size)    # จำนวนสมาชิก
```

การคำนวณทางคณิตศาสตร์ (Vectorized Operations) นี่คือนวัตกรรมที่สุดของ NumPy สังเกตว่าเราสามารถดำเนินการกับทุกสมาชิกในอาเรย์ได้ทันที โดยไม่ต้องเขียน for loop

```
arr1 = np.array([10, 20, 30])
arr2 = np.array([2, 5, 6])

print(arr1 + arr2)    # การบวก
print(arr1 - arr2)    # การลบ
print(arr1 * arr2)    # การคูณ
print(arr1 / arr2)    # การหาร
print(arr1 * 10)      # Broadcasting
```

ผลลัพธ์

```
[12 25 36]
[ 8 15 24]
[ 20 100 180]
[5.  4.  5.]
[100 200 300]
```

การดำเนินการกับค่าคงที่ (Broadcasting) จากตัวอย่างเป็นการคูณทุกสมาชิกด้วย 10 การคำนวณนี้ใช้เวลาเพียงเสี้ยววินาที แม้ว่าจะมีสมาชิกหลายล้านตัวก็ยังรวดเร็ว เนื่องจากการทำงานแบบ Vectorization

ตัวอย่างการใช้งาน NumPy ในงานจริง

1. การประมวลผลภาพ (Image Processing)

ภาพดิจิทัลคืออาเรย์ของค่าพิกเซล การใช้ NumPy ทำให้สามารถปรับสี แปลงขนาด หรือวิเคราะห์คุณสมบัติของภาพได้อย่างมีประสิทธิภาพ

2. การคำนวณทางวิศวกรรม

การแก้สมการเชิงเส้น การคำนวณ Eigenvalue/Eigenvector และการจำลองทางฟิสิกส์สามารถทำได้อย่างง่ายดายด้วย NumPy

3. งาน Machine Learning

ไลบรารีเช่น Scikit-learn และ TensorFlow ใช้ NumPy array ในการประมวลผลข้อมูลสำหรับการฝึกโมเดล (Training) และการทดสอบ (Testing)

4. การวิเคราะห์ข้อมูลสถิติ

NumPy มีฟังก์ชันสำหรับหาค่าเฉลี่ย (Mean), ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation) และค่าทางสถิติอื่น ๆ ที่จำเป็นสำหรับ Data Analysis

## บทสรุป

NumPy ไม่ได้เป็นเพียงไลบรารีทางเลือก แต่เป็นองค์ประกอบที่ขาดไม่ได้ในงานคำนวณเชิงตัวเลขและวิทยาศาสตร์ด้วยภาษา Python ด้วยโครงสร้างข้อมูล ndarray ที่ถูกออกแบบมาเพื่อประสิทธิภาพสูงสุด ทั้งในด้านความเร็วและหน่วยความจำ พร้อมด้วยฟังก์ชันทางคณิตศาสตร์ที่ครอบคลุม ทำให้ NumPy เป็นเครื่องมือที่ทรงพลังและเป็นจุดเริ่มต้นที่สำคัญสำหรับทุกคนที่ต้องการทำงานในสายงาน Data Science วิศวกรรมข้อมูล และการวิจัยเชิงคำนวณ การมีความเข้าใจในหลักการทำงานและสามารถใช้งาน NumPy ได้อย่างคล่องแคล่วจึงเป็นทักษะพื้นฐานที่จะต่อยอดไปสู่การใช้ไลบรารีขั้นสูง เช่น Pandas TensorFlow และ PyTorch ได้อย่างมั่นคง