

LECTURE - 00 INTRODUCTION

What You Will Learn

- Python history
- Why Python
- Install Python
- "Hello World"
- VSCode
- GitHub
- `print()` command /format

ABOUT ME



Anirach Mingkhwan 

King Mongkut's University of Technology North Bangkok, Thailand
Verified email at fitm.kmutnb.ac.th

AI Network Technologies Information Science Natural Inspired Computing Mobile Computing

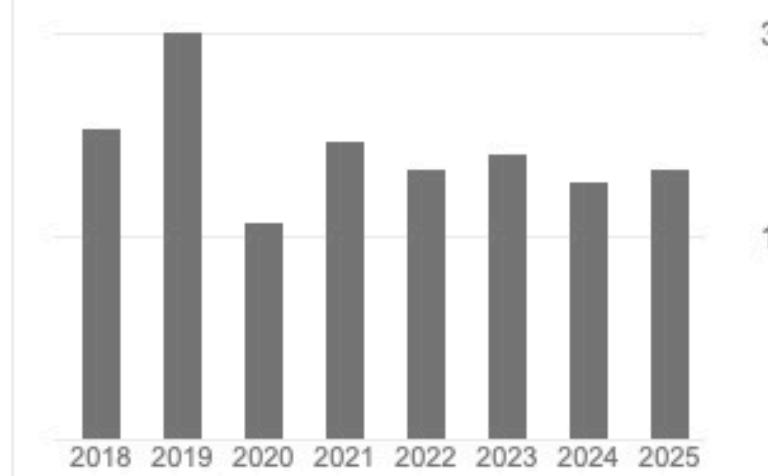
TITLE  

	CITED BY	YEAR
Optimizing knowledge management using knowledge map J Watthananon, A Mingkhwan Procedia Engineering 32, 1169-1177	56	2012
Dynamic service composition in home appliance networks A Mingkhwan, P Fergus, O Abuelmaatti, M Merabti, B Askwith, ... Multimedia Tools and Applications 29, 257-284	44	2006
Disus: Mobile ad hoc network unstructured services P Fergus, A Mingkhwan, M Merabti, M Hanneghan Personal Wireless Communications: IFIP-TC6 8th International Conference, PWC ...	22	2003
Implicit functionality: Dynamic services composition for home networked appliances A Mingkhwan, P Fergus, O Abuelmaatti, M Merabti 2004 IEEE International Conference on Communications (IEEE Cat. No ...)	20	2004
Semantic searching it careers concepts based on ontology P Singto, A Mingkhwan Journal of Advanced Management Science 1 (1), 102-106	19	2013
Distributed emergent semantics in P2P networks P Fergus, A Mingkhwan, M Merabti, M Hanneghan Proc. of the Second IASTED International Conference on Information and ...	18	2003
A semantic Framework for self-adaptive networked appliances P Fergus, M Merabti, MB Hanneghan, A Taleb-Bendiab, A Mingkhwan Second IEEE Consumer Communications and Networking Conference, 2005. CCNC ...	17	2005
Multi-dimensional visualization for network forensic analysis N Promrit, A Mingkhwan, S Simcharoen, N Namvong 7th International Conference on Networked Computing, 68-73	15	2011

 FOLLOWING

Cited by [VIEW ALL](#)

	All	Since 2020
Citations	508	121
h-index	12	5
i10-index	14	1



Co-authors [EDIT](#)

	Madjid Merabti University of Sharjah, UAE >
	Paul Fergus Professor of Applied Machine Le... >
	Martin Hanneghan Subject Head for Computer Scie... >
	Phayung Meesad King Mongkut's University of Tec... >
	Julaluk Watthananon Rajamangala University of Techn... >
	Herwig Unger FernUniversität in Hagen >

anirach.m@fitm.kmutnb.ac.th

GRADE CATEGORIES

Grade categories		
Grade category*	Default points*	X
Class Participation	5	X
20 / 100		
Grade category*	Default points*	X
HW-01	5	X
5 / 100		
Grade category*	Default points*	X
HW-02	5	X
5 / 100		
Grade category*	Default points*	X
HW-03	5	X
5 / 100		
Grade category*	Default points*	X
HW-04	5	X
5 / 100		
Grade category*	Default points*	X
HW-05	5	X
5 / 100		
Grade category*	Default points*	X
Project	10	X
7 / 100		
Grade category*	Default points*	X
Midterm	30	X
7 / 100		
Grade category*	Default points*	X
Final	30	X

WHY NOT AI ?



🧠 ทำไมต้องเรียนเขียนโปรแกรม ในเมื่อ AI ก็ทำให้ได้?

เพราะเครื่องคิดเลขไม่ได้มีวิชาคณิตศาสตร์

Google ไม่ได้มีความสามารถในการคิด

ChatGPT ก็จะไม่มีการเขียนโปรแกรม

เครื่องมือเปลี่ยนได้

แต่กระบวนการคิดไม่เปลี่ยน

เครื่องคิดเลขให้คำตอบกับคุณได้

แต่คุณยังต้องรู้ว่าจะถามคำน้ออะไร

AI ก็เช่นกัน

มันไม่ใช่แค่การพิมพ์โค้ด

แต่คือการเข้าใจตรรกะ โครงสร้าง ลำดับงาน ข้อจำกัด และสถาปัตยกรรมของระบบ

AI มาเพื่อช่วย

แต่ถ้าคุณไม่รู้วิธีคิดแบบนักพัฒนา

คุณก็แค่ปล่อยให้เครื่องจักรเดาแทนคุณ

เรียนเขียนโปรแกรมไม่ใช่ เพราะ AI ทำไม่ได้

แต่เพราะคุณจะรู้ว่าเมื่อไหร่มัน “ไม่ควร” ทำแทนคุณ

💬 ถ้า AI หายไปพรุ่งนี้ อาชีพของคุณจะอยู่รอดไหม?

Example: Sentiment Classification

Software 1.0

```
python Copy
def simple_sentiment(review: str) -> str:
    """Return 'positive' or 'negative' based on a tiny keyword lexicon."""
    positive = {
        "good", "great", "excellent", "amazing", "wonderful", "fantastic",
        "awesome", "loved", "love", "like", "enjoyed", "superb", "delightful"
    }
    negative = {
        "bad", "terrible", "awful", "poor", "boring", "hate", "hated",
        "dislike", "worst", "dull", "disappointing", "mediocre"
    }

    score = 0
    for word in review.lower().split():
        w = word.strip(".,!?:;")      # crude token clean-up
        if w in positive:
            score += 1
        elif w in negative:
            score -= 1

    return "positive" if score >= 0 else "negative"
```

Software 2.0

10,000 positive examples
10,000 negative examples
encoding (e.g. bag of words)

train binary classifier

parameters

Software 3.0

You are a sentiment classifier. For every review that appears between the tags

<REVIEW> ... </REVIEW>, respond with **exactly one word**, either POSITIVE or NEGATIVE (all-caps, no punctuation, no extra text).

Example 1

<REVIEW>I absolutely loved this film—the characters were engaging and the ending was perfect.</REVIEW>

POSITIVE

Example 2

<REVIEW>The plot was incoherent and the acting felt forced; I regret watching it.</REVIEW>

NEGATIVE

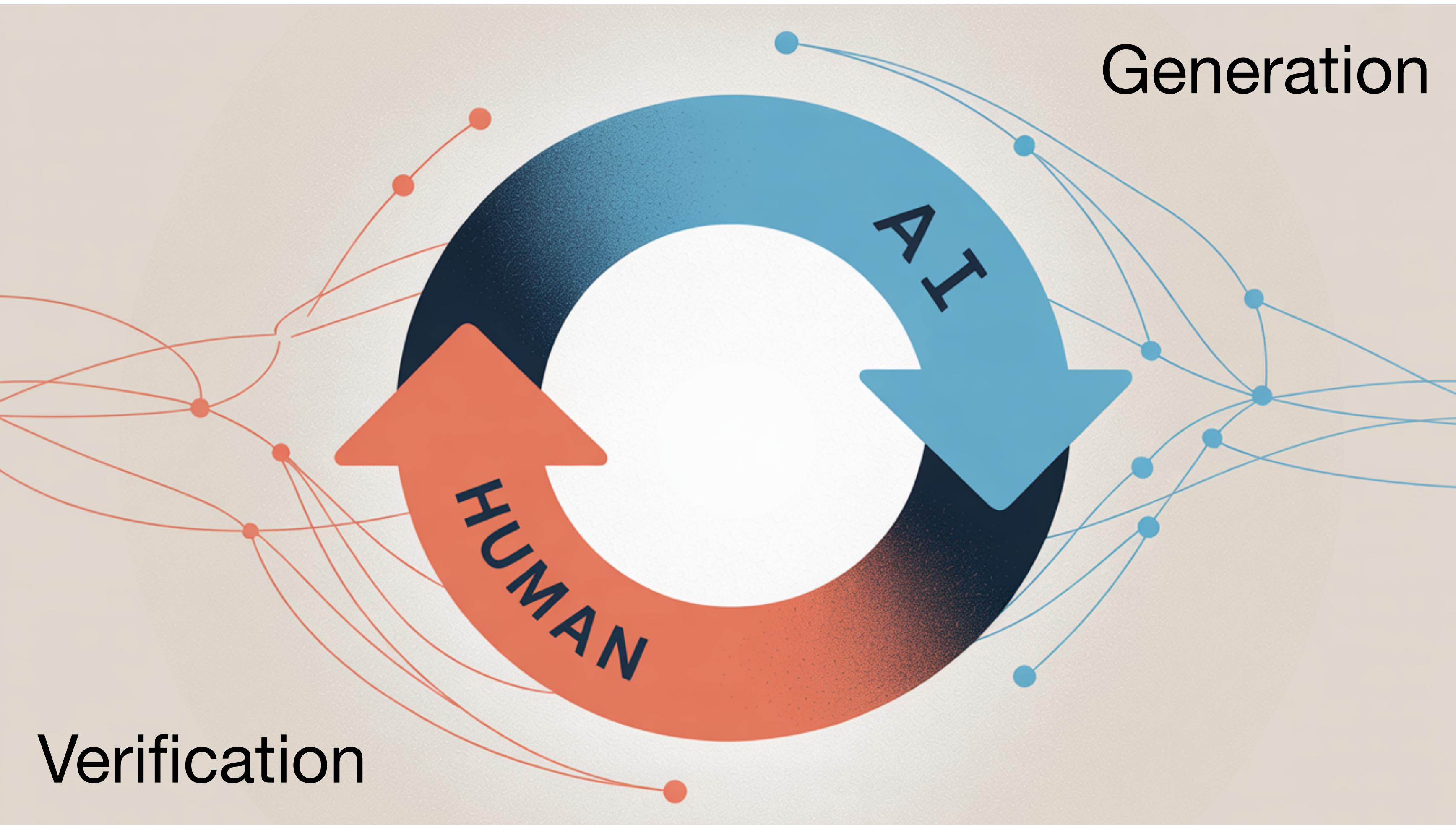
Example 3

<REVIEW>An energetic soundtrack and solid visuals almost save it, but the story drags and the jokes fall flat.</REVIEW>

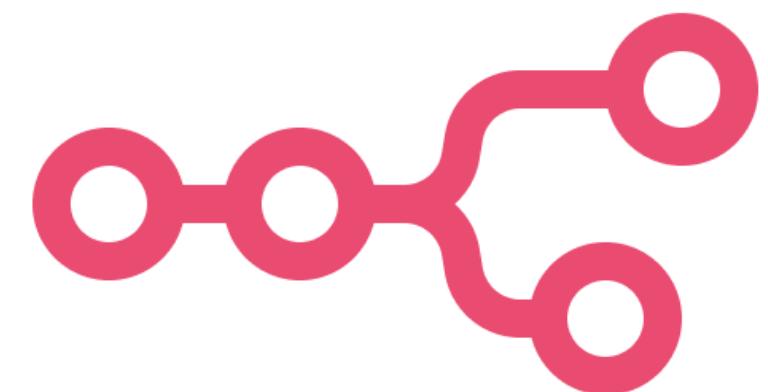
NEGATIVE

Now classify the next review.

Software 3.0 Coding Loop



Vibe Coding



SOURCE CONTROL

CHANGES

Message (⌘Enter to commit or...)

✓ Commit

Changes

enhanced_patient_da... risk_trends_verification.sql

GRAPH

- ChartDone Anirach Ming... @main
- Fix chart.. Anirach Mingkhwan
- Done login Anirach Mingkhwan
- Done Migrate.. Anirach Mingkhwan
- Migrate from Supabase to PgSQL An...
- generate data for multiple view lab r...
- Try fixing data Anirach Mingkhwan
- Before fix visit and results Anirach Mi...
- The Chat Scroll is working..Nlce Anir...
- The chat window working. Anirach Mi...
- Visual edit in Lovable gpt-engineer-a...
- Run SQL migration gpt-engineer-app[...]
- Revert AI chat changes gpt-engineer-...
- Make patient detail page full frame g...
- Fix patient data and AI chat gpt-engi...

enhanced_patient_data.sql U RiskChart.tsx X risk_trends_verification.s

```

src > components > RiskChart.tsx > ...
16   export const RiskChart = ({ data, title }: RiskChartProps) => {
17     <NS CLASSNAME="text--bold font-semibold text--foreground">{title}</NS>
18     <div className="flex items-center space-x-2">
19       {getTrendIcon()}
20       <span className={`font-semibold ${getTrendColor()}`}>
21         {trend > 0 ? '+' : ''}{trend.toFixed(1)}%
22       </span>
23     </div>
24   </div>

25   <div className="h-80">
26     <ResponsiveContainer width="100%" height="100%">
27       <LineChart data={data} margin={{ top: 20, right: 30, left: 70, bottom: 60 }}>
28         <CartesianGrid strokeDasharray="3 3" className="opacity-30" />
29         <XAxis
30           dataKey="date"
31           tick={{ fontSize: 12, fill: 'hsl(var(--muted-foreground))' }}
32           angle=-45
33           textAnchor="end"
34           height={80}
35           interval={0}
36         />
37         <YAxis
38           tick={{ fontSize: 12, fill: 'hsl(var(--muted-foreground))' }}
39           label={{
40             value: 'ASCVD Risk (%)',
41             angle: -90,
42             position: 'insideLeft',
43           }}
44         />
45       </LineChart>
46     </ResponsiveContainer>
47   </div>
48 
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Michael Brown	0	2
Robert Wilson	0	2
Sarah Johnson	0	2
(5 rows)		

Copilot Copilot

CHAT

151.177.14/ASCVD -c "SELECT p.name, COUNT(DISTINCT v.id) as visit_count, COUNT(DISTINCT lr.id) as lab_count FROM patients p LEFT JOIN visits v ON p.id = v.patient_id LEFT JOIN lab_results lr ON p.id = lr.patient_id GROUP BY p.id, p.name ORDER BY p.name;"

Let me also check that the API server is still running and restart it if needed:

curl -s http://localhost:3001/api/patients | head -20

The API server seems to be down. Let me restart it:

cd api && npm start

cd api && node server.js

node server.js
Server running on port 3001
Database connected successfully

RiskChart.tsx:61-87

GitHub Copilot

1 file changed
RiskChart.tsx src/components

Add Context... RiskChart.tsx Current file

Edit files in your workspace in agent mode

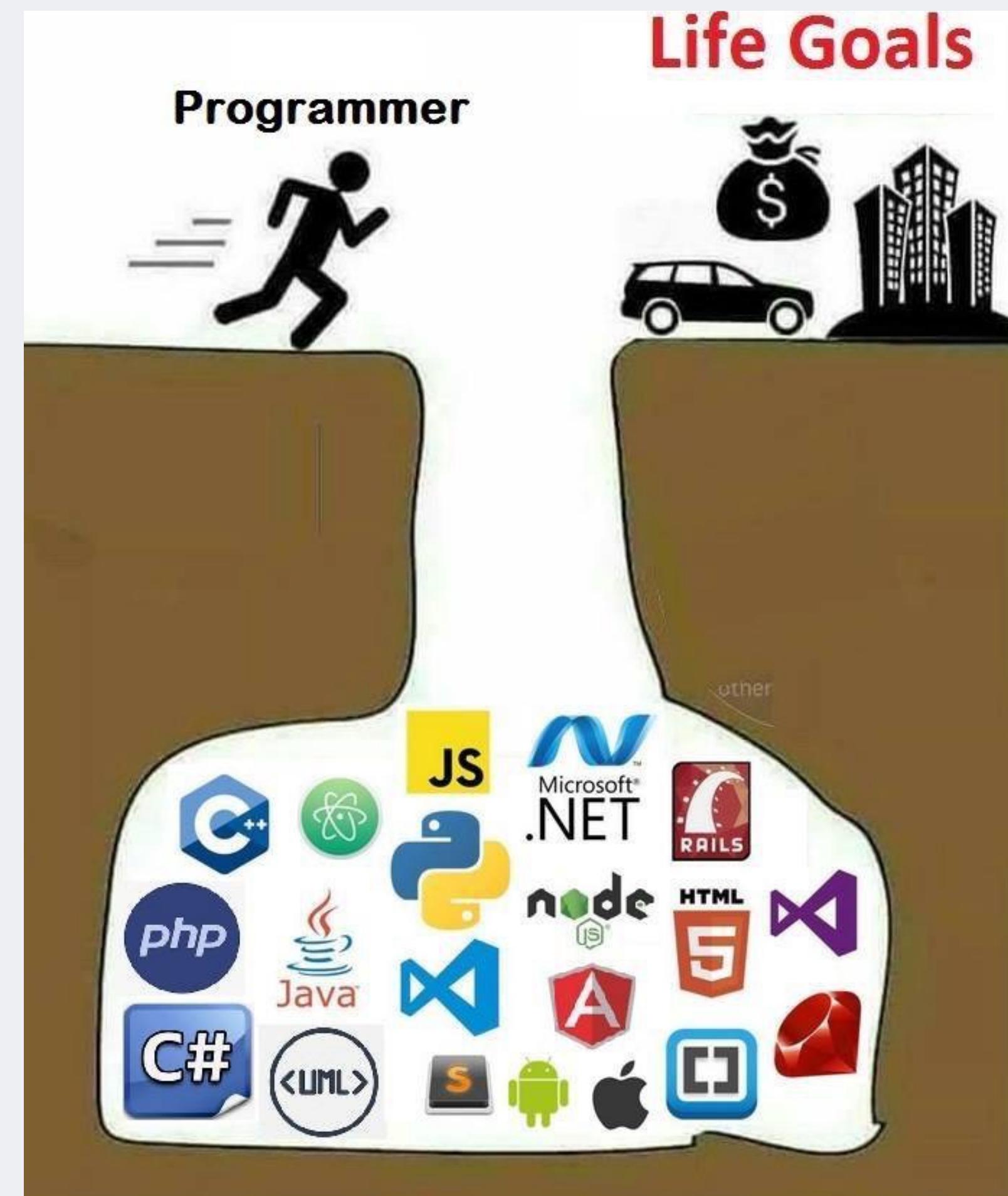
Agent Claude Sonnet 4 (Preview)

main* 0 △ 0 Connect Git Graph Select Postgres Server Ln 1, Col 1 Spaces: 2 UTF-8 LF {} TypeScript JSX Go Live Prettier

PROGRAMMER



PROGRAMMER



PROGRAMMER



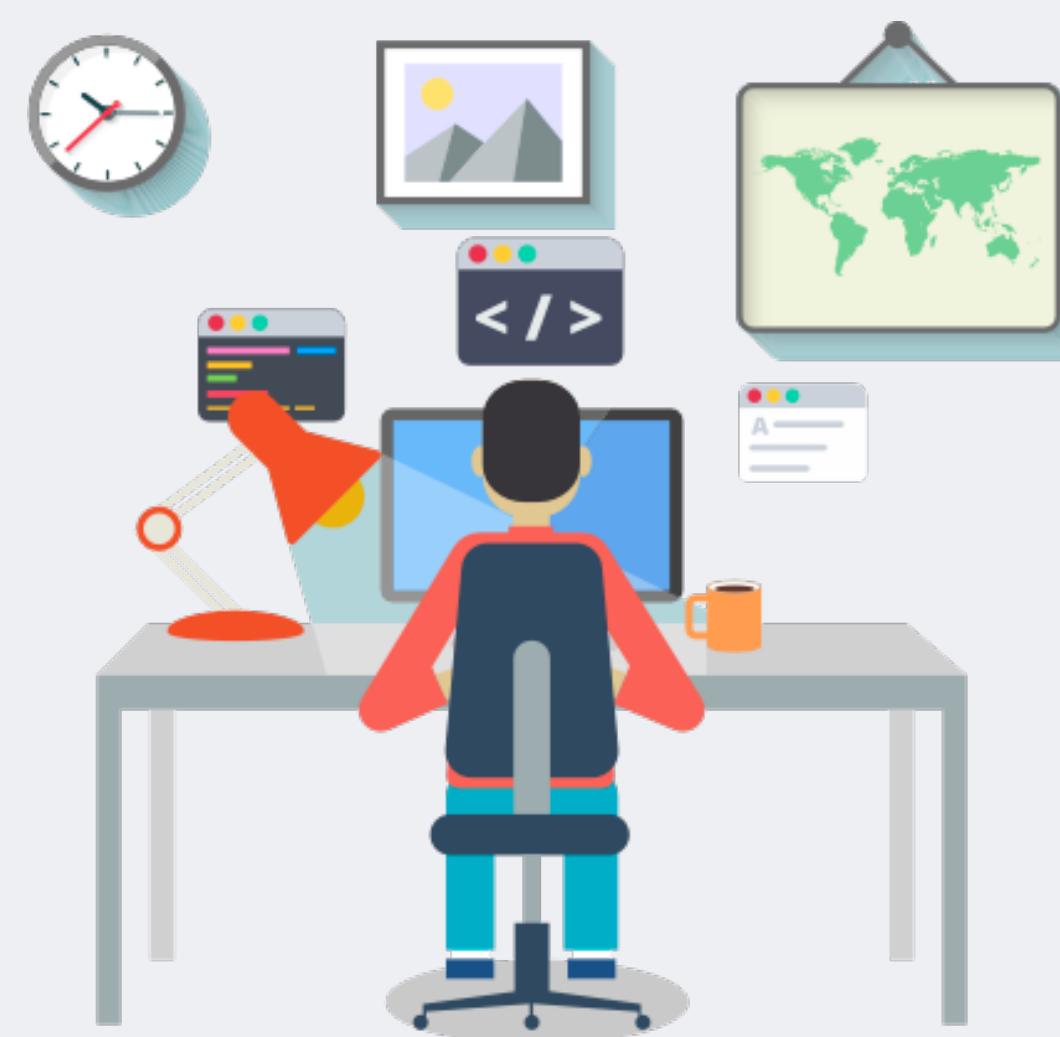
pro·gram·mer

/'prō,gramər/

noun

noun: **programmer**; plural noun: **programmers**; noun: **programer**; plural noun: **programers**

a person who writes computer programs.
"computer programmers and analysts"



THE WAY OF THE PROGRAM

"The single most important skill for a computer scientist is problem solving."

Problem-solving means formulating problems, thinking creatively about solutions, and expressing a solution clearly and accurately. As it turns out, learning to program is an excellent opportunity to practice problem-solving skills. That's why it is called "The Way of the Program".

CONCEPT

แนวคิดสำคัญ:

- อัลกอริทึม (Algorithm): ขั้นตอนการแก้ปัญหาอย่างเป็นลำดับ
- ชอร์สโค้ด (Source Code): คำสั่งที่มนุษย์เขียนและสามารถอ่านได้
- การคอมไพล์/การแปลผล (Compilation/Interpretation): กระบวนการแปลงชอร์สโค้ดให้กลายเป็นรหัสเครื่องที่คอมพิวเตอร์สามารถประมวลผลได้
- ภาษาการเขียนโปรแกรม (Programming Languages): ภาษาที่ใช้ในการเขียนโปรแกรม เช่น Python, Java, C++

ALGORITHM

อัลกอริทึมคือขั้นตอนหรือชุดคำสั่งที่มีการจัดเรียงอย่างชัดเจน เพื่อทำงานใดๆ หนึ่งหรือแก้ไขปัญหาที่เจาะจง เปรียบได้กับสูตรอาหารในตำรา ที่ระบุขั้นตอนแต่ละขั้นเพื่อให้ได้ผลลัพธ์ตามที่ต้องการ เช่นเดียวกับสูตรอาหารที่มีวิธีประกอบอาหารอย่างละเอียด อัลกอริทึมก็มีขั้นตอนที่ชัดเจนเพื่อบรรลุเป้าหมายของโปรแกรม

ALGORITHM

ตัวอย่าง: อัลกอริทึมในการทำแซนด์วิช

1. เตรียมวัตถุดิบและอุปกรณ์: ขนมปัง เนย ชีส มีด และจาน
2. ทาเนย: ใช้มีดทาเนยลงบนด้านหนึ่งของขนมปังแต่ละแผ่น
3. วางชีส: วางชีสลงบนด้านที่ทาเนยของแผ่นขนมปังแผ่นหนึ่ง
4. ประกับขนมปัง: นำขนมปังอีกแผ่นประกับด้านบน โดยให้ด้านที่ทาเนยหันเข้าด้านใน
5. เสิร์ฟ: วางแซนด์วิชลงบนจานและพร้อมเสิร์ฟ

ALGORITHM

ในบริบทของการเขียนโปรแกรม อัลกอริทึมมีแนวทางคล้ายกัน:

ตัวอย่าง: อัลกอริทึมในการบวกเลขสองจำนวน

1. **เริ่มต้น:** เริ่มกระบวนการ
2. **รับข้อมูลเข้า:** รับตัวเลขสองจำนวนจากผู้ใช้
3. **ประมวลผล:** ทำการบวกตัวเลขทั้งสอง
4. **แสดงผลลัพธ์:** แสดงผลรวมที่ได้
5. **สิ้นสุด:** จบกระบวนการ

ALGORITHM

Here is how this might look in Python:

```
1 #Add2Numbers.py
2 # Step 1: Start
3 # Step 2: Input
4 number1 = int(input("Enter the first number: "))
5 number2 = int(input("Enter the second number: "))

6
7 # Step 3: Process
8 sum = number1 + number2

9
10 # Step 4: Output
11 print("The sum is:", sum)

12
13 # Step 5: End
```

Listing 1.1: Example: Algorithm to Add Two Numbers

PROGRAMMING LANGUAGE

Natural languages are the **languages people speak**, such as English, Spanish, and French. They were not designed by people (although people try to impose some order on them); they evolved naturally.

Formal languages are **languages that are designed by people for specific applications**. For example, the notation that mathematicians use is a formal language that is particularly good at denoting relationships among numbers and symbols. Chemists use a formal language to represent the chemical structure of molecules. And most importantly:

Programming languages are **formal languages that have been designed to express computations**.

HIGH-LEVEL LANGUAGE

1. ภาษาในระดับสูง (High-Level Languages):

- ภาษาเหล่านี้มีลักษณะใกล้เคียงกับภาษามนุษย์ และอ่านเขียนได้ง่าย
- ไม่จำเป็นต้องมีความรู้เฉพาะทางเกี่ยวกับสถาปัตยกรรมคอมพิวเตอร์
- ตัวอย่างเช่น Python, Java, C++, JavaScript, และ Ruby

ตัวอย่างของภาษาในระดับสูง (Python):

```
1 #HelloWorld.py
2 print("Hello, World!")
```

Listing 1.5: Example of High-Level Language (Python)

LOW-LEVEL LANGUAGE

2. ภาษาในระดับต่ำ (Low-Level Languages):

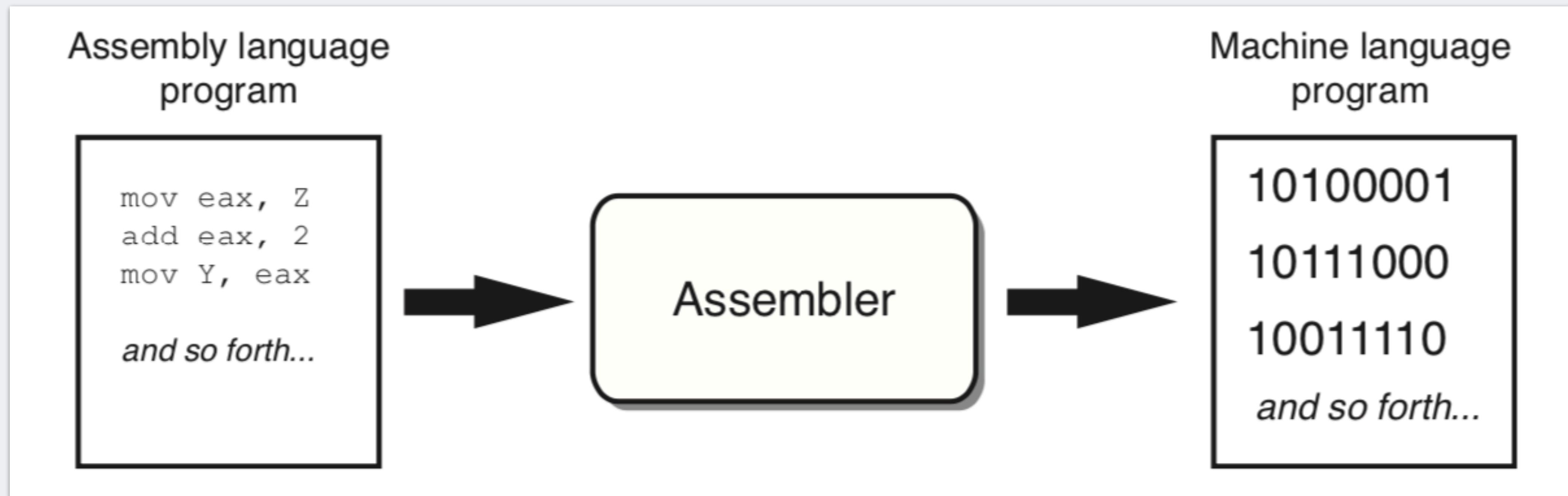
- ภาษาเหล่านี้มีลักษณะใกล้เคียงกับภาษาเครื่อง (binary) และอ่านเขียนได้ยาก
- มีความเกี่ยวข้องโดยตรงกับสถาปัตยกรรมของเครื่องคอมพิวเตอร์
- ตัวอย่างเช่น ภาษาแอสเซมบลี (Assembly) และรหัสเครื่อง (Machine Code)

ตัวอย่างของภาษาในระดับต่ำ (Assembly):

```
1 MOV AX, 4C00h  
2 INT 21h
```

Listing 1.6: Example of High-Level Language (Assembly)

ASSEMBLY LANGUAGE



low-level language

The first assembly language was most likely that developed in the 1940s at Cambridge University for use with a historic computer known as the EDSAC

SOURCE CODE

1.1.2 ชอร์สโค้ด (Source Code)

ชอร์สโค้ดหมายถึงชุดคำสั่งและข้อความที่เขียนโดยโปรแกรมเมอร์โดยใช้ภาษาการเขียนโปรแกรม คำสั่งเหล่านี้สามารถอ่านเข้าใจได้โดยมนุษย์ ซึ่งแตกต่างจากการหัสรเครื่องที่อยู่ในรูปแบบเลขฐานสองซึ่งสามารถอ่านได้เฉพาะโดยคอมพิวเตอร์เท่านั้น

ประเด็นสำคัญเกี่ยวกับชอร์สโค้ด:

1. อ่านได้โดยมนุษย์: ชอร์สโค้ดเขียนด้วยภาษาการเขียนโปรแกรมที่มีลักษณะคล้ายภาษาบ้านๆ เช่น Python, Java และ C++ ซึ่งเป็นภาษาที่นิยมใช้ในการเขียนชอร์สโค้ด
2. คำสั่ง: โค้ดประกอบด้วยชุดคำสั่งที่บอกให้คอมพิวเตอร์ดำเนินการตามที่กำหนด ตั้งแต่คำสั่งง่าย ๆ เช่น การบวกเลข ไปจนถึงกระบวนการซับซ้อน เช่น การรันเว็บเซิร์ฟเวอร์
3. การแก้ไขและปรับปรุง: โปรแกรมเมอร์สามารถแก้ไขและปรับปรุงชอร์สโค้ดได้อย่างง่ายดาย ทำให้สามารถพัฒนาและดูแลระบบซอฟต์แวร์อย่างต่อเนื่อง

SOURCE CODE

```
1 #Add2Numbers.py
2 # This is a comment explaining that the following lines of code
   will add two numbers
3
4 # Input: Asking the user to enter two numbers
5 number1 = int(input("Enter the first number: "))
6 number2 = int(input("Enter the second number: "))
7
8 # Process: Adding the two numbers
9 sum = number1 + number2
10
11 # Output: Displaying the result
12 print("The sum is:", sum)
```

คำอธิบาย:

- คำอธิบาย (Comments): บรรทัดที่ขึ้นต้นด้วยเครื่องหมาย # เป็นคำอธิบาย ซึ่งคอมพิวเตอร์จะไม่ประมวลผล แต่ช่วยให้โปรแกรมเมอร์เข้าใจได้ง่ายขึ้น
- Input: `input("Enter the first number: ")` เป็นคำสั่งให้ผู้ใช้ป้อนตัวเลข และ `int()` ใช้แปลงค่าที่รับเข้ามาให้เป็นจำนวนเต็ม
- ตัวแปร: `number1` และ `number2` เป็นตัวแปรที่ใช้เก็บค่าตัวเลขที่ผู้ใช้ป้อนเข้ามา
- กระบวนการ: `sum = number1 + number2` เป็นขั้นตอนการคำนวณผลรวมของสองตัวเลข
- การแสดงผล: `print("The sum is:", sum)` ใช้แสดงผลลัพธ์ให้ผู้ใช้เห็น

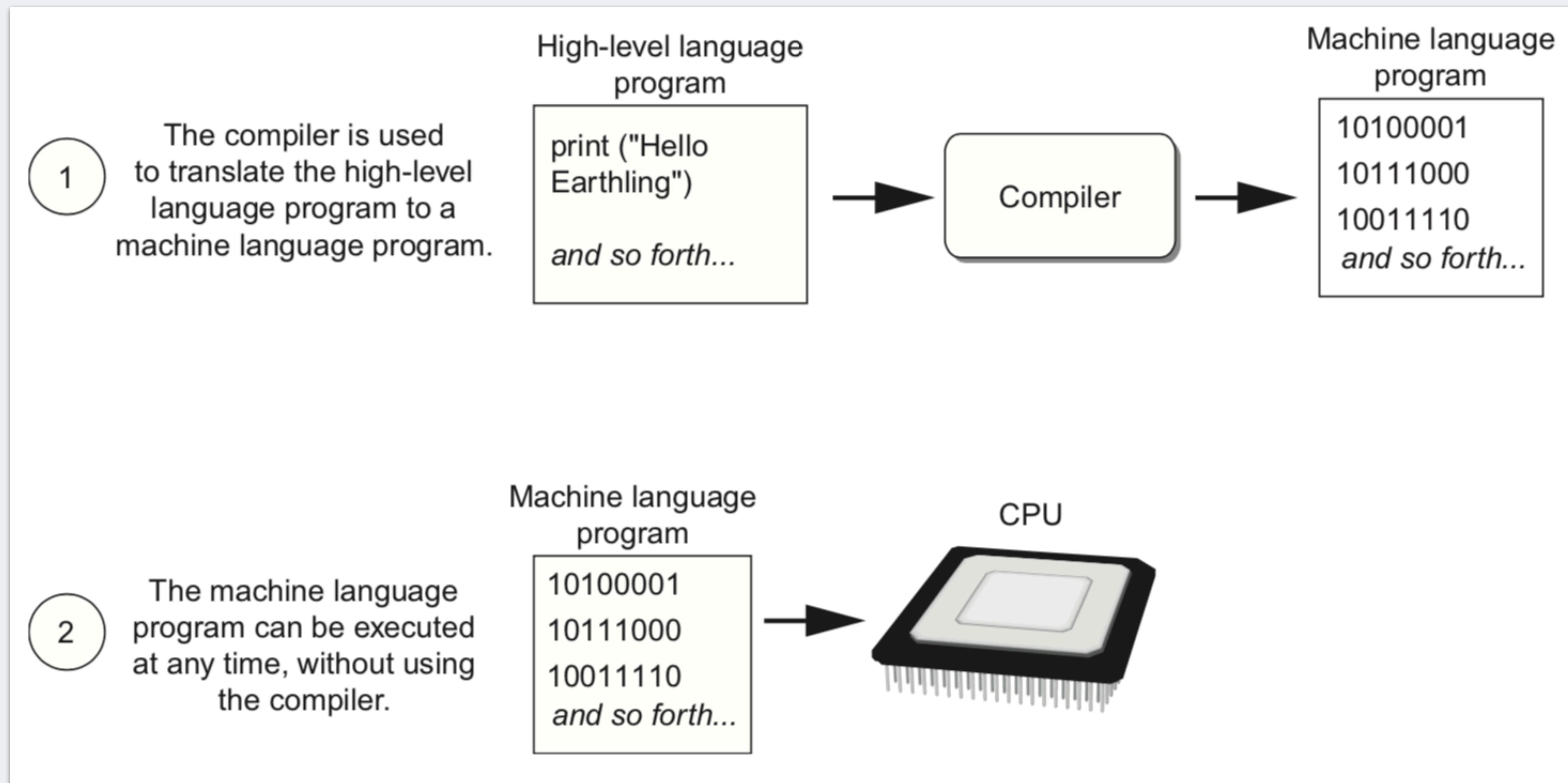
SOURCE CODE

เหตุผลที่ชอร์สโค้ดมีความสำคัญ:

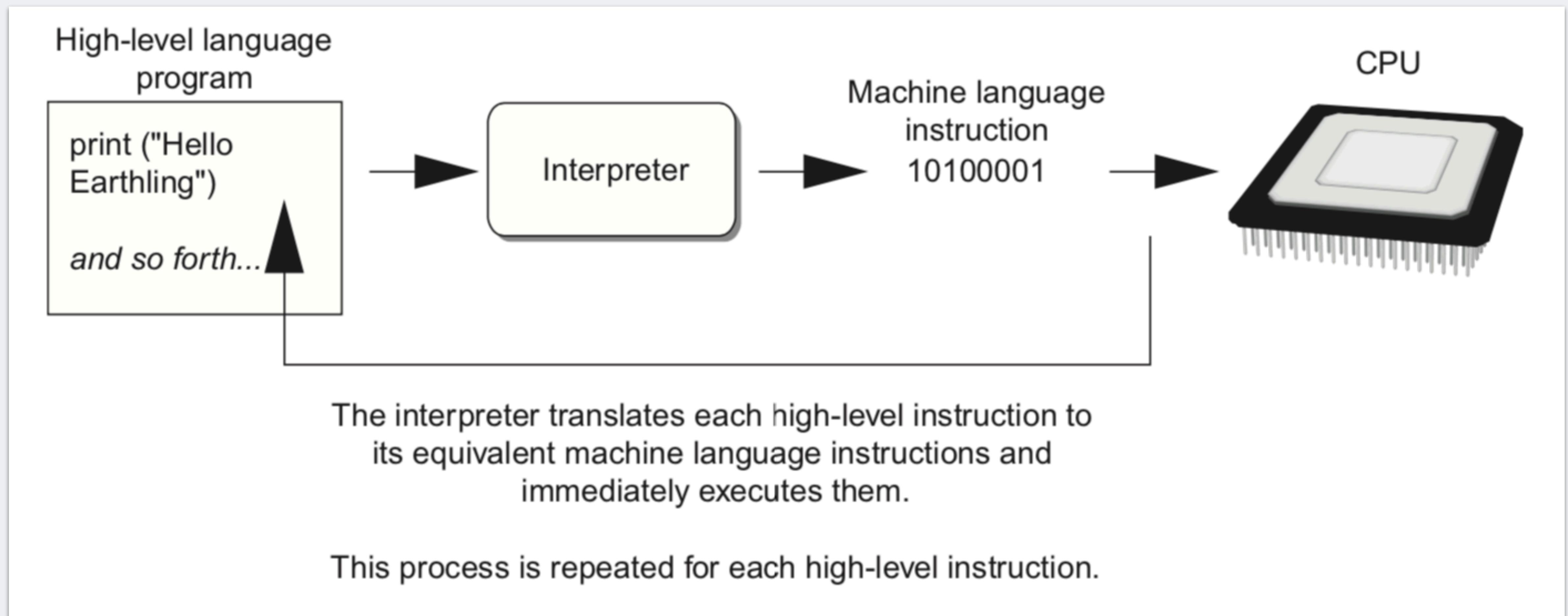
- การสื่อสาร: ช่วยให้โปรแกรมเมอร์สามารถสื่อสารคำสั่งไปยังคอมพิวเตอร์ได้ในรูปแบบที่เป็นระบบและเข้าใจง่าย
- การทำงานร่วมกัน: โปรแกรมเมอร์หลายคนสามารถทำงานร่วมกันในโปรเจกต์เดียวกันได้โดยการอ่านและแก้ไขชอร์สโค้ด
- การตรวจหาข้อผิดพลาด: เมื่อตรวจพบข้อผิดพลาด โปรแกรมเมอร์สามารถอ่านชอร์สโค้ดเพื่อหาสาเหตุและแก้ไขได้
- การบำรุงรักษา: ชอร์สโค้ดสามารถปรับปรุงเพิ่มเติมเพื่อเพิ่มคุณสมบัติใหม่ หรือเพิ่มประสิทธิภาพในการทำงานได้ในอนาคต

กล่าวโดยสรุป ชอร์สโค้ดคือแบบแปลน (blueprint) ของซอฟต์แวร์ทุกประเภท เป็นพื้นที่ที่โปรแกรมเมอร์นำร่อง ความคิดสร้างสรรค์ และทักษะในการแก้ปัญหามาใช้เพื่อสร้างโปรแกรมที่ทำงานได้จริงและมีประสิทธิภาพ

COMPILER

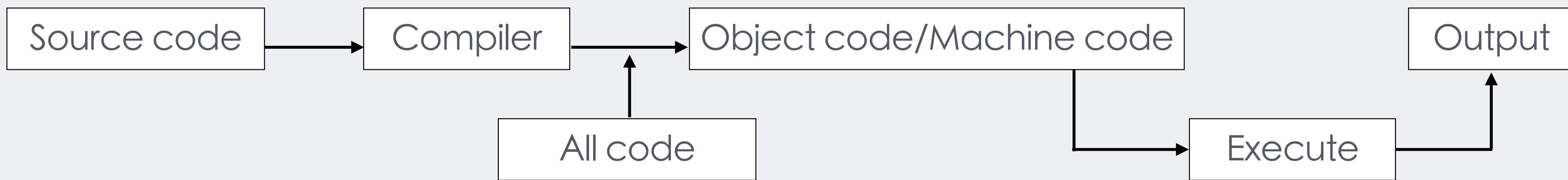


INTERPRETER

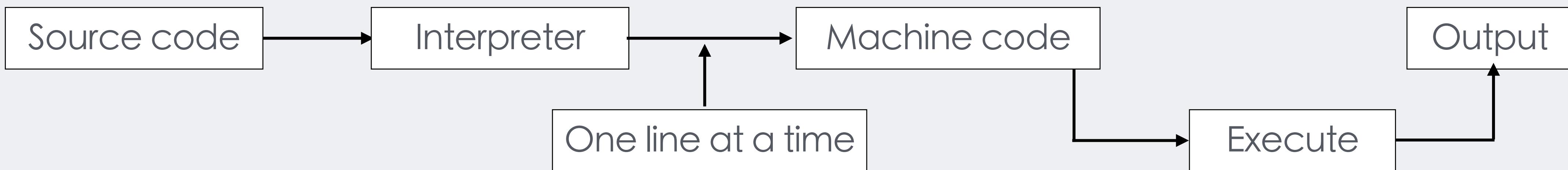


COMPILER VS INTERPRETER

Compiler



Interpreter



COMPILER VS INTERPRETER

ความแตกต่างที่สำคัญ:

- **ความเร็ว:** โปรแกรมที่ถูกคอมไพล์มักจะทำงานได้เร็วกว่า เพราะซอฟต์แวร์สโตร์โค้ดทั้งหมดถูกแปลงเป็นรหัสเครื่องล่วงหน้า ในขณะที่โปรแกรมที่แปลผลจะทำงานช้ากว่า เพราะแปลงโค้ดในขณะรัน
- **การตรวจจับข้อผิดพลาด:** คอมไพล์เตอร์สามารถตรวจพบข้อผิดพลาดได้ก่อนโปรแกรมถูกรัน ขณะที่ตัวแปลจัดตรวจพบข้อผิดพลาดเมื่อโปรแกรมทำงาน
- **ความยืดหยุ่น:** ตัวแปลให้ความยืดหยุ่นในการทดสอบและดีบกมากกว่า เพราะสามารถโค้ดได้ทันทีโดยไม่ต้องคอมไابل์ใหม่

COMPILER VS INTERPRETER

เหตุผลที่กระบวนการเหล่านี้มีความสำคัญ:

- **ประสิทธิภาพ:** การเข้าใจกระบวนการเหล่านี้ช่วยให้เลือกภาษาการเขียนโปรแกรมและเครื่องมือที่เหมาะสมกับงาน
- **การจัดการข้อผิดพลาด:** การทราบว่าภาษาใดเป็นภาษาแบบคอมไพล์หรือแปลผลช่วยให้คาดการณ์ได้ว่าเมื่อได้ข้อผิดพลาดจะถูกตรวจพบ
- **สมรรถนะของโปรแกรม:** ภาษาที่ถูกคอมไпал์มักเหมาะสมสำหรับโปรแกรมที่ต้องการประสิทธิภาพสูง ส่วนภาษาที่แปลผลจะเหมาะสมกับการพัฒนาอย่างรวดเร็วและการทดลองแนวคิดใหม่ ๆ

โดยสรุป การคอมไпал์และการแปลผลเป็นกระบวนการสำคัญในการแปลงชอร์สโค้ดที่มนุษย์เขียนให้อยู่ในรูปแบบที่คอมพิวเตอร์สามารถดำเนินการได้ โดยแต่ละกระบวนการมีจุดเด่นและข้อดีเฉพาะที่เหมาะสมกับบริบทที่แตกต่างกัน

PYTHON



PYTHON HISTORY

HISTORY OF PYTHON

- Python was conceptualized by **Guido Van Rossum** in the late **1980s**.
- Rossum published the first version of Python code (0.9.0) in February **1991** at the CWI (Centrum Wiskunde & Informatica) in the Netherlands , Amsterdam.
- Python is derived from **ABC** programming language, which is a general-purpose programming language that had been developed at the CWI.
- Rossum chose the name "**Python**", since he was a big fan of Monty Python's Flying Circus.
- Python is now maintained by a core development team at the institute, although Rossum still holds a vital role in directing its progress.



WHY PYTHON

Jun 2025	Jun 2024	Change	Programming Language	Ratings	Change
1	1		 Python	25.87%	+10.48%
2	2		 C++	10.68%	+0.65%
3	3		 C	9.47%	+0.24%
4	4		 Java	8.84%	+0.44%
5	5		 C#	4.69%	-1.96%
6	6		 JavaScript	3.21%	-0.11%
7	7		 Go	2.28%	+0.35%
8	9		 Visual Basic	2.20%	+0.54%
9	11		 Delphi/Object Pascal	2.15%	+0.62%
10	10		 Fortran	1.86%	+0.33%

WHY PYTHON

Java

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```

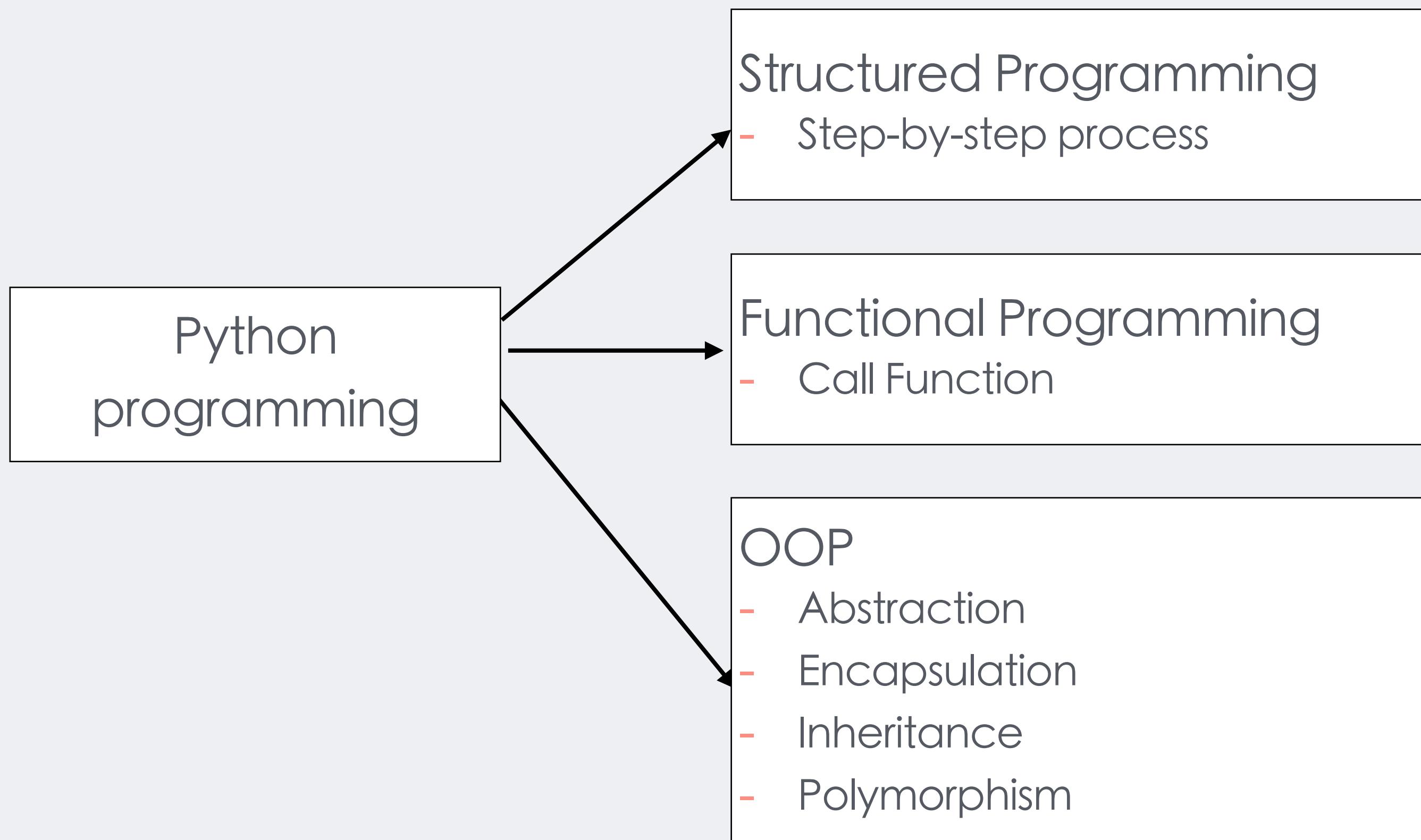
C++

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello World!";
    return 0;
}
```

Python

```
print('Hello World!')
```

MULTI-PARADIGM PROGRAMMING



PYTHON LANGUAGE FEATURES

คุณลักษณะเด่นของไพธอน:

- ความสามารถในการอ่านง่าย (Readability): โครงสร้างคำสั่งที่อ่านง่าย
- ไลบรารีที่ครอบคลุม (Extensive Libraries): มีไลบรารีและเฟรมเวิร์กมาย
- ความยืดหยุ่น (Versatility): ใช้ได้ในหลายด้าน เช่น เว็บ, ข้อมูล, AI, งานอัตโนมัติ
- การสนับสนุนจากชุมชน (Community Support): มีชุมชนผู้ใช้งานใหญ่และช่วยเหลือกันดี

PYTHON APPLICATIONS

- การพัฒนาเว็บ: Django, Flask
- วิทยาการข้อมูล: Pandas, NumPy, Matplotlib
- การเรียนรู้ของเครื่อง: TensorFlow, PyTorch
- งานอัตโนมัติ: สคริปต์สำหรับการทำงานอัตโนมัติ

CODING ENVIRONMENT



PYTHON INSTALL

The screenshot shows the Python.org homepage with a dark blue header featuring the Python logo and the word "python™". A navigation bar below includes links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The main content area has a yellow background. It features a large yellow button labeled "Download Python 3.12.4". Below it, text says "Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)". Another section mentions "Want to help test development versions of Python 3.13? [Prereleases](#), [Docker images](#)". To the right, there's a graphic of two packages descending from the sky on parachutes. At the bottom, a white footer bar contains the Python logo and the text "python.org".

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

Download the latest version for macOS

Download Python 3.12.4

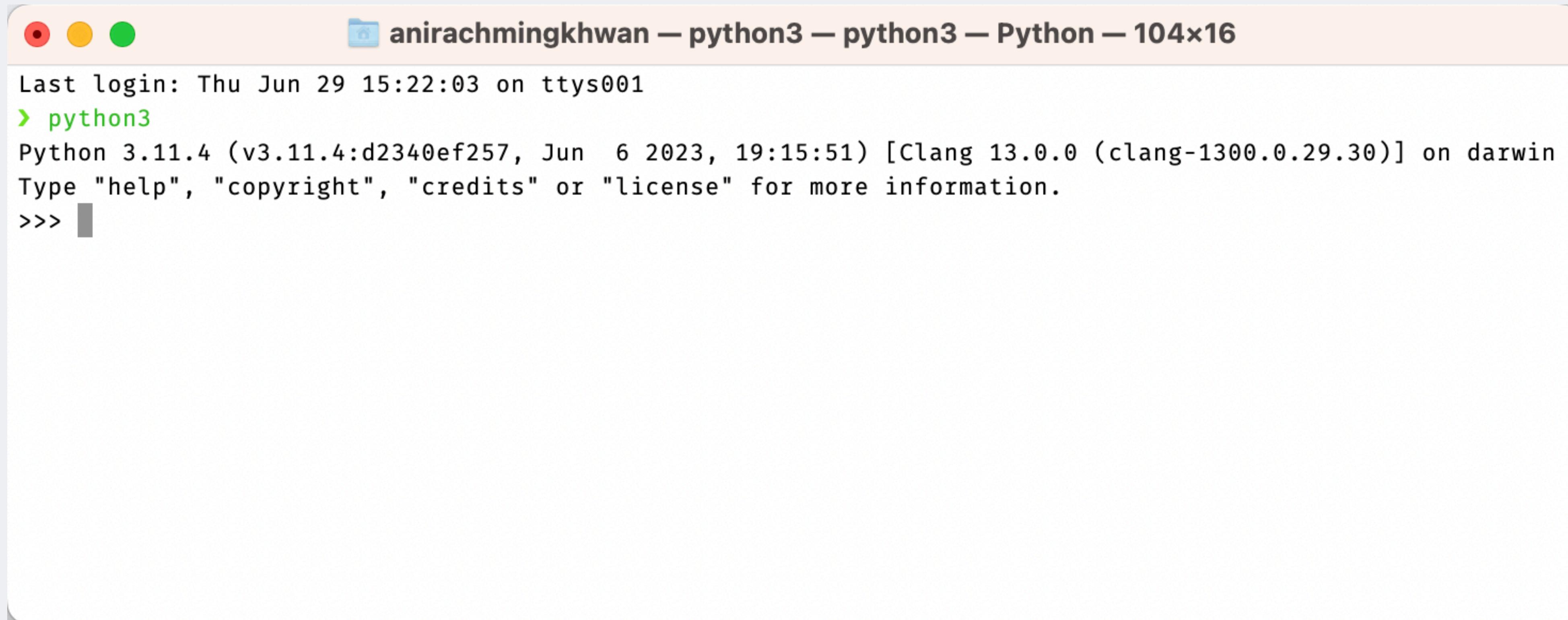
Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)

Want to help test development versions of Python 3.13? [Prereleases](#), [Docker images](#)

python.org

www.python.org

PYTHON COMMAND PROMPT



The screenshot shows a macOS terminal window titled "anirachmingkhwan — python3 — python3 — Python — 104x16". The window contains the following text:

```
Last login: Thu Jun 29 15:22:03 on ttys001
> python3
Python 3.11.4 (v3.11.4:d2340ef257, Jun 6 2023, 19:15:51) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

FIRST PROGRAM

Traditionally, the first program you write in a new language is called “Hello, World!” because all it does is display the words “Hello, World!”. In Python, it looks like this:

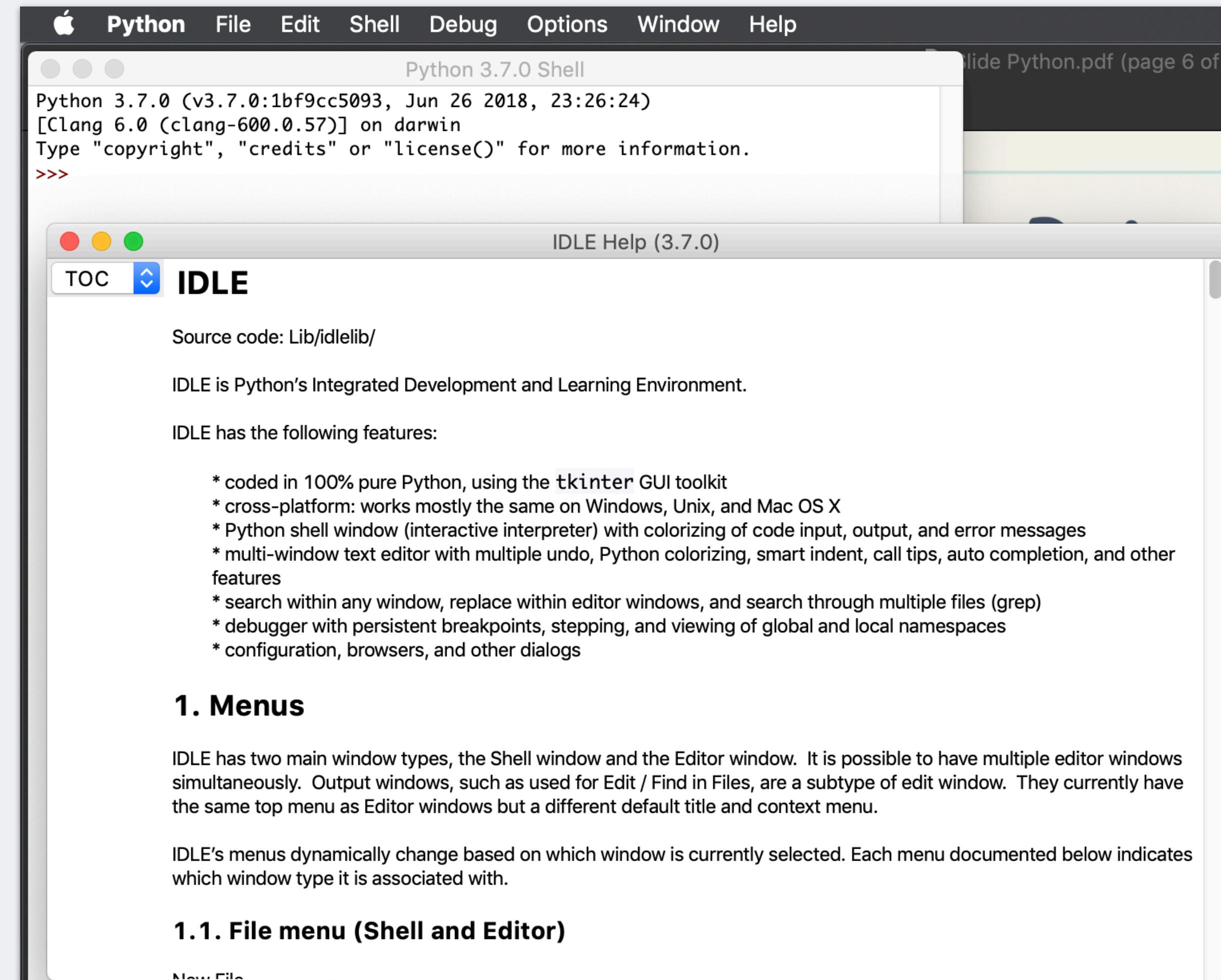
```
>>> print('Hello, World!')
```

This is an example of a print statement, although it doesn't actually print anything on paper. It displays a result on the screen. In this case, the result is the words

Hello, World!

The quotation marks in the program mark the beginning and end of the text to be displayed; they don't appear in the result. The parentheses indicate that print is a function.

IDLE



JUPYTER NOTEBOOK

Installing Jupyter with pip

As an existing or experienced Python user, you may wish to install Jupyter using Python's package manager, pip, instead of Anaconda.

If you have Python 3 installed (which is recommended):

```
python3 -m pip install --upgrade pip  
python3 -m pip install jupyter
```

If you have Python 2 installed:

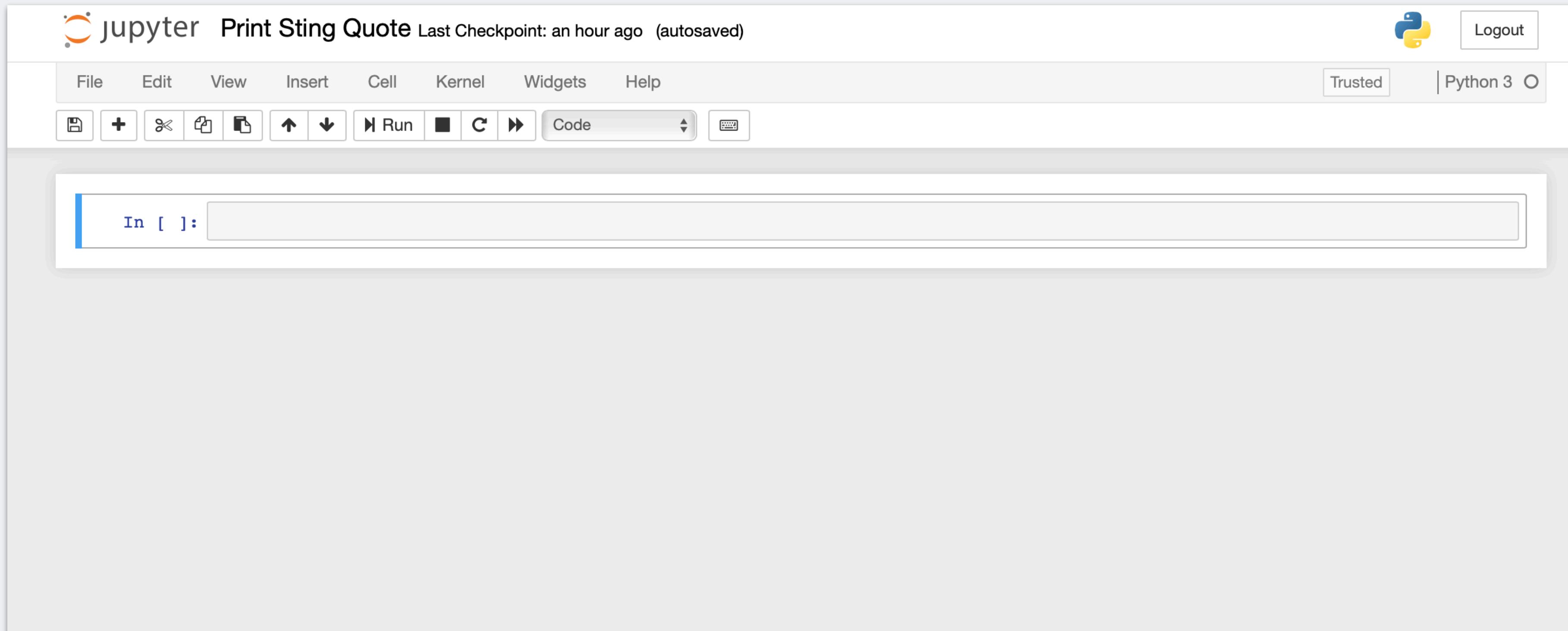
```
python -m pip install --upgrade pip  
python -m pip install jupyter
```

Congratulations, you have installed Jupyter Notebook! To run the notebook, run the following command at the Terminal (Mac/Linux) or Command Prompt (Windows):

```
jupyter notebook
```

See [Running the Notebook](#) for more details.

JUPYTER NOTEBOOK



VISUAL STUDIO CODE



VISUAL STUDIO CODE

The screenshot displays the official Visual Studio Code website and a window of the VS Code application.

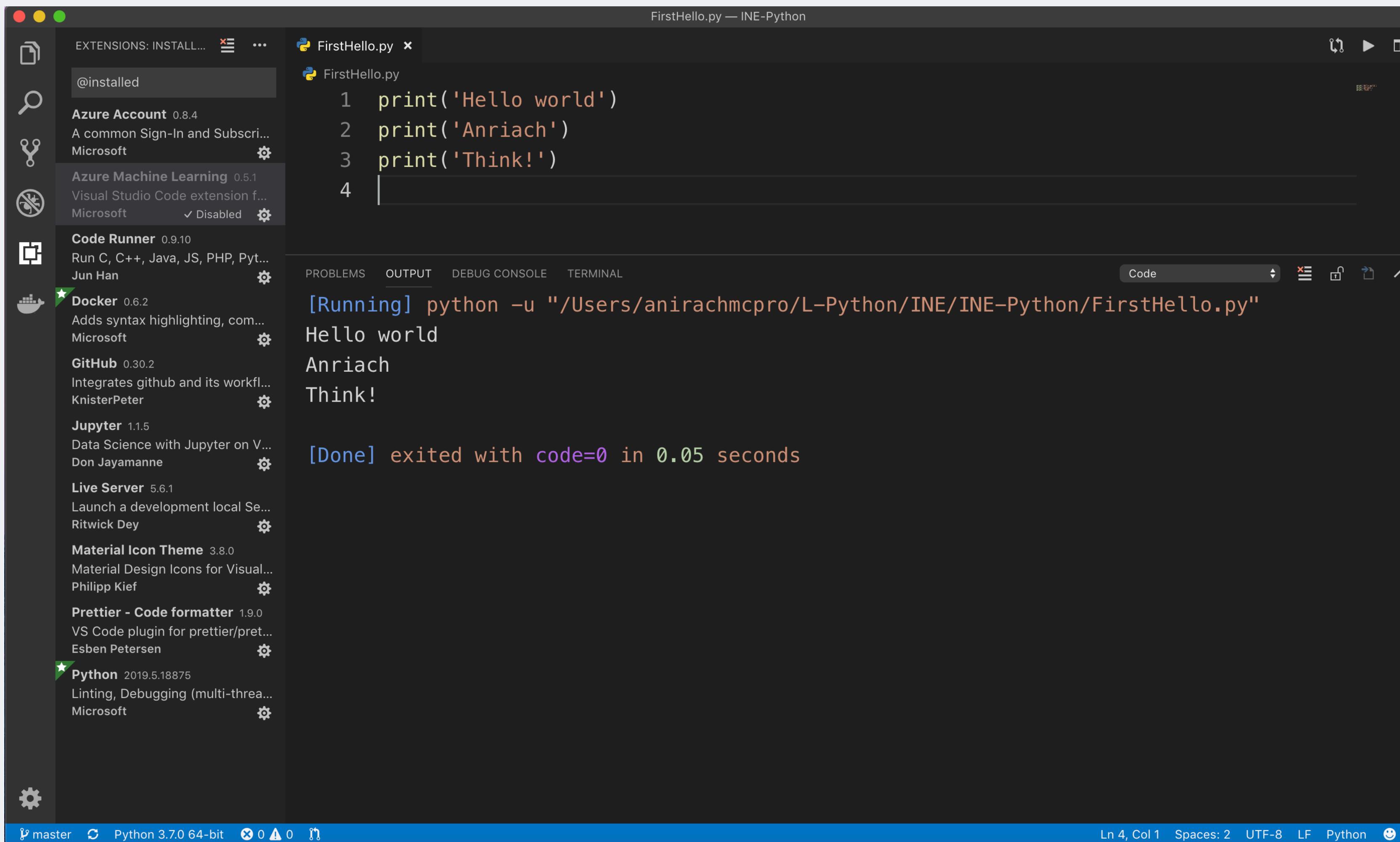
Visual Studio Code Website:

- Header:** Visual Studio Code, Docs, Updates, Blog, API, Extensions, FAQ, Learn, Search Docs, Download.
- Middle Section:** A large white box contains the text "Code editing. Redefined." and "Free. Built on open source. Runs everywhere." Below this are download buttons for "Download Mac Universal Stable Build" and "Web, Insiders edition, or other platforms".
- Bottom Section:** A note about terms of service: "By using VS Code, you agree to its license and privacy statement."

VS Code Application Window:

- Left Sidebar:** EXTENSIONS: MARKETPLACE showing a list of installed extensions: Python, GitLens, C/C++, ESLint, Debugger for C#, Language Support, vscode-icons, and Vetur.
- Editor Area:** Three tabs are visible: blog-post.js (active), index.js, and utils.js. The code editor shows a snippet of JavaScript related to a Gatsby GraphQL application.
- Bottom Status Bar:** Shows the file path "blog-post.js — gatsby-graphql-app", the status "2: Task - develop", and other system information like "Ln 6, Col 21, Spaces: 2, UTF-8, LF, JavaScript".

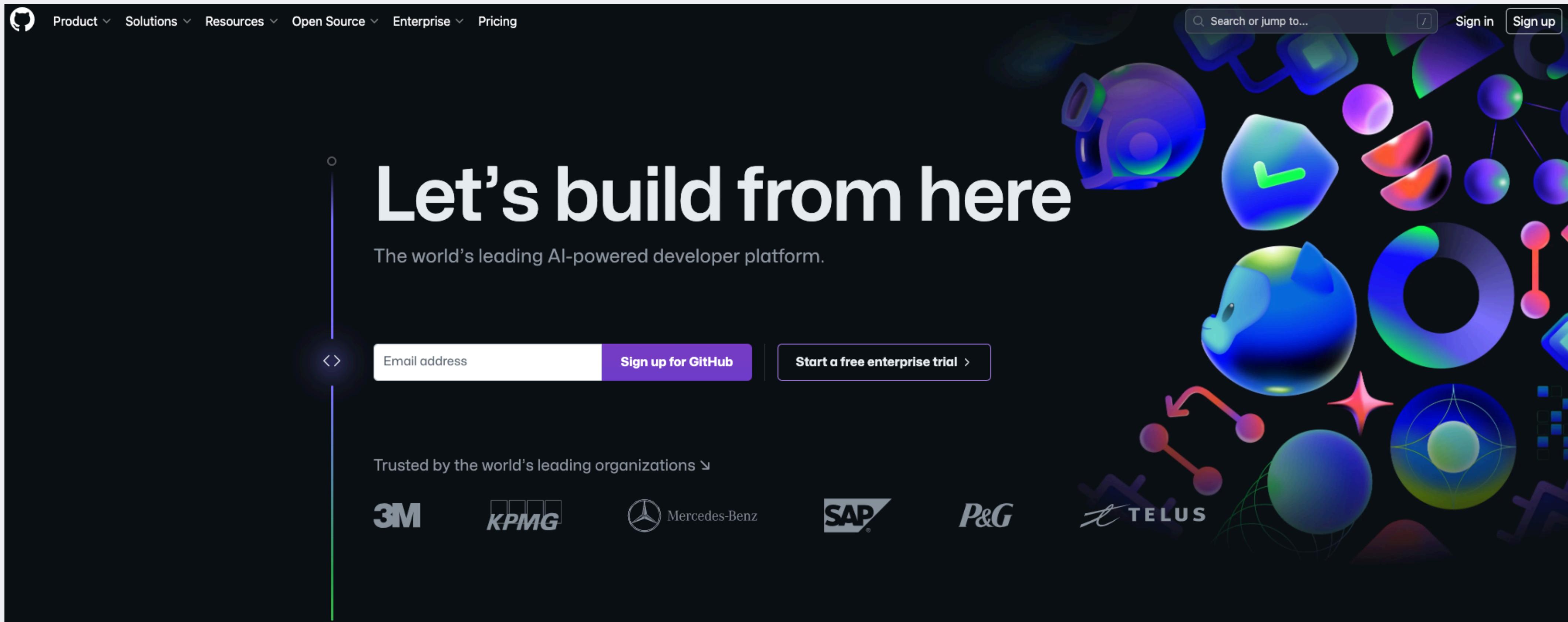
VISUAL STUDIO CODE



GITHUB



GITHUB



The screenshot shows the GitHub homepage with a dark background featuring a colorful, abstract graphic of interconnected nodes and shapes. At the top, there is a navigation bar with links for Product, Solutions, Resources, Open Source, Enterprise, and Pricing, along with a search bar and sign-in options. The main headline reads "Let's build from here" with the subtitle "The world's leading AI-powered developer platform." Below this, there is a sign-up form with fields for "Email address" and "Sign up for GitHub", and a link to "Start a free enterprise trial". A vertical sidebar on the left contains icons for code, a person, and a gear, followed by the text "Trusted by the world's leading organizations" and logos for 3M, KPMG, Mercedes-Benz, SAP, P&G, and TELUS.

Product Solutions Resources Open Source Enterprise Pricing

Search or jump to... Sign in Sign up

Let's build from here

The world's leading AI-powered developer platform.

Email address [Sign up for GitHub](#) [Start a free enterprise trial >](#)

Trusted by the world's leading organizations

3M KPMG Mercedes-Benz SAP P&G TELUS

WHY GITHUB

- A Nice Profile
- It makes it easy to contribute to your open source projects
- Documentation
- Showcase your work
- Markdown
- GitHub is a repository
- Track changes in your code across versions
- Integration options with common platform

GITHUB STEP

- Sign In GitHub
- GitHub Extension
- Create Repository
- Clone GitHub Repository from VSCode
- Commit
- Push

PYTHON PRINT COMMAND



`print()`

CO-LAB

The screenshot shows the CO-LAB interface. At the top, there's a navigation bar with a 'CO' logo, the title 'Untitled1.ipynb' with a star icon, and a file menu ('File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', 'Help'). To the right of the title are 'Comment', 'Share', 'Settings', and a user profile picture. Below the navigation bar is a toolbar with icons for '+ Code' and '+ Text'. A main workspace area contains a button with a play icon and the text 'Start coding or generate with AI.' On the left side, there's a sidebar with icons for a list, search, code block, key, and folder.

PYTHON PRINT COMMAND

- Print Command
 - 1. Print String
 - 2. Print Numeric
 - 3. Print String + Numeric
 - 4. Print String + String
 - 5. Print Numeric + Numeric

PYTHON PRINT COMMAND



Print String

PYTHON PRINT COMMAND

- Print String

```
In [1]: print('Hello World!')
```

```
Hello World!
```

```
In [2]: print('Information and Network Engineering')
```

```
Information and Network Engineering
```

```
In [3]: print('Hello Anirach you will be great')
```

```
Hello Anirach you will be great
```

PYTHON PRINT COMMAND

- Print String

```
In [1]: print("Hello World")
```

```
Hello World
```

```
In [2]: print("Information and Network Engineering")
```

```
Information and Network Engineering
```

```
In [3]: print("Hello Anirach you will be success")
```

```
Hello Anirach you will be success
```

PYTHON PRINT COMMAND

- Print String

```
In [4]: print("I'm hungry")
```

```
I'm hungry
```

```
In [5]: print('He ask me "Who got the book?"')
```

```
He ask me "Who got the book?"
```

PYTHON PRINT COMMAND

• Print String

```
: print("I'm good and \"I will do my best\"")
```

I'm good and "I will do my best"

```
print('\\\\')
```

八

```
print('\ta\na\ta\ta\n\t')
```

a
a a
a

PYTHON PRINT COMMAND

- Print String

```
print('''\ta  
a\ta\ta  
\ta''' )
```

a
a a
a

If you want to print multiple line the use '''...'''

PYTHON PRINT COMMAND



Print Numeric

PYTHON PRINT COMMAND

- Print Numeric

```
print('100.00')
```

```
100.00
```

```
print('%d' %100)
```

```
100
```

```
print('%d' %100.58)
```

```
100
```

```
print('%f' %-100.89)
```

```
-100.890000
```

```
print('.2f' %100.89)
```

```
100.89
```

PYTHON PRINT COMMAND

- Print Numeric

```
from math import pi  
print('%.2f' %pi)
```

3.14

```
print('%.4f' %pi)
```

3.1416

```
print('%.50f' %pi)
```

3.14159265358979311599796346854418516159057617187500

PYTHON PRINT COMMAND



Print String + Numeric

PYTHON PRINT COMMAND

- Print String + Numeric

```
print('My age is', 25,'I have', 3500.50,'Bth.')
```

My age is 25 I have 3500.5 Bth.

```
print('My age is %d I have %.2f Bth.' %(25,3500.50))
```

My age is 25 I have 3500.50 Bth.

```
print('My age is '+ str(25) +' I have '+ str(3500.50) +' Bth.')
```

My age is 25 I have 3500.5 Bth.

PYTHON PRINT COMMAND

- Print String + Numeric

```
print('5+4 = ', 5+4)
```

5+4 = 9

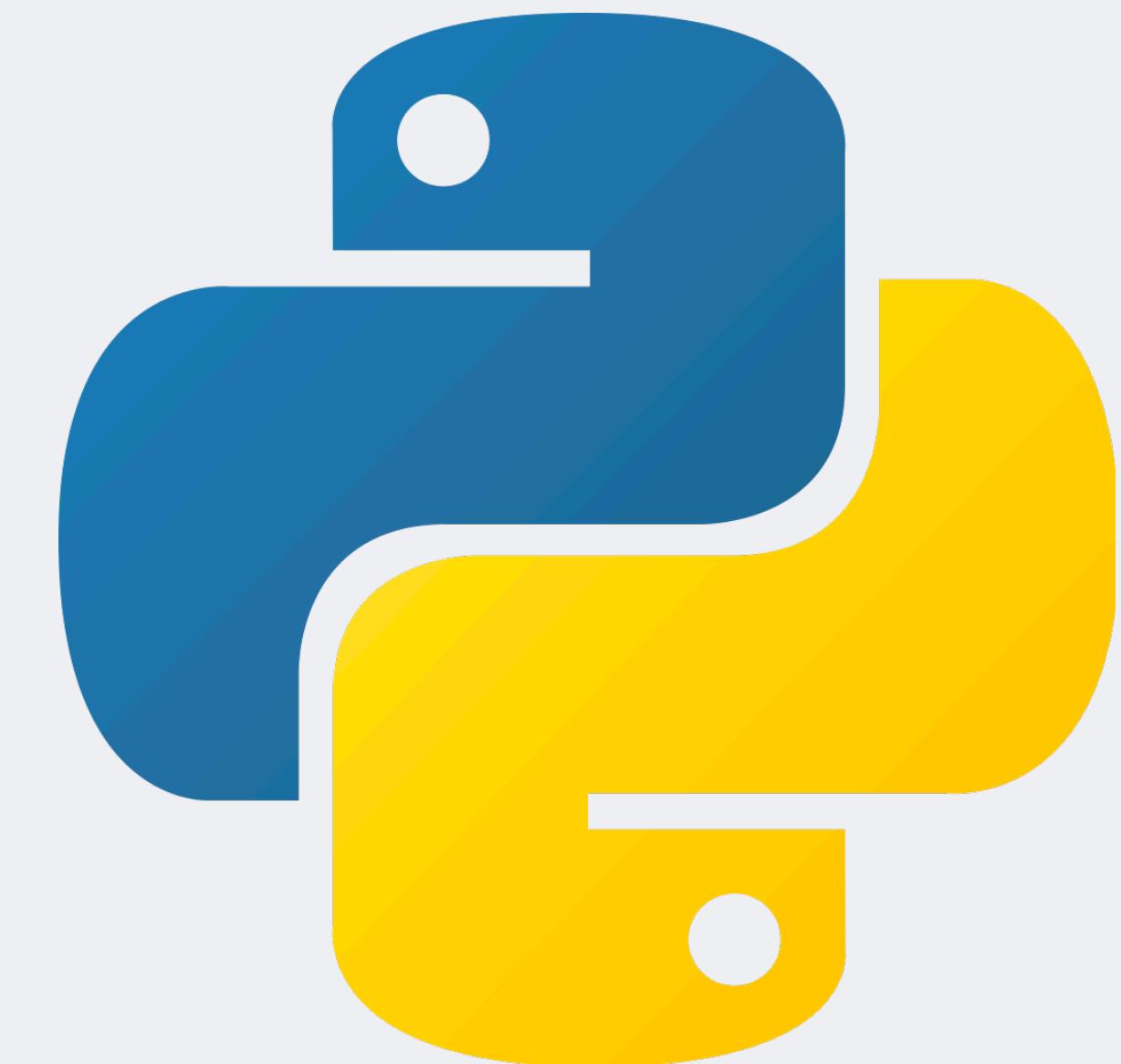
```
print('5+4 = %.2f' % (5+4))
```

5+4 = 9.00

```
print('5+4 = ' +str(5+4))
```

5+4 = 9

PYTHON PRINT COMMAND



Print String + String

PYTHON PRINT COMMAND

- Print String + String

```
print("I'm Anirach", "I'll keep practicing!")
```

I'm Anirach I'll keep practicing!

```
print("I'm Anirach " + "I'll keep practicing!")
```

I'm Anirach I'll keep practicing!

PYTHON PRINT COMMAND



Print Numeric + Numeric

PYTHON PRINT COMMAND

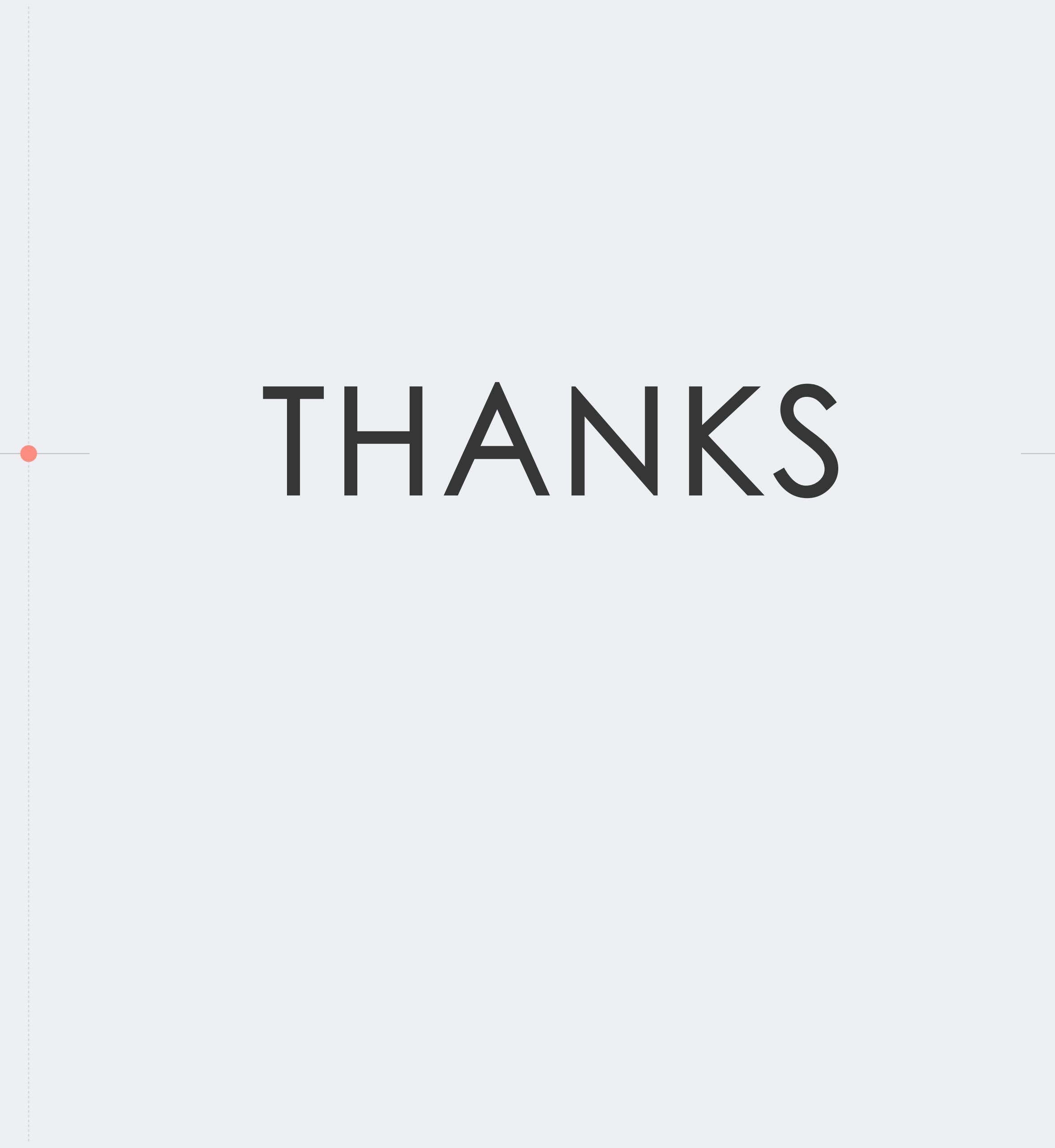
- Print Numeric + Numeric

```
print(1, 1, 2, 3, 5, 8, 13, 21, 34, 55)
```

```
1 1 2 3 5 8 13 21 34 55
```

```
print('%d %d %d %d %d %d %d %d' %(1,1,2,3,5,8,13,21,34,55))
```

```
1 1 2 3 5 8 13 21 34 55
```



THANKS

WORD OF THE WISE

