

ก่อร่าง Pygram

แผนการพัฒนาและทดสอบ (Development, Description & Discernment)

การเขียนและปรับปรุง Prompt (Iterative Prompting) เพื่อให้ได้ผลลัพธ์ที่ดีที่สุด

Prompt Log

คุณคือ Senior Frontend Engineer ระดับมืออาชีพ มีความเชี่ยวชาญด้าน Web Application Architecture, In-Browser Runtime (Pyodide), UI/UX Design และ AI Integration ให้สร้าง Web Application แบบ Single-File HTML ชื่อ: "Pygram IDE"

ข้อกำหนดสำคัญ:

- ต้องอยู่ในไฟล์ HTML เพียงไฟล์เดียวเท่านั้น
- ใช้ `<style>` และ `<script>` ภายในไฟล์เท่านั้น
- ห้ามใช้ CSS Framework ภายนอก เช่น Bootstrap, Tailwind
- ใช้ System Fonts เท่านั้น
- ใช้ Pyodide ผ่าน CDN สำหรับวัน Python ใน Browser
- ใช้ ES Module import สำหรับ @google/generative-ai ผ่าน esm.run
- ห้ามใช้ build tools หรือ bundler ใด ๆ
- ต้องเป็นโค้ดที่พร้อมเปิดใน Browser ได้ทันที

1) สถาปัตยกรรม (Architecture)

- เขียนโค้ดแบบ Modular ภายในไฟล์เดียว
- แยกโครงสร้างเป็นฟังก์ชันตามหน้าที่
- หลีกเลี่ยงการใช้ Global variable ที่ไม่จำเป็น
- ใช้ IIFE หรือ scoped module pattern หากเหมาะสม
- โค้ดต้องอ่านง่าย มีโครงสร้างชัดเจน

2) Theme & UI Design

ต้องเป็น Dark Theme คล้าย VS Code โดยกำหนด CSS Variablesอยู่ในรีมังค์นี้

```
--bg:#0d0f14  
--panel:#181c26  
--border:#252a38  
--acc:#00d4aa  
--text:#cdd6f4
```

ข้อกำหนดเพิ่มเติม:

- Editor ต้องใช้ Monospace
- Layout ใช้ Flexbox
- Sidebar กว้าง 44px
- File Explorer พับ/ขยายได้
- AI Panel อยู่ด้านขวา กว้าง 320px และพับได้
- Responsive: ถ้าหน้าจอเล็ก ให้ซ่อน AI Panel อัตโนมัติ
- Scrollbar ต้องถูกตัดแต่งให้เข้ากับธีม

3) โครงสร้าง Layout หลัก

ต้องประกอบด้วย:

3.1) Top Title Bar

- โลโก้:  PygramIDE
- แสดงชื่อไฟล์ปัจจุบัน
- ปุ่มสลับภาษา TH / EN

3.2) Left Vertical Sidebar

- ช่อง Explorer , Run, Clear, Copy, Load Sample, Settings, AI Toggle

3.2)File Explorer Panel

- แสดงรายการไฟล์

- ขวาจับ Context Menu (Right Click)

3.3)Main Editor Area

- Tab bar

- Code Editor

- Output Panel

3.4)Output Panel

- Console Tab

- Graph Tab (สำหรับ matplotlib)

3.5)AI Assistant Panel

- พื้น/ขยายได้

4) Code Editor (ห้ามใช้ Monaco หรือ Library ภายนอก)

- ใช้ textarea เป็นฐาน

- มี Line Numbers

- มี Current Line Highlight

- Syntax Highlight สำหรับ Python:

- keywords

- builtins

- strings

- comments

- numbers

- decorators

- operators
- function calls
- ใช้ span wrapping สำหรับ highlight
 - อัปเดต highlight เมื่อมีการพิมพ์
- Sync scroll ระหว่าง textarea และ background layer
- รองรับ Tab = 4 spaces
- รองรับ Ctrl+Enter เพื่อ Run

5) Output System

Console:

- แสดง stdout
- แสดง error เป็นสีแดง
- แสดง execution time
- มี Clear button

Graph:

- ถ้าโค้ดใช้ matplotlib
- แสดงภาพใน Graph tab
- สามารถ Save ขึ้นได้

Status Bar:

- Ln, Col
- จำนวนบรรทัด
- Python version
- AI status

6) Python Execution (ใช้ Pyodide)

- โหลด Pyodide จาก CDN
- Override sys.stdout เพื่อจับ print()
- Override sys.stderr
- รองรับ matplotlib ด้วย Agg backend
- ถ้ามี error ให้แสดง traceback
- วัดเวลา execution
- แสดงผลลัพธ์อย่างเดียว (ไม่ขอขยาย)

7) Virtual File System

- ใช้ Array เก็บไฟล์ในหน่วยความจำ
- รองรับ:
 - Create File
 - Rename
 - Duplicate
 - Delete
- มี Tabs system
- มี Context Menu (Right Click)
- ต้องมีไฟล์เริ่มต้น main.py

8) AI Assistant Panel

มี 5 โหมด: - Explain – Debug – Theory – Algorithm – Flowchart

แต่ละโหมดต้องมี:- ช่อง Input - ปุ่ม Pull from Editor - ปุ่ม Execute – ช่องแสดงผล
ผลลัพธ์ต้อง Render Markdown

ใช้ Markdown Renderer แบบ minimal inline (ไม่ใช้ library)

9) Flowchart Mode

- วิเคราะห์โค้ด Python
- สร้างโครงสร้าง Flowchart แบบ JSON
- Render block diagram แบบง่าย
- มีปุ่ม Zoom In / Zoom Out
- มีปุ่ม Export

10) Internationalization (i18n)

รองรับ: - ภาษาไทย - ภาษาอังกฤษ

- เก็บค่าภาษาใน localStorage
- ใช้ dictionary object แปล:

 - Buttons
 - Labels
 - Placeholder
 - Status messages

11) Settings Modal

- Modal overlay สีเข้ม
- ช่องกรอก API Key
- บันทึกใน localStorage
- ปุ่ม Save / Cancel

12) UX Enhancements

- Toast notification system

- Smooth transition animation
- Styled scrollbar
- ផ្សែន hover effect
- Keyboard shortcuts

13) Default Sample Code

នឹងបានការងារ តូចមីគឺដែល:

- ផែកច័ន្ទ greet()
- ឬទេស list comprehension
- Fibonacci function
- លាយ print statements

14) ខ្លួនរបាយការណ៍ពិសេស

- គឺគឺតុកដាក់ចាត់ចាយ
- តុកដាក់ចាត់ចាយបានចិត្តទៅជាមីនុយុទ្ធផល
- នឹងមីនុយុទ្ធផលរបាយនៃការងារ (យកពី Pyodide CDN ឬ generative-ai esm)
- តុកដាក់ចាត់ចាយបានចិត្តទៅជាមីនុយុទ្ធផល
- ផែកច័ន្ទអិបាយទិន្នន័យ
- សំណើនាំការងារជាពេលវេលា

កំណត់គឺគឺតុកដាក់ចាត់ចាយ:

ផ្តល់នូវការងារក្នុងការងាររបាយការណ៍ពិសេស

ដោយផ្តល់នូវការងារក្នុងការងាររបាយការណ៍ពិសេស

និងការងារក្នុងការងាររបាយការណ៍ពិសេស

Senior Frontend Engineer
ผู้เชี่ยวชาญด้านการพัฒนา Web Application Architecture, In-Driver Runtime (Pyodide), UI/UX Design และ AI Integration...

PASTED

Architected comprehensive IDE with multiple integrated components >

Pygram IDE v9 พร้อมใช้งานแล้ว — ไฟล์ HTML เต็มรูปแบบสมบูรณ์

ฟีเจอร์น่าสนใจ:

- 🐍 Python Runtime ผ่าน Pyodide CDN (รองรับ matplotlib)
- 🎨 Syntax Highlighting แบบ custom (ไม่ใช้ library ภายนอก) สำหรับ keywords, builtins, strings, comments, decorators, functions
- 📁 Virtual File System + Tabs + Context Menu (rename/duplicate/delete)
- 🤖 AI Assistant 5 ใหม่: Explain, Debug, Theory, Algorithm, Flowchart ผ่าน Gemini API
- 📊 Graph Tab และผล matplotlib + ปุ่ม Save
- 🔢 Line Numbers + current line highlight + sync scroll
- 💻 Shortcuts: **Ctrl+Enter** = Run, **Tab** = 4 spaces, **Ctrl+/** = Toggle comment
- 🌐 i18n TH/EN ใน localStorage
- ⚙️ Settings Modal สำหรับ Gemini API Key
- 🔔 Toast Notifications + Resize output panel

เปิดไฟล์ใน Browser ได้ทันที ไม่ต้องติดตั้งอะไรเพิ่ม



Pygram ide v9
Code - HTML

Download

ยังมีบางที่งก็ใช้ภาษาไทยได้

1. Python Runtime ผ่าน Pyodide CDN บุรัณรัตน์ไม่ได้
- 2.i18n TH/EN ใน localStorage

ปรับส่วนที่ key ของไฟล์ใน google app script เพื่อป้องกัน API key ให้มีความปลอดภัย

วางแผนแก้ไขไฟล์ด้วยความปลอดภัย API >

Pygram IDE v9 อัปเดตแล้ว — แก้ปัญหาทั้งหมด:

↳ สิ่งที่แก้ไข

1. Pyodide Runtime (รันได้จริงแล้ว)

- เปลี่ยนจาก `import js` callback → ไฟ `StringIO capture` ที่เชื่อถือได้ 100%
- โค้ด Python ถูกส่งผ่าน `pyodide.globals.set('_user_code', code)` ก่อนรัน
- `stdout` / `stderr` ถูก redirect ผ่าน `sys.stdout = StringIO()` → อ่านค่าหลังรันเสร็จ

2. i18n TH/EN (ทำงานได้แล้ว)

- แก้ duplicate ID ทั้งหมด
- `applyLang()` map ทุก element ด้วย ID ที่ขัดเจนไม่ซ้ำกัน
- ปุ่ม lang toggle แสดง `TH` (เมื่อภาษาปัจจุบันเป็น EN) และ `EN` (เมื่อเป็น TH)

3. 🚨 GAS Proxy Security (ใหม่)

- ไม่มี API Key ใน browser อีกต่อไป — key เก็บใน Google Script Properties
- Settings modal มีโค้ด Apps Script ให้ copy พร้อม step-by-step วิธีติดตั้ง
- IDE ส่ง prompt ไปที่ GAS URL → GAS เรียก Gemini → ส่งผลกลับมา
- วาง GAS Web App URL แค่ครั้งเดียว บันทึกใน localStorage

การทดสอบระบบ ตรวจสอบความถูกต้อง (Verification) และแก้ไขข้อผิดพลาด (Troubleshooting)

1. ระบบอัจฉริยะโดยใช้ Ai

The screenshot shows the Pyram IDE interface. On the left, the code editor displays a Python script named `main.py`. The code includes functions for greeting and calculating squares and sums. On the right, the **AI Assistant** panel is open, showing a sidebar with tabs like "อธิบาย", "แก้ไข", "ตรวจสอบ", "Algorithm", and "Flow". Below the sidebar, there's a text area with a green button labeled "ขอให้เลือกนี่". A detailed explanation of the code is provided in Thai, followed by a section titled "1. ฟังก์ชัน greet(name):" which contains the original code. The AI also lists several bullet points about the code's behavior and provides a note at the bottom about using list comprehension.

```
# Pyram IDE - น้า Ctrl+Enter เพื่อเรียก
# Claude จะวินาทีเดียวและผลลัพธ์ทันที

def greet(name):
    return f"สวัสดี, {name}! 🌟"

nums = [1, 2, 3, 4, 5]
sq = [n**2 for n in nums]

print(greet("มานะใน CS"))
print("Squares:", sq)
print("Sum:", sum(sq))

def fib(n):
    a,b=0,1
    res=[]
    for _ in range(n):
        res.append(a);a,b=a+b,a

print("Fibonacci:", fib(10))
```

2. ระบบแก้ปัญหาโดยใช้ Ai

This screenshot shows the Pyram IDE interface with a similar setup to the first one. The code editor has the same `main.py` script. However, the AI Assistant panel now displays an "Error Message" section with a red box highlighting the word "Traceback". It provides a detailed analysis of the error, mentioning a recent call last, the file path, and the specific line number where the error occurred (line 19). The AI also suggests a fix for the error, which is to import the 're' module. The overall layout remains consistent with the first screenshot, showing the AI's role in both verifying the correctness of the code and helping to resolve runtime errors.

```
# Pyram IDE - น้า Ctrl+Enter เพื่อเรียก
# Claude จะวินาทีเดียวและผลลัพธ์ทันที

def greet(name):
    return f"สวัสดี, {name}! 🌟"

nums = [1, 2, 3, 4, 5]
sq = [n**2 for n in nums]

print(greet("มานะใน CS"))
print("Squares:", sq)
print("Sum:", sum(sq))

def fib(n):
    a,b=0,1
    res=[]
    for _ in range(n):
        res.append(a);a,b=a+b,a

print("Fibonacci:", fib(10))
```

3.ระบบบททฤษฎีต่างๆเกี่ยวกับโครงสร้างข้อมูลแบบโครงสร้างตัวอย่าง

The screenshot shows the Pygram IDE interface. In the code editor, there is a Python script named `main.py` containing the following code:

```
def recursion(n):
    """การเรียกฟังก์นั้น recursion เพื่อทำการบวกตัวเอง"""
    if n == 0:
        return 0
    else:
        return n + recursion(n - 1)
print(recursion(5)) # แสดง 15 (0 + 1 + 2 + 3 + 4)
```

In the AI Assistant panel, the tab "Algorithm" is selected. A search bar at the top has "Recursion" entered. Below it, a green button says "ดูรายละเอียด" (View details). The main content area displays the following text:

คำอธิบาย:
“การเรียกฟังก์นั้น recursion” หรือ “การเรียกฟังก์นั้นโดยมีตัวเองเป็น参数”
หมายความว่าฟังก์นั้นเรียกตัวเองซ้ำๆ ต่อไปจนกว่าจะได้ผลลัพธ์ที่ต้องการ
นั่นคือ “การเรียกตัวเอง” หรือ “การเรียกฟังก์นั้นโดยมีตัวเองเป็น参数”
เช่นในกรณีของฟังก์นั้น recursion ที่เราเขียนมา

คำอธิบายภาษา Python:
เมื่อเราเรียกฟังก์นั้นแล้ว ฟังก์นั้นจะเรียกตัวเองซ้ำๆ จนกว่าจะได้ผลลัพธ์ที่ต้องการ

def recursion(n):
 """การเรียกฟังก์นั้น recursion เพื่อทำการบวกตัวเอง"""

 if n == 0:
 return 0

 else:
 return n + recursion(n - 1)

#แสดงผลลัพธ์ที่ต้องการ

print(recursion(5)) # แสดง 15 (0 + 1 + 2 + 3 + 4)

4.ระบบสรุปข้อมูลเฉพาะเจาะจงที่เกี่ยวกับโครงสร้าง

The screenshot shows the Pygram IDE interface. In the code editor, there is a Python script named `main.py` containing the same recursive function as before:

```
def recursion(n):
    """การเรียกฟังก์นั้น recursion เพื่อทำการบวกตัวเอง"""
    if n == 0:
        return 0
    else:
        return n + recursion(n - 1)
print(recursion(5)) # แสดง 15 (0 + 1 + 2 + 3 + 4)
```

In the AI Assistant panel, the tab "Data Structure / Algorithm" is selected. A search bar at the top has "stack" entered. Below it, a purple button says "ดูรายละเอียด" (View details). The main content area displays the following text:

หัวข้อ “Stack” ที่พบบ่อยที่สุดใน CS (ภาษาไทย)
Stack คือโครงสร้างข้อมูลที่มีตัวถัดไปเป็น Computer Science และภาษาคอมพิวเตอร์

1. แนวคิด (Concept)

- Stack คือที่สุด: Stack คือโครงสร้างข้อมูลที่มีการเพิ่มและลดจำนวนข้อมูลที่มีอยู่ในตัวถัดไป
- หลักการทำงาน (LIFO - Last In, First Out): ข้อมูลที่เพิ่มเข้าไปใน Stack ล่าสุด จะถูกนำออกจาก Stack ก่อนเสมอ (เมื่อมีการนำข้อมูลออกจากตัวถัดไป)
- ประโยชน์ที่สำคัญ:
 - * Push: การเพิ่มข้อมูลเข้าไปใน Stack (เมื่อมีการเพิ่มข้อมูลใหม่)
 - * Pop: การนำข้อมูลออกจาก Stack (เมื่อมีการลดจำนวนข้อมูล)
 - * Peek/Top: การดูข้อมูลที่อยู่บนยอดของ Stack แต่ไม่ได้移除
 - * IsEmpty: ตรวจสอบว่า Stack ยังคงว่างเปล่าหรือไม่
 - * Size: ตรวจสอบจำนวนข้อมูลที่อยู่ใน Stack

2. ใช้ Cas (กรณีการใช้งาน)

Stack ใช้ในหลายสถานที่ เช่น ในการจัดการหน้าจอ ฯลฯ

- การเรียกฟังก์นั้น (Function Call Stack): เมื่อเราเรียกฟังก์นั้นแล้ว ฟังก์นั้นจะเรียกฟังก์นั้นต่อไป จนกว่าจะได้ผลลัพธ์ที่ต้องการ

5. ระบบสร้าง Flow chart อย่างง่าย

The screenshot shows a Python development environment with the following components:

- EXPLORER**: Shows a project named "PROJECT" containing a file "main.py".
- main.py**: The code for the Bubble Sort algorithm.
- Console**: Displays the execution results:

 - Before: [64, 34, 25, 12, 22, 11, 98]
 - After: [11, 12, 22, 25, 34, 64, 98]
 - / เวลา: 18ms

- AI Assistant**: A panel showing the AI-generated flowchart for the algorithm.
- Flowchart**: The generated flowchart for the Bubble Sort algorithm, showing the flow from input to output.

```
# Bubble Sort
def bubble_sort(arr):
    n=len(arr)
    for i in range(n):
        for j in range(0,n-i-1):
            if arr[j]>arr[j+1]:
                arr[j],arr[j+1]=arr[j+1],arr[j]
    return arr

data=[64,34,25,12,22,11,98]
print("Before:",data)
print("After: ",bubble_sort(data))
```

The flowchart consists of the following steps:

```
graph TD
    Start([ ]) --> Input[/ Bubble Sort /]
    Input --> Process[ ]
    Process --> Output[/ Loop /]
    Output --> End([ ])
```

The flowchart is annotated with comments in Thai:

- Start: รับข้อมูล
- Input: จัดการข้อมูล
- Process: คำนวณ
- Output: ผลลัพธ์
- End: จบการทำงาน