



C

# PHYTHON

การออกแบบขั้นตอนการทำงาน  
และ  
การเขียนโปรแกรมเบื้องต้น



# หลักการเขียนโปรแกรม



# โปรแกรมภาษาไพทอน

ไพทอน (python) ถูกพัฒนาครั้งแรกเมื่อปี 1989 โดย กิโด ฟาน รอสซัม (Guido van Rossum) โดยโครงสร้างของภาษานั้นจะไม่มีการยึดติดกับแพลตฟอร์มที่ใช้ ทำให้ python เป็น Open Source เต็มรูปแบบที่ใช้พัฒนาได้อย่างอิสระ

โครงสร้างของภาษา  
ไพทอนเบื้องต้น

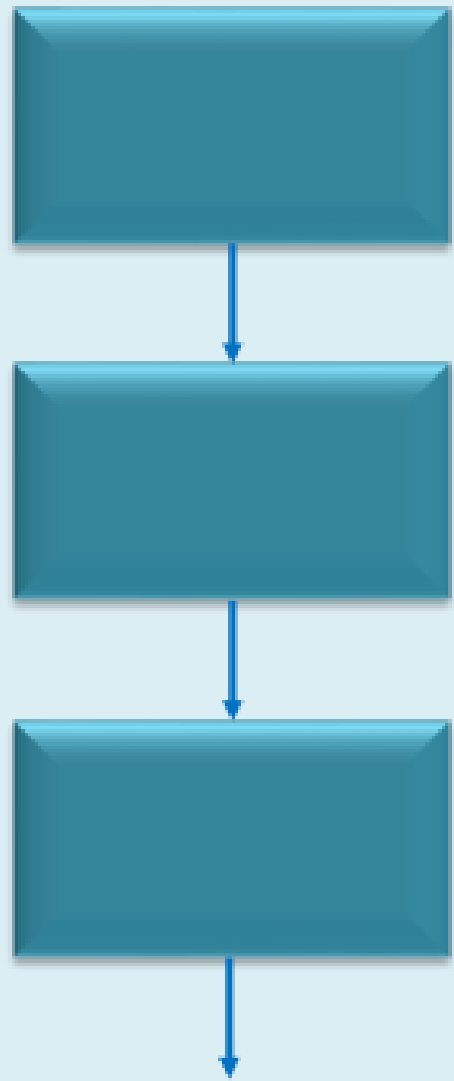
คอมเมนต์หรือการประกาศตัวแปรภาษา

การนำเข้าไลบรารี หรือคลาสของไพทอนมาใช้

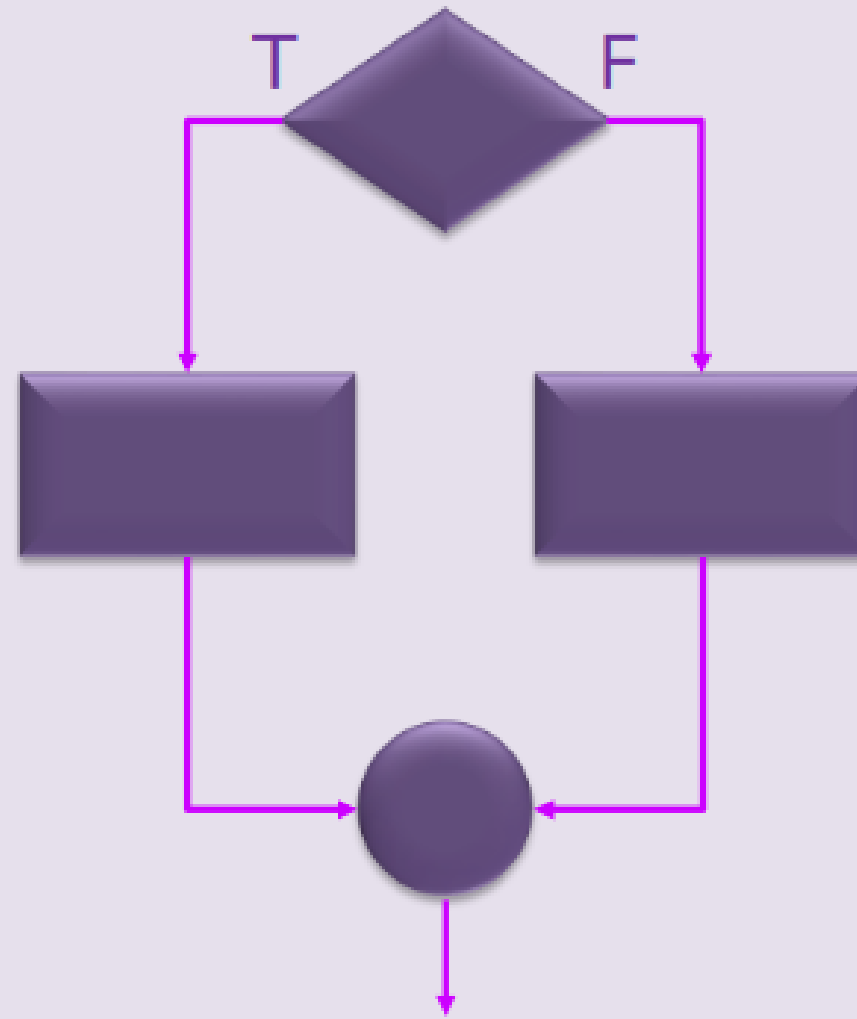
ประกาศตัวแปร ฟังก์ชัน และคำสั่งควบคุมการทำงาน



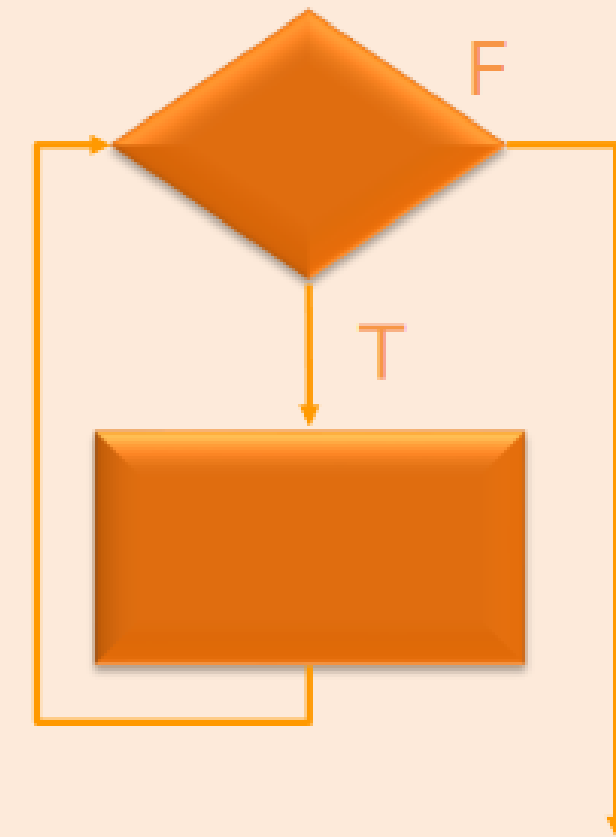
# รูปแบบการเขียนโปรแกรมคอมพิวเตอร์



โครงสร้างการทำงานแบบเรียงลำดับ  
(sequence structure)



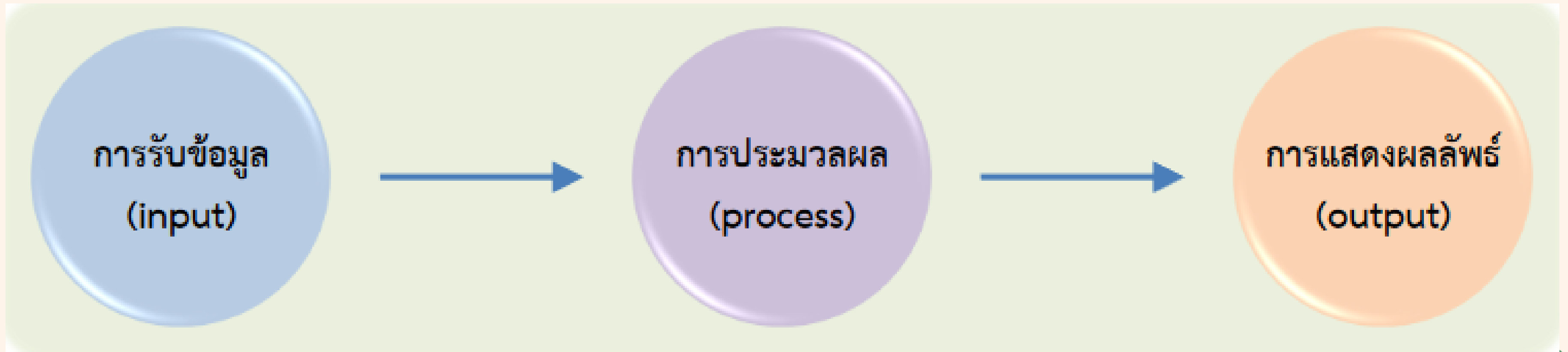
โครงสร้างการทำงานแบบมีเงื่อนไข  
(condition structure)



โครงสร้างการทำงานแบบทำซ้ำ  
(iteration structure)

# กระบวนการทำงานของโปรแกรม

กระบวนการทำงานของโปรแกรมคอมพิวเตอร์มี **3** กระบวนการ ได้แก่





# ภาษา Python

พัฒนาขึ้นครั้งแรกในปี 1989 โดย กิโด ฟาน รุสซิม ซึ่งเป็นภาษาโปรแกรมระดับสูงที่ใช้กันอย่างแพร่หลาย ถูกออกแบบเพื่อให้มีโครงสร้างและไวยากรณ์ของภาษาที่ไม่ซับซ้อน เข้าใจง่าย มีการใช้พัฒนาแอปพลิเคชัน เว็บไซต์ รวมถึงแอปบนมือถือหรือ อุปกรณ์เคลื่อนที่ด้วย หน้าที่ของ Python ก็คือการทำงานแปลชุดคำสั่งที่ละบรรทัดเพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ทำงานตามที่เราต้องการ



# เครื่องมือในการเขียนภาษาไพทอน

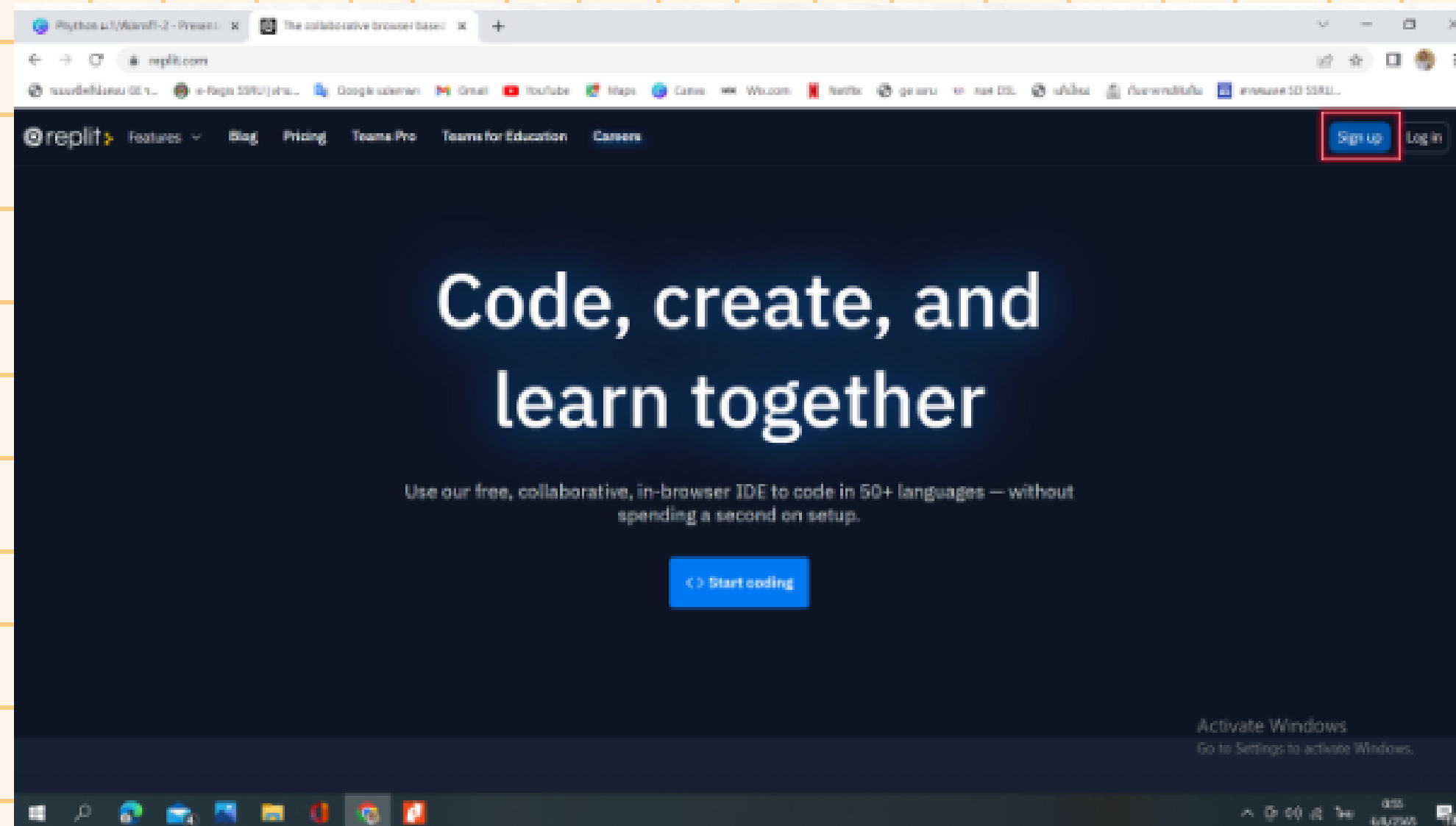
Repl.it คือ เว็บไซต์สำหรับการเขียนโปรแกรมภาษาไพทอน ที่มีเครื่องมืออย่าง

source code editor, build automation tools and a debugger

มีให้ใช้งานอย่างครบถ้วน ซึ่งใช้ได้ทั้งรูปแบบฟรี และเสียรายเดือน และสามารถใช้ได้ทุกแพลตฟอร์ม ทั้ง PC android ios



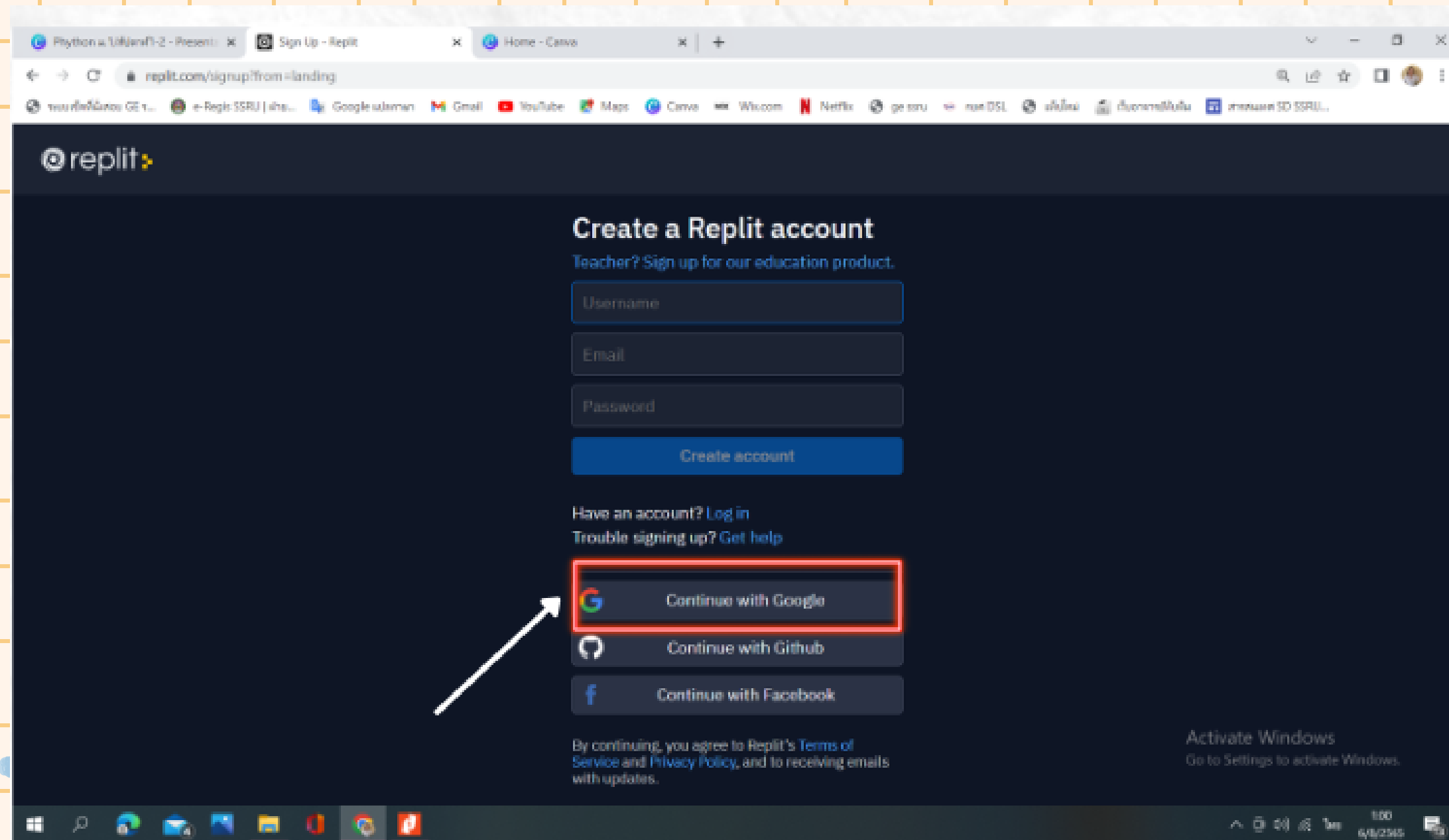
# การเข้าใช้งานเว็บไซต์ Repl.it



- 1.เข้าเว็บไซต์ <https://replit.com/>
- 2.กด Sign Up เพื่อสมัครสมาชิก



# การเข้าใช้งานเว็บไซต์ Repl.it



3. เลือก Continue with Google  
เพิ่มบัญชี E-mail ของโรงเรียน

# วิธีการสร้างไฟล์เพื่อเขียนโปรแกรม

**1.คลิก Create**

**2.เลือกภาษา Python**

**3.ตั้งชื่อไฟล์**

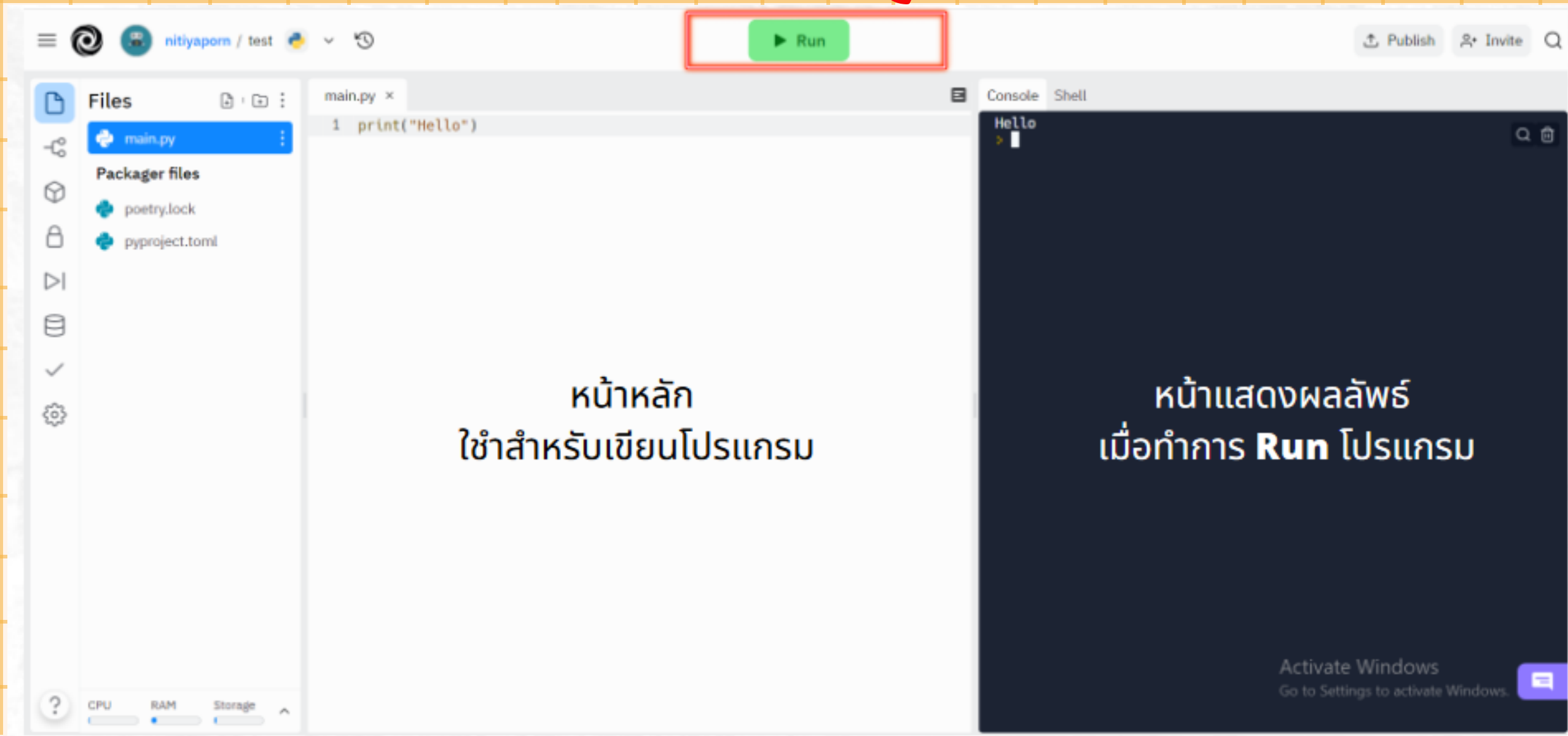
**4.คลิก Create Repl**

The image shows a step-by-step guide for creating a new Repl on the Replit website. It consists of four numbered steps with red arrows pointing to the corresponding UI elements:

- 1.คลิก Create**: A red arrow points to the '+ Create' button in the top navigation bar of the Replit user interface.
- 2.เลือกภาษา Python**: A red arrow points to the 'Python' option in the 'Template' dropdown menu of the 'Create a Repl' dialog box.
- 3.ตั้งชื่อไฟล์**: A red arrow points to the 'Title' input field in the 'Create a Repl' dialog box, which contains the placeholder text 'Name your Repl'.
- 4.คลิก Create Repl**: A red arrow points to the '+ Create Repl' button at the bottom of the 'Create a Repl' dialog box.

# เขียนโปรแกรม

ปุ่ม Run โปรแกรม



หน้าหลัก  
ใช้สำหรับเขียนโปรแกรม

หน้าแสดงผลลัพท์  
เมื่อทำการ **Run** โปรแกรม

# ฟังก์ชัน print

คือ ฟังก์ชันคำสั่งในการแสดงผลผ่านทางหน้าจอ โดยสามารถแสดงผลข้อมูลได้ทั้งตัวอักษร และ ตัวเลข

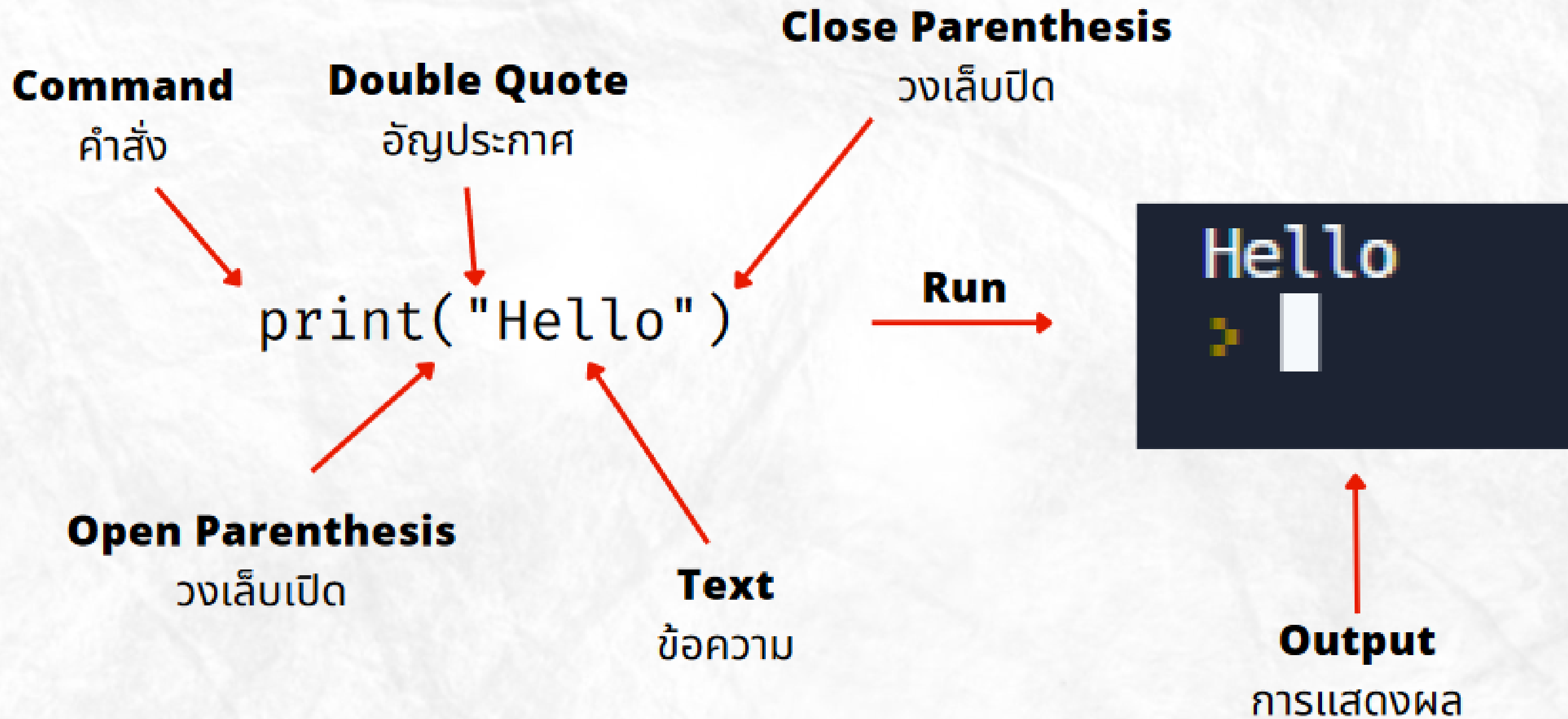
<code>print("Hello")</code>	Hello
<code>print("Hello"*3)</code>	HelloHelloHello
<code>print("Hello", "GIFT")</code>	Hello GIFT
<code>print(10+10)</code>	20

# ฟังก์ชัน print

<pre>print("Hello\nMy son")</pre>	Hello My son
<pre>print(20*5)</pre>	100
<pre>print("10"+"10")</pre>	1010
<pre>print(6/3)</pre>	2.0



# โครงสร้างฟังก์ชัน print



# โครงสร้างฟังก์ชัน print

ฟังก์ชันในวงเล็บสามารถใส่ อาร์กิวเมนต์ ที่ค่าก็ได้  
แต่ต้องใส่เครื่องหมาย จุลภาค Comma

```
print("Hello", "GIFT")
```

Run

```
Hello GIFT
```

เครื่องหมาย**Comma**  
จุลภาค

**Output**  
การแสดงผล

# โครงสร้างฟังก์ชัน print

การเขียนโปรแกรมให้ขึ้นบรรทัดใหม่โดยใช้ห้สควบคุมคือ \n

หน้าคำที่ต้องการขึ้นบรรทัดใหม่ \n

```
1 print("Hello\nMy Name is Nitiyaporn")
```

**Output**

การแสดงผล

```
Hello
My Name is Nitiyaporn
> |
```

หรือ

```
1 print("Hello","\n","My Name is Nitiyaporn")
```

หน้าคำที่ต้องการขึ้นบรรทัดใหม่ "\n"

```
Hello
My Name is Nitiyaporn
> |
```

# โครงสร้างฟังก์ชัน print

ฟังก์ชันยังมี keyword อาร์กิวเมนต์ sep

ซึ่งเป็นตัวแบ่งหากอาร์กิวเมนต์ที่ส่งเข้าไปนั้นมากกว่า 1 ตัว

โดยให้ใส่อักขระที่เป็นตัวแบ่งลงในสัญลักษณ์ฟลทอง (Single quote)

```
1 print("Hello", "my name is", "GIFT", sep='//') |
```

Single quote  
(ฟลทอง)

keyword  
อาร์กิวเมนต์ **sep**

อักขระที่ใช้แบ่ง  
อาร์กิวเมนต์

```
Hello//my name is//GIFT  
> 
```

Output  
การแสดงผล

# การเขียนโปรแกรม



The image shows a screenshot of a code editor interface. At the top, there is a navigation bar with a user profile icon, the name 'nitiyaporn / test', and a green 'Run' button. Below the navigation bar is a file explorer on the left showing a file named 'main.py'. The main editor area displays the code: 

```
1 print("Hello")
```

. A red box highlights this line of code. To the right, a terminal window shows the output: 

```
Hello
```

. A red box highlights the output. Below the code editor, there is a larger, semi-transparent box containing the code: 

```
1 print("Hello")
```

. A red arrow points from the code in the editor to this larger box. To the right of this box, there is another terminal window showing the output: 

```
Hello
```

. A red arrow points from the terminal output in the editor to this larger terminal window.



# ERROR

การเจอจุดผิดพลาด หรือ **error** ถือเป็นเรื่องปกติของการเขียนโปรแกรมมักมีการเกิด **Error** ใหญ่ด้วยกันอยู่ 3 ประเภทมีดังนี้

- 1 ข้อผิดพลาดที่เกิดจากการใช้คำสั่งผิดรูปแบบที่กำหนดหรือก็คือ **(Syntax Error)**
- 2 ข้อผิดพลาดที่เกิดระหว่างการทำงานของคำสั่ง **(Runtime Error)**
- 3 ข้อผิดพลาดที่เกิดจากการประมวลผลของกลุ่มคำสั่ง แต่ได้ผลลัพธ์ที่ไม่ตรงตามความต้องการ **(Logic Error)**

โดยในโปรแกรม **IDE** ส่วนใหญ่จะมีการช่วยระบุ **error** ให้

\*IDE ย่อมาจาก Integrated Development Environment คือ เครื่องมือที่ช่วยในการพัฒนาโปรแกรม โดยมีสิ่งอำนวยความสะดวกต่างๆ

# การเขียนโปรแกรม

บรรทัดที่เกิด **Error**

```
File "main.py", line 2
print("Nitiyaporn")
^
SyntaxError: EOL while scanning string literal
```

The screenshot shows a code editor with a file named 'main.py' containing the following code:

```
1 print("Hello","GIFT")
2 print("Nitiyaporn")
3 print(20*5)
```

The console output shows the error message: 'SyntaxError: EOL while scanning string literal' with a caret pointing to the end of the second line of code.

สาเหตุที่เกิด **Error**



ใบงานที่1 เขียนโปรแกรมคำสั่งอย่างง่าย  
ให้นักเรียนเขียนโปรแกรมภาษา  
Python อย่างง่าย  
หน้าจอแสดงผลต้องประกอบด้วย

- 1.ชื่อ
- 2.นามสกุล
- 3.ชื่อเล่น
- 4.เลขที่
- 5.ระดับชั้น

