

# Programming For Beginners With Common Lisp

unsigned\_nerd

14 พฤษภาคม พ.ศ. 2563

## สารบัญ

1	การเริ่มต้น	1
2	Common Lisp Package, ASDF, System และ Quicklisp	3
3	การติดตั้งระบบเพื่อเริ่มเขียนโปรแกรมด้วยภาษา Common Lisp	4

## 1 การเริ่มต้น

การเขียนโปรแกรม คือ การเขียนชุดคำสั่งเพื่อสั่งให้คอมพิวเตอร์ทำงานตามที่เรากำหนด

เราเขียนชุดคำสั่งด้วย Programming Language

โดยทั่วไปหากเราพูดถึง Programming Language เราจะหมายถึงภาษาแบบ High-Level เช่น Common Lisp, Python, Java, PHP, Perl, C, JavaScript เป็นต้น แต่ไม่ได้หมายถึงภาษา Assembly<sup>1</sup>

ภาษา Common Lisp เป็นภาษาที่ปัจจุบันไม่ได้รับความนิยมเลยทุกอย่างที่มี Feature ต่างๆมากมาย โดยเฉพาะอย่างยิ่ง Macro ใน Common Lisp ที่ทำให้เราสามารถเพิ่ม Syntax ต่างๆที่ต้องการเองได้ ทำให้เราสามารถเขียนโค้ดที่มีความกระชับและสั้นได้ง่าย<sup>2</sup> ตัวอย่างเช่น หากเราต้องการเขียนโปรแกรมที่ทำการอ่านข้อมูลแบบ Plain Text จาก Standard Input แล้วทำการใส่ Prefix String "Common Lisp is fun?: " นำหน้าทุกบรรทัด ก่อนที่จะพิมพ์ข้อมูลในแต่ละบรรทัดออกไปยัง Standard Output โดยปกติเราอาจเขียนโค้ดดังนี้:

---

<sup>1</sup> (let (line)  
<sup>2</sup> (loop for line = (read-line \*standard-input\* nil 'eof)

<sup>1</sup>ภาษา Assembly เป็น Low-Level Programming Language

<sup>2</sup>การเขียนโค้ด หากเขียนได้ยิ่งสั้นก็ถือว่ายิ่งดี โค้ดที่สั้นย่อมมีโอกาสเกิด Bug ได้น้อยกว่าโค้ดที่ยาว อีกทั้งยังสามารถอ่านได้เข้าใจได้ง่ายกว่าอีกด้วย ท่านคงเคยได้ยินคำกล่าวที่ว่า การเขียน Function ใดๆ ไม่ควรเขียนให้มีความยาวเกินความสูงของหน้าจอคอมพิวเตอร์ของท่าน

```
3   until (eq line 'eof)
4   do
5     (format t "Common Lisp is fun? ~A~%" line)))
```

---

ซึ่งจะเห็นได้ว่ามันดูเข้าใจยากและรายละเอียดเยอะไปหมด ทั้ง loop, until, do, 'eof และ nil แต่หากเราเขียน Macro ดังนี้:

---

```
1 (defmacro for-each-$line-in (in-stream &rest body)
2   (let (( $line (intern (symbol-name '$line))))
3     `(let (, $line)
4       (loop for , $line = (read-line ,in-stream nil 'eof)
5         until (eq , $line 'eof)
6         do
7           ,@body))))
```

---

ซึ่งจะเห็นได้ว่าเป็นเพียงการลอกโค้ดที่เราเขียนไว้ก่อนหน้านี้มาสร้างเป็น Macro<sup>3</sup> จะทำให้เราสามารถเขียนโค้ดใหม่ได้ดังนี้:

---

```
1 (for-each-$line-in *standard-input*
2   (format t "Common Lisp is fun? ~A~%" $line))
```

---

ซึ่งสั้นกว่าเดิมมาก  
ท่านผู้อ่านที่ยังไม่รู้ Macro คืออะไรอาจจะเห็นโค้ดนี้แล้วเห็นว่าสามารถใช้การเขียน Function ทำได้เหมือนกัน ซึ่งหากลองเพ่งดูดีๆ จะพบว่าไม่สามารถทำได้ โดยมีจุดที่น่าสังเกตดังนี้

1. a

2. b

สามารถดูการจัดอันดับภาษาคอมๆ ที่ได้รับความนิยมล่าสุดได้ที่นี้: <http://pypl.github.io/PYPL.html>

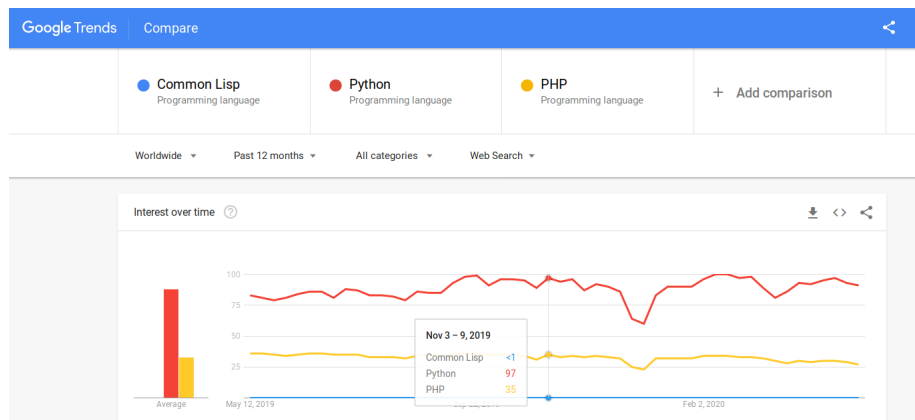
อันดับหนึ่งคือภาษา Python ส่วนภาษา Common Lisp ไม่ติดอันดับบน PYPL เลย ทาง PYPL แนะนำว่าถ้าหากอยากทราบความนิยมของภาษาที่ไม่ติดอันดับให้ใช้ Google Trends ดู

เมื่อลองใช้ Google Trends เปรียบเทียบความนิยมระหว่าง 3 ภาษา ได้แก่ Python, PHP และ Common Lisp เราจะพบว่ากระแสความนิยมในภาษา Common Lisp นั้นต่ำมาก ดังจะเห็นได้จากรูปที่ 1:

อย่างไรก็ตาม แม้ว่า Common Lisp จะไม่ได้รับความนิยมในปัจจุบัน แต่ก็ไม่ได้หมายความว่ามันจะไม่เหมาะกับการใช้ เป็นเครื่องมือในการเรียนรู้การเขียนโปรแกรมสำหรับมือใหม่

---

<sup>3</sup>ซึ่งอ่านเข้าใจยากกว่าเก่า แต่โดยปกติ Macro จะอยู่ในลักษณะของ Package ให้เราเรียกใช้ เราจึงไม่ต้องอ่านโค้ดที่ซับซ้อนเหล่านั้นเวลาใช้งาน



รูปที่ 1: Language popularity

เมื่อท่านได้เรียนรู้การเขียนโปรแกรมด้วยภาษา Common Lisp แล้ว ท่านจะสามารถเรียนภาษาอื่นๆเพิ่มเติมได้ง่ายขึ้นมาก

เป็นที่ยอมรับกันโดยทั่วไปว่าในชีวิตหนึ่งของคนเรา เราย่อมต้องเรียนภาษาคอมพิวเตอร์หลายภาษากันอยู่แล้ว แล้วทำไมจึงไม่ลองเรียนภาษา Common Lisp กันเล่า!

**รู้หรือไม่?** ภาษา Logo (เจ้าเต้าน้อย Logo) ที่หลายๆท่านอาจได้เคยเรียนในสมัยเด็กๆเป็น Dialect หนึ่งของภาษา Lisp นี่ก็ยิ่งชี้ให้เห็นว่า Common Lisp เหมาะกับการเป็นภาษาเขียนโปรแกรมแรกของทุกท่านขนาดไหน

## 2 Common Lisp Package, ASDF, System และ Quicklisp

เรื่อง Package, ASDF, System และ Quicklisp เป็นเรื่องหนึ่งที่ผู้เขียนมีความสับสนมากในช่วงเริ่มต้นของการเรียนภาษา Common Lisp

ความสับสนที่มากเกินไป กับ Documentation ที่ไม่ดีนักก็ทำให้เกิดผลเสียต่อความนิยมของภาษาออกมาหนึ่งได้

เรื่องนี้เกี่ยวกับระบบ Software Library ใน Common Lisp

สำหรับมือใหม่ ให้ลองทำตามนี้ก่อน จะได้ไม่สับสน แล้วเมื่อเชี่ยวชาญแล้วค่อยปรับแต่งตามใจชอบ

Software Library ต่างๆใน Common Lisp จะอยู่ในสิ่งที่เรียกว่า System

System หนึ่งๆจะประกอบไปด้วย Package ตั้งแต่ 1 อันขึ้นไป โดยจะใช้ ASDF เป็นตัวบอกไฟล์ต่างๆของ Package ทั้งหลายนั้นอยู่ตรงไหน

เวลาเราจะเปิด Project ใหม่ สมมุติว่าชื่อ hello-world ให้เราไปที่ Directory ชื่อ ~/common-lisp/ แล้วสร้าง directory ชื่อ hello-world/ ขึ้นมา ซึ่งจะเป็นที่ๆเราจะใส่ Source Code ของเราในนั้น

ต่อมาเรามี Project ใหม่ สมมุติว่าชื่อ goodbye-universe เราก็สร้าง Directory ชื่อ ~/common-lisp/goodbye-universe/ แล้วใส่ Source Code ของ Project นี้ไปในนั้น

สมมุติว่าเรามี Function ใน Project ชื่อ hello-world ที่สามารถนำมาใช้ใน Project ชื่อ goodbye-universe ด้วย ทางหนึ่งที่ทำได้คือ Copy & Paste Source Code ของ Function นั้นมาใส่ แต่วิธีนี้ไม่ดี

วิธีที่ดีกว่าคือ เอา Function นั้นไปสร้างเป็น Package แล้วให้ Project ชื่อ hello-world และ goodbye-universe เรียกใช้ Function นั้นรวมกันผ่านระบบ Package

เมื่อเราจัดเอา Function และ/หรือ โค้ดส่วนอื่นๆที่สามารถใช้ร่วมกันได้ระหว่างหลาย Project มาจัดใส่ Package ต่างๆแล้ว เราก็ต้องมาคิดต่อว่าจะต้องทำอะไร Sbc1 จึงจะรู้ว่าไฟล์ Package ต่างๆอยู่ที่ไหน ซึ่งนี่จะเป็นหน้าที่ของ ASDF

ASDF คือเครื่องมือที่จะบอก Sbc1 ว่าไฟล์ต่างๆอยู่ตรงไหน

### 3 การติดตั้งระบบเพื่อเริ่มเขียนโปรแกรมด้วยภาษา Common Lisp

Common Lisp มีหลาย Implementation เราเลือกใช้ Sbc1 ซึ่งเป็น Common Lisp Implementation ที่ได้รับความนิยมที่สุด

ผู้เขียนใช้ Debian 10 เป็น Operating System

ทำการติดตั้ง Sbc1, Quicklisp และ un-utils บนคอมพิวเตอร์ของคุณโดยทำตามขั้นตอนในลิงค์นี้: <https://github.com/unsigned-nerd/un-utils>

Sbc1

ดังได้กล่าวไปก่อนหน้านี้ Sbc1 เป็น Implementation หนึ่งของ Common Lisp

เป็นโปรแกรมที่ใช้คอมไพล์และรันโปรแกรมที่เราเขียนขึ้นด้วยภาษา Common Lisp

Quicklisp

Quicklisp เป็นโปรแกรมเชิงระบบที่นิยมใช้ในการติดตั้ง Systems ต่างๆจากผู้พัฒนาคนอื่น

System ใน Common Lisp ก็คือ Library ต่างๆที่เราเรียกกันในภาษาอื่นนั่นเอง

โดยทั่วไป ให้เราคิดเสียว่า หากเราต้องการดาวน์โหลด System ของคนอื่นมาใช้ โดยคิดว่าเราจะใช้อย่างเดียว ไม่ได้ต้องการจะแก้ไขอะไรมัน ก็ควรจะใช้ Quicklisp ในการดาวน์โหลดและติดตั้ง

แต่ถ้าเราต้องการเขียน System เอง หรือ ต้องการแก้ไข System ของผู้อื่น ก็ให้ใช้เครื่องมือชื่อ ASDF ในการจัดการ

ASDF เป็นเครื่องมือที่ใช้ในการจัดการ

un-utils

un-utils เป็น Common Lisp System ที่ทางผู้เขียนพัฒนาขึ้นมา ซึ่งมี Package ที่น่าสนใจชื่อ un-utils.simple-syntax

un-utils.simple-syntax นี้จะ

---

```
1 // Hello.java
2 import javax.swing.JApplet;
3 import java.awt.Graphics;
4
5 public class Hello extends JApplet {
6     public void paintComponent(Graphics g) {
7         g.drawString("Hello, world!", 65, 95);
8     }
9 }
```

---

---

```
1 ; comment
2 ( setf a 'a)
```

---