

Programming For Beginners With Common Lisp

unsigned_nerd

10 พฤษภาคม พ.ศ. 2563

สารบัญ

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

1 การเริ่มต้น

การเขียนโปรแกรม คือ การเขียนชุดคำสั่ง เพื่อสั่งให้คอมพิวเตอร์ทำงานตามที่เรากำหนด
เราเขียนชุดคำสั่งด้วย Programming Language

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

ภาษา Common Lisp เป็นภาษาที่ปัจจุบันไม่ได้รับความนิยมมากนัก

สามารถดูการจัดอันดับภาษาคอมที่ได้รับความนิยมล่าสุดได้ที่: <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 ดังจะเห็นได้จากภาพประกอบด้านล่าง

อย่างไรก็ตาม แม้ว่า Common Lisp จะไม่ได้รับความนิยมในปัจจุบัน แต่ก็ไม่ได้หมายความว่ามันจะไม่เหมาะกับการใช้เป็นเครื่องมือในการเรียนรู้การเขียนโปรแกรมสำหรับมือใหม่

เมื่อท่านได้เรียนรู้การเขียนโปรแกรมด้วยภาษา Common Lisp แล้ว ท่านจะสามารถเรียนภาษาคอมฯอื่นๆเพิ่มเติมได้ง่ายขึ้นมาก
เป็นที่ยอมรับกันโดยทั่วไปว่าในชีวิตหนึ่งของคนเรา เราย่อมต้องเรียนภาษาคอมพิวเตอร์หลายภาษา แล้วทำไมไม่ลองเริ่มด้วยภาษา Common Lisp กันเล่า!

รู้หรือไม่ว่า? 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 เป็นเรื่องหนึ่งที่ผู้เขียนมีความสับสนมากในช่วงเริ่มแรกของการเรียนภาษา Common Lisp

ความสับสนที่มากเกินไป กับ Documentation ที่ไม่ดีนัก ก็ทำให้เกิดผลเสียต่อความนิยมของภาษาคอมฯหนึ่งๆได้

เรื่องนี้เกี่ยวกับระบบ Software Library หรือ Software Package ใน Common Lisp สำหรับมือใหม่ ให้ลองทำตามนี้ก่อน จะได้ไม่สับสน แล้วเมื่อเชี่ยวชาญแล้วค่อยปรับแต่งตามใจชอบ

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 ทางหนึ่งที่ได้ก็คือ Copy & Paste Source Code ของ Function นั้นมาด้วย แต่วิธีนี้ไม่ Cool เท่าไหร่

วิธีที่ Cool กว่าคือ เอา Function นั้นไปทำเป็น Package แล้วให้ Project ชื่อ hello-world และ goodbye-universe เรียกใช้ร่วมกัน

พอเรามี Source Code เยอะๆ เริ่มมีการใช้งาน Source Code ชุดเดิมซ้ำๆกันในหลายๆ Project เราก็เลยเอา Source Code ชุดต่างๆมาจัดกลุ่ม ใส่เข้าไปอยู่ในแต่ละ Package

เมื่อเราจัดเอา Source Code ชุดต่างๆใส่เข้าไปแต่ละ Package แล้ว เราก็ต้องมาคิดต่อว่าจะทำอย่างไรให้ Project อื่นๆ

3 การติดตั้งระบบเพื่อเริ่มเขียนโปรแกรมด้วยภาษา Common Lisp

Common Lisp มีหลาย Implementation เราเลือกใช้ Sbcl ซึ่งเป็น Common Lisp Implementation ที่ได้รับความนิยมที่สุด

ผู้เขียนใช้ Debian 10 เป็น Operating System

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

Sbcl

ดังได้กล่าวไปก่อนหน้านี้ Sbcl เป็น Implementation หนึ่งของ Common Lisp

เป็นโปรแกรมที่ใช้คอมไพล์และรันโปรแกรมที่เราเขียนขึ้นด้วยภาษา Common Lisp

Quicklisp

Quicklisp เป็นโปรแกรมเชิงระบบที่นิยมใช้ในการติดตั้ง Systems ต่างจากผู้พัฒนาคนอื่น

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)
```
