

การทดสอบ (Testing)

ใครว่า Programming มันง่าย จริงๆ แล้วไม่เลย ยิ่งถ้าเจอกับ Programmer ที่ชั่วโมงบินน้อย บาดแผลบนร่างกายยังน้อย หรือยังเก่าไม่พอ ความมั่นใจในเรื่องของคุณภาพก็น้อยตามไปด้วย ดังนั้นการทดสอบ (Testing) จึงเป็นส่วนสำคัญมากในกระบวนการพัฒนาซอฟต์แวร์ ดังนั้นการเขียน Code Test จึงเป็นเรื่องที่สำคัญที่ช่วยให้เรามั่นใจว่าคุณภาพและสุขภาพของ Code ของเรายังดีอยู่

Go ได้เตรียมวิธีการเขียน Test แบบใช้งานได้ง่ายๆ ไว้ให้แล้ว ลองมาเริ่มเขียน Test กันจาก Code ที่เราสร้างไว้จากบทที่ 11 ใน folder math ให้สร้างไฟล์ใหม่ชื่อว่า math_test.go ตามตัวอย่าง Code ด้านล่างนี้

```
package math

import "testing"

func TestAverage(t *testing.T) {
    var v float64
    v = Average([]float64{1,2})
    if v != 1.5 {
        t.Error("Expected 1.5, got ", v)
    }
}
```

แล้วพิมพ์คำสั่งเพื่อทำการ Run

```
go test
```

จะได้ผลลัพธ์ออกมาตามนี้

```
$ go test
PASS
ok   golang-book/chapter11/math    0.032s
```

คำสั่ง go test ใช้สำหรับสั่ง Run Test ที่อยู่ใน Folder นั้นๆ เมื่อจะเขียน Test คราใด ให้เริ่มต้นชื่อ function ด้วยคำว่า Test ครานั้น และใช้ *testing.T เป็น argument ดังนั้นจากโจทย์ของเรา เราต้องเขียน Test ขึ้นมาเพื่อทดสอบ function Average ดังนั้นชื่อ function Test เราจึงเป็น TestAverage

เมื่อเรามี Test แรกที่ใช้ในการทดสอบการหาค่าเฉลี่ยของ [1,2] ที่จะต้องได้ค่าเฉลี่ยออกมาเป็น 1.5 แล้วนั้น ลองเพิ่มความสนุกในการทดสอบลงไปด้วย Test Cases กรณีต่างๆ เพิ่มเข้าไปเพื่อให้การทดสอบนั้นครอบคลุมมากขึ้น ตามตัวอย่างด้านล่าง

```
package math

import "testing"

type testpair struct {
    values []float64
    average float64
}

var tests = []testpair{
    { []float64{1,2}, 1.5 },
    { []float64{1,1,1,1,1,1}, 1 },
    { []float64{-1,1}, 0 },
}

func TestAverage(t *testing.T) {
    for _, pair := range tests {
        v := Average(pair.values)
        if v != pair.average {
            t.Error(
                "For", pair.values,
                "expected", pair.average,
                "got", v,
            )
        }
    }
}
```

เราสร้าง struct มาเพื่อกำหนดค่า input และ output สำหรับการเรียกใช้งานใน function แล้วเราก็สร้างชุดของข้อมูลไว้ใน struct (pairs) เมื่อเราสั่งคำสั่ง go test คราใด ค่า input และ output แต่ละคู่ใน struct จะถูกส่งเข้าไปใน function TestAverage เพื่อทำการทดสอบ

ปัญหาท้าทาย

- การเขียนชุด Test ให้ดีไม่ใช่เรื่องง่าย แต่ก็ไม่ยากที่จะทำโดยการ ต้องฝึกฝน และเขียนบ่อยๆ จะเกิดการพัฒนาทักษะและเข้าใจมากยิ่งขึ้นเรื่อยๆ ยกตัวอย่างเช่น จะเกิดอะไรขึ้นถ้าค่าของ

[]float64{ } เป็นค่าว่าง (empty list)? เราจะแก้ไข function อย่างไรให้ return 0 กลับมาให้
ในกรณีนี้?

- เขียน Test เพื่อทดสอบ function Min และ Max ที่สร้างไว้จากบทที่แล้ว