The Go Programming Language

TDD

Anuchit Prasertsang
Developer

Testing

- *_test.go files
- go test

Test Functions

Each test file must import the testing package. Test functions have the following signature:

```
import "testing"
func TestName(t *testing.T) {
  // ...
}
```

Test function names must begin with **Test**; the optional suffix **Name** must begin with a **capital**
letter:

```
func TestSin(t *testing.T) { /* ... */ }
func TestCos(t *testing.T) { /* ... */ }
func TestLog(t *testing.T) { /* ... */ }
```

The **t** parameter provides methods for reporting test failures and logging additional information

information.

Exercise

```
package main

func Sum(x int, y int) int {
    return x + y
}

func main() {
    Sum(5, 5)
}
```

Test sum function

```
package main
import "testing"

func TestSum(t *testing.T) {
    total := Sum(5, 5)
    if total != 10 {
      t.Errorf("Sum was incorrect, got: %d, want: %d.", total, 10)
    }
}
```

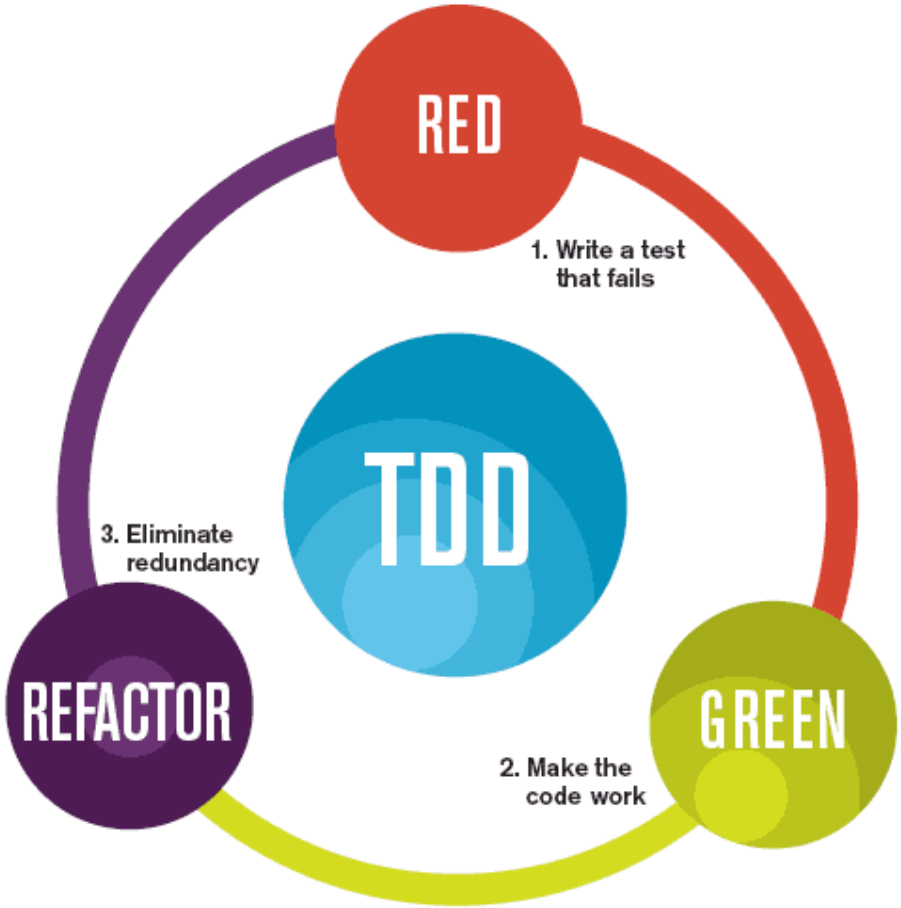
FizzBuzz

```
input  => output
  1    => "1"
  2    => "2"
  3    => "Fizz"
  4    => "4"
  5    => "Buzz"
  6    => "Fizz"
  7    => "7"
  8    => "8"
  9    => "Fizz"
  10   => "Buzz"
  11   => "11"
  12   => "Fizz"
  13   => "13"
  14   => "14"
  15   => "FizzBuzz"
```

What is TDD?

- **Test-Driven Development (TDD)** is a technique for building software that guides software development by writing test. (Martin Fowler's definition)
- Developers write unit tests (NOT testers) and *then* code.

TDD mantra



The mantra of Test-Driven Development (TDD) is "red, green, refactor."

TDD mantra

- RED - write a little test that doesn't work, perhaps doesn't even compile at first.
- Green - make the test work quickly, committin whatever sins necessary in the process.
- Refactor - eliminate all the duplication and smells created in just getting the test to work.

Uncle Bob describes TDD with three rules:

- You are not allowed to write any production code unless it is to make a failing unit test pass.
- You are not allowed to write any more of a unit test than is sufficient to fail; and compilation failures are failures.
- You are not allowed to write any more production code than is sufficient to pass the one failing unit test.

TDD FizzBuzz

```
input  => output
  1    => "1"
  2    => "2"
  3    => "Fizz"
  4    => "4"
  5    => "Buzz"
  6    => "Fizz"
  7    => "7"
  8    => "8"
  9    => "Fizz"
  10   => "Buzz"
  11   => "11"
  12   => "Fizz"
  13   => "13"
  14   => "14"
  15   => "FizzBuzz"
```


Captcha

12

Phone Number Normalizer

1234567890

```
1234567890
123 456 7891
(123) 456 7892
(123) 456-7893
123-456-7894
123-456-7890
1234567892
(123)456-7892
```


ROMAN coverter

ROMAN NUMERALS CHART

1 TO 100

1	I	21	XXI	41	XLI	61	LXI	81	LXXXI
2	II	22	XXII	42	XLII	62	LXII	82	LXXXII
3	III	23	XXIII	43	XLIII	63	LXIII	83	LXXXIII
4	IV	24	XXIV	44	XLIV	64	LXIV	84	LXXXIV
5	V	25	XXV	45	XLV	65	LXV	85	LXXXV
6	VI	26	XXVI	46	XLVI	66	LXVI	86	LXXXVI
7	VII	27	XXVII	47	XLVII	67	LXVII	87	LXXXVII
8	VIII	28	XXVIII	48	XLVIII	68	LXVIII	88	LXXXVIII
9	IX	29	XXIX	49	XLIX	69	LXIX	89	LXXXIX
10	X	30	XXX	50	L	70	LXX	90	XC
11	XI	31	XXXI	51	LI	71	LXXI	91	XCI
12	XII	32	XXXII	52	LII	72	LXXII	92	XCII
13	XIII	33	XXXIII	53	LIII	73	LXXIII	93	XCIII
14	XIV	34	XXXIV	54	LIV	74	LXXIV	94	XCIV
15	XV	35	XXXV	55	LV	75	LXXV	95	XCV
16	XVI	36	XXXVI	56	LVI	76	LXXVI	96	XCVI
17	XVII	37	XXXVII	57	LVII	77	LXXVII	97	XCVII
18	XVIII	38	XXXVIII	58	LVIII	78	LXXVIII	98	XCVIII
19	XIX	39	XXXIX	59	LIX	79	LXXIX	99	XCIX
20	XX	40	XL	60	LX	80	LXXX	100	C

Rotation Array

```
[1, 2, 3, 4, 5] => [5, 1, 2, 3, 4] => [4, 5, 1, 2, 3]
["c", "d", "e", "a", "b"] => ["b", "c", "d", "e", "a"]
```

Money change

- หาเงินทอน
- ราคาที่ต้องจ่าย

- เงินที่ลูกค้าให้มา
- ส่งกลับค่าเงินทอน

```
- slice ของธนบัตรหรือเหรียญคู่กับจำนวน
[(1000,2), (50,1), (2, 1), (1,1)]
```

- มีธนบัตร กับ เหรียญ 1000,500,100,50,20,10,5,2,1

Benchmark Functions

- In Go, a benchmark function looks like a test function, but with the **Benchmark**

prefix and a *** testing.B** parameter that provides most of the same methods as a *testing.T, plus a few ext ra re lated to per for mance measurement. It als o exp oses an integer field N

plus a few extra related to performance measurement. It also exposes an integer field N, which
specifies the number of times to perform the operation being measured.
Here’sabenchmark for IsPalindrome that calls it N times in a loop.

```
import "testing"
func BenchmarkIsPalindrome(b *testing.B) {
  for i := 0; i < b.N; i++ {
    IsPalindrome("A man, a plan, a canal: Panama")
  }
}
```


Benchmark Palindrome

```
func BenchmarkIsPalindrome(b *testing.B) {
  for i := 0; i < b.N; i++ {
    IsPalindrome("A man, a plan, a canal: Panama")
  }
}
```

Thank you

Anuchit Prasertsang
Developer
anuchit.prasertsang@gmail.com (mailto:anuchit.prasertsang@gmail.com)
https://github.com/AnuchitO (https://github.com/AnuchitO)
@twitter_AnuchitO (http://twitter.com/twitter_AnuchitO)