GoLang

===Ghi chú

//Ghi chú bằng cách này

/\*Hoặc cách này\*/

===Biến

--Các kiểu dữ liệu

- int: int, int8, int16, ...

-> Số nguyên không dấu: uint

- float: float32, float64

- string

- bool

--Tên biến

-Bắt đầu bằng chữ cái hoặc \_

-Không giới hạn độ dài tên

-Không có khoảng trắng

-Không chứa từ đặc biệt Go hoặc kí tự đặc biệt

-- Đặt biến

var <tenbien> <kieudulieu>

var <tenbien> <kieudulieu> = <giatri>

var <tenbien1>, <tenbien2> <kieudulieu> = <giatri>, <giatri>

<tenbien> := <giatri> -> Không thể sử dụng ngoài hàm

--Ex

Var std sring

Var std tring = “Tam”

Var std = “Tam”

X:=2

Var num = 50

Var x, y, z = 1, 2, 3

Var num int = 90

Var mybool bool = true

Var a, b = 6, “Hello”

===Hằng số

const <tên hằng> <kiểu> = <giá trị>

-- ex

const pi = 3.14

===Print

fmt.Print(<tenbien>)

fmt.Print("Noi dung")

fmt.Print("Noi dung",bien)

fmt.Print(i, "thông báo ", j)

fmt.Print("Giá trị i: %v và kiểu %T /n", i, i)

-> %v Giá trị biến

-> %#v Giá trị kiểu cú pháp Go

-> %t Kiểu dữ liệu biến

%s chuỗi

%q chuỗi có dấu "chuỗi"

%8s chuỗi căn phải 8

%-8s chuỗi căn trái 8

%t boole

%.2f số thập phân chính xác 2 chữ số

%e kí hiệu khoa học với e là số mũ

/t : TAB

/n : ENTER

===Mảng

var <tenmang> = [chieudai]<kieudulieu>{}

var <tenmang> = [...]<kieudulieu>{}

array\_name := [length]datatype{values}

array\_name := [...]datatype{values}

var arr1 = [3]int{1,2,3}

arr2 := [5]int{4,5,6,7,8}

arr1 := [5]int{1:10,2:40}

->[0 10 40 0 0]

fmt.Println(arr1)

len(a)

-Mảng bắt đầu từ 0

var cars = [4]string{"a","b","c","d"}

fmt.Print(cars)

fmt.Print(cars[1])

cars[0] = "opel"

arr1 := [6]int{1, 2, 3, 4, 5, 6}

arr2 := arr1[2:4]

->arr2 = {3,4)

--Tạo mảng

arr := make([]int, <chieudai>)

--Cộng mảng

slice\_name = append(slice\_name, element1, element2, ...)

slice2 = append(slice1, slice2)

===if

if <điều kiện> {

}

if <điều kiện> {

} else {

}

if <điều kiện 1> {

} else if <điều kiện 2> {

} else {

}

===switch case

switch <bien> {

case(dk1):

case(dk2):

case(dk3):

}

switch <bien> {

case(dk1):

case(dk2):

default:

}

===For

for i:=0; i <10; i++ {

if i==5{

break;

}

if i==3{

continue;

}

}

for <i>, <val> := range <array> {

fmt.Print("%v %v", <i>, <val>)

}

--Bỏ qua chỉ mục

for \_, val := range array {

fmt.Print("%v", i, val)

}

===Hàm

func myFunction(x int) int {

return x

}

func main() {

fmt.print(myFunction(3))

}

--Đặt tên biến trả về

func myFunction(x int, y int) (result int) {

result = x + y

return

}

--Hai biến trả về

func myFunction(x int, y string) (result int, txt1 string) {

result = x + x

txt1 = y + " World!"

return

}

func main() {

a, b := myFunction(5, "Hello")

fmt.Println(a, b)

}

===struct

type struct\_name struct {

member1 datatype;

member2 datatype;

member3 datatype;

...

}

type Person struct {

name string

age int

job string

salary int

}

func TestVar(a *int*, b *string*) (x *int*, y *string*) {

    x = a + 1

    y = b

    return

}

func main() {

    fmt.Print("Hello \n")

    a := 5

    b := "Xin chao"

    var x, y = TestVar(a, b)

    fmt.Print("x = ", x, " y = ", y, "\n")

}