Programas:

\*\*\*1\*\* primer programa\*\*\*\*

// Mi primer programa

package main

import "fmt"

func main() {

fmt.Println("Hola, Mundo")

}

\*\*\*2\*\* (tipo de Variables) \*\*\*\*

package main

import "fmt"

func main() {

// Declaracion de Variables

var myStr string = "Hola"

var myInt int = 100

var myFloat float64 = 45.12

fmt.Println(myStr, myInt, myFloat)

//================================

// Multiple Declaraciones

var (

employeeId int = 5

firstName, lastName string = "Satoshi", "Nakamoto"

)

fmt.Println(employeeId, firstName, lastName)

//================================

// syntax corta de declaración de variables

name := "Rudy Gómez"

edad, salario, esprogramador := 29, 50000.0, true

fmt.Println(name, edad, salario, esProgramador)

}

\*\*\*\*\*

package main

import "fmt"

func main() {

// Tipo inferenciado

var name = "Rajeev Singh" // La declaración del tipo de variable aquí es opcional(es inferida).

fmt.Printf("Variable 'name' es de tipo %T\n", name)

//================================

// Multiple variable declarations with inferred types

var firstName, lastName, age, salary = "Rita", "Valladare", 28, 50000.0

fmt.Printf("Nombre: %T, Apellido: %T, edad: %T, salario: %T\n",

firstName, lastName, age, salary)

}

\*\*\*\*\*

package main

import "fmt"

func main() {

var (

firstName, lastName string

age int

salary float64

isConfirmed bool

)

// firstName, lastName, age, salary = "Rita", "Valladare", 28, 50000.0

fmt.Printf("Nombre: %s, Apellido: %s, edad: %d, salario %f, esprogramdor: %t\n",

firstName, lastName, age, salary, isConfirmed)

}

\*\*\* 3 \*\*\*Tipos básicos \*\*\*\*\*\*\*\*

// tipo numerico

package main

import "fmt"

func main() {

var myInt8 int8 = 97

/\*

cuando no se declara el tipo de entero de manera explicita se infiere que tipo `int`

(El tipo por defecto para enteros)

\*/

var myInt = 1200

var myUint uint = 500

var myHexNumber = 0xFF // Use prefixo '0x' or '0X' para declarar numeros hexadecimales

var myOctalNumber = 034 // Use prefixo '0' para declarar números octales

var myFloat32 float32 = 4.5

var myFloat = 9.12 // // El tipo inferido es `float64` (El tipo por defecto para punto flotante)

fmt.Printf("%d, %d, %d, %#x, %#o %f %f\n", myInt8, myInt, myUint, myHexNumber, myOctalNumber, myFloat32, myFloat)

}

\*\*\*\*\*

// tipo caracteres

package main

import "fmt"

func main() {

var myByte byte = 'a'

var myRune rune = '♥'

fmt.Printf("%c = %d and %c = %U\n", myByte, myByte, myRune, myRune)

}

\*\*\*\*\*

// tipo booleano

package main

import "fmt"

func main() {

var myBoolean bool = true

var anotherBoolean = false // Inferred type

var truth = 3 <= 5

var falsehood = 10 != 10

// Short Circuiting

var res1 = 10 > 20 && 5 == 5 // Second operand is not evaluated since first evaluates to false

var res2 = 2\*2 == 4 || 10%3 == 0 // Second operand is not evaluated since first evaluates to true

fmt.Println(myBoolean, anotherBoolean, truth, falsehood, res1, res2)

}

\*\*\*\*\*\*

// números complejos

package main

import "fmt"

func main() {

// === Creating complex numbers ====

/\*

complex64: both real and imaginary parts are of float32 type.

complex128: both real and imaginary parts are of float64 type.

\*/

var x complex64 = 3.4 + 2.9i

var y = 5 + 7i // Type inferred as `complex128` (default type for complex numbers)

fmt.Println(x, y)

// Creating complex no from variables

var a1 = 4.5

var a2 = 7.1

var c = complex(a1, a2) // a1 + a2\*i won't work

fmt.Println(c)

// ===== Complex No Operations =====

var a = 3 + 5i

var b = 2 + 4i

var res1 = a + b

var res2 = a - b

var res3 = a \* b

var res4 = a / b

fmt.Println(res1, res2, res3, res4)

}

\*\*\*\*\*\*\*\*\*\*

// cadenas

package main

import "fmt"

func main() {

// Normal String (no contienen nuevas líneas, y pueden tener carácteres de escape como `\n`, `\t` etc)

var website = "\thttps://www.callicoder.com\t\n"

// Raw String (pueden agrupar lineas multiple líneas. Los carácteres de Escape no son interpretados)

var siteDescription = `\t\tCalliCoder es un blog de programación donde usted puede encontrar

guías practicas y tutoriales en lenguaje de programación,

desarrollo web, y desarrollo de aplicaciones de escritorios.\t\n`

fmt.Println(website, siteDescription)

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// conversión

package main

import "fmt"

func main() {

// Type Conversion

var a int64 = 4

var b int = int(a) //tipo de conversión explicita

var c float64 = 6.5

// tipo de conversión explicita

var result = float64(b) + c // Works

fmt.Println(result)

// ==================

// la sintaxi general para convertir un valor B a uno de tipo T es T(B)

var myInt int = 65

var myUint uint = uint(myInt)

var myFloat float64 = float64(myInt)

fmt.Println(myInt, myUint, myFloat)

}

\*\*\* 4 \*\*\* constantes

package main

import "fmt"

func main() {

// Constantes sin declarar tipo

const myFavLanguage = "Python"

const sunRisesInTheEast = true

fmt.Println(myFavLanguage, sunRisesInTheEast)

// declaración múltiple de constantes

const country, code = "India", 91 // sin declarar tipo

const ( // declarando tipo

employeeId string = "E101"

salary float64 = 50000.0

)

fmt.Println(country, code, employeeId, salary)

// Constantes declarando tipo

const typedInt int = 1234

const typedStr string = "Hi"

fmt.Println(typedInt, typedStr)

}

\*\*\*\* 5 \*\*\* control de flujo

// condición if

package main

import "fmt"

func main() {

// If Statement

var x = 25

if(x % 5 == 0) {

fmt.Printf("%d es un multiplo de 5\n", x)

}

// los Parentesis son Opcionales en la expresión condicional(y<0)

var y = -1

if y < 0 {

fmt.Printf("%d es negativo\n", y)

}

// if con una condición con la conjunción y(&&)

var age = 21

if age >= 17 && age <= 30 {

fmt.Println("Mi edad esta entre 17 y 30")

}

// If con una instruccion corta( n:=10) la condición si el resto de dividir n entre 2 es cero para determinar si n es par

if n := 10; n%2 == 0 {

fmt.Printf("%d es par\n", n)

}

}

\*\*\*\*\*\*

// condición if-else

package main

import "fmt"

func main() {

var edad = 18

if edad >= 18 {

fmt.Println("Usted es eligible para votar!")

} else {

fmt.Println("Usted no es eligible para votar!")

}

}

\*\*\*\*\*\*\*\*

// condición if-else-if

package main

import "fmt"

func main() {

var BMI = 21.0

if BMI < 18.5 {

fmt.Println("Usted esta bajo de peso");

} else if BMI >= 18.5 && BMI < 25.0 {

fmt.Println("Usted tiene un peso normal");

} else if BMI >= 25.0 && BMI < 30.0 {

fmt.Println("Usted esta pasado de peso")

} else {

fmt.Println("Usted esta obeso")

}

}

\*\*\*\*\*\*\*\*\*

// instrucción switch

package main

import "fmt"

func main() {

var diadesemana = 6

switch diadesemana {

case 1: fmt.Println("Lunes")

case 2: fmt.Println("Martes")

case 3: fmt.Println("Miercoles")

case 4: fmt.Println("Jueves")

case 5: {

fmt.Println("Viernes")

fmt.Println("Viernes de aroma. Fino")

}

case 6: {

fmt.Println("Sabado")

fmt.Println("Fin de semana. Yuujo!")

}

case 7: {

fmt.Println("Domingo")

fmt.Println("Fin de semana. Para dormir!")

}

default: fmt.Println("Dia invalido")

}

}

\*\*\*\*\*\*\*\*\*\*\*

switch-casos combinados

package main

import "fmt"

func main() {

switch diadesemana := 5; diadesemana {

case 1, 2, 3, 4, 5:

fmt.Println("Dia de semana normal")

case 6, 7:

fmt.Println("Día de fin de semana")

default:

fmt.Println("Día invalido")

}

}

\*\*\*\*\*\*\*\*\*\*

switch-sin expresion condicional

package main

import "fmt"

func main() {

var peso = 21.0

switch {

case peso < 18.5:

fmt.Println("Usted esta bajo de peso")

case peso >= 18.5 && peso < 25.0:

fmt.Println("Usted tiene un peso normal")

case peso >= 25.0 && peso < 30.0:

fmt.Println("Usted tiene sobre peso")

default:

fmt.Println("Usted esta obeso")

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*

for-loop

// imprime los valores de 0 hasta 9

package main

import "fmt"

func main() {

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

fmt.Printf("%d ", i)

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*

for-without-init-statement

package main

import "fmt"

// multiplos de 2 menores o iguales a 10

func main() {

i := 2

for ;i <= 10; i += 2 {

fmt.Printf("%d ", i)

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

for-without-increment-statement

// imprime las potencias de 2

package main

import "fmt"

func main() {

i := 2 // i se inicia en 2

for ;i <= 20; { // i debe ser menor a 20

fmt.Printf("%d ", i) // imprime los valores de que va tomando i

i \*= 2 // el último valor de i se multiplica por 2 y el valor se coloca en i

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

0-for-infinite

package main

func main() {

// Infinite Loop

for {

}

}

\*\*\*\*\*\*\*\*\*\*\*\*

for-break

// Evalua los número del 1 al 1oo y chequea cual puede ser dividido entre 5 y 3, lo imprime y para el proceso.

package main

import "fmt"

func main() {

for num := 1; num <= 100; num++ {

if num%3 == 0 && num%5 == 0 {

fmt.Printf("Primer numero positivo divisible por 5 y 3 es is %d\n", num)

break

}

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

for- continue

// imprime los números de 1 a 10 pero si es divisible por 2 no y continua,lo que equivale a imprime los impares menores que 10.

package main

import "fmt"

func main() {

for num := 1; num <= 10; num++ {

if num%2 == 0 {

continue;

}

fmt.Printf("%d ", num)

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*