# Go

1. **Go is statically typed**: Statically typed is a programming language characteristic in which variable types are explicitly declared and thus are determined at compile time
2. **GO is** compiled high-level programming language

## Data Type

1. Basic type
2. Composite type
3. Basic type:
   1. Numeric type:
      1. Integers
      2. Floats
      3. Complex Numbers
   2. String:
      1. String always in “” quotation.
   3. Byte:
   4. Rune: it use Unicode character.
   5. Boolean
4. Composite type:
   1. Aggregate type/Non-Reference type:
      1. Array
      2. Structure
   2. Reference type
      1. Pointer
      2. Slices
      3. Maps
      4. Function
      5. Channels
   3. Interface type

## Variable declaration

1. with var keyword:

var variable\_name type = value

var name string = “shuvo”

2. without var

Shuvo := name

## Flow Control

If, else, else if

For:

For- range:

## Function

1. Access modifier:

1. First letter of function Capital indicate that function is public.
2. First letter small indicate private function

2. Multiple value return

Func square(num float64) float64 int{

Return x,y

}

Variadic function:

func myFunc(numbers ...int) {

fmt.Println(numbers)

}

Defer:

Named Result Parameter

## Struct

type StructName struct {

Field1 datatype

Field2 datatype

}

type Player struct {

name string

age int

}

Unexported: The first word of variable should be small. We can access this variable from the other struct.

type Player struct {

city string // Unexported

state string

}

Exported: the first word of variable should be Capital. We can not access this variable from the other struct.

type Player struct {

Name string // Exported

Age int

}

## Interface