

0.1 Inline Markup

Paragraphs contain text and may contain inline markup: *emphasis*, **strong emphasis**, `inline literals`,
If a software library or a package is mentioned then use `LIB PackageName`

0.1.1 code format

Everything in a `LIB golang` program which is stored somewhere has a memory address. If one declares a variable and assigns a value to it then one can `reference` the memory address of the stored value using `&` operator(not exactly, because `&` operator returns a `pointer`, its in the next subsection). This operate from the left of the variable(operation on the right has no meaning and will throw an error) for Example: pointers.go Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var x int = 7
7     fmt.Println(x)
8     fmt.Println(&x)
9 }
```

If we run the above program

```
go run pointers.go
7
0xc00001c0a8
```

Line two is the value of the `variable` x (i.e. 7) and line three is the address where it is stored (i.e. 0xc00001c0a8)

$$e^{i\pi} + 1 = 0 \quad (1)$$

0.1.2 Pointers

A pointer is a special kind of datatype which takes a reference as its value . It is defined in the program as `*datatype` (i.e. `*int`, `*uint`, `*char`, `*byte`, etc,) they declare a `pointer` object which points to something(variables, constants, etc.) with a given datatype. In the program given below line 9 and 10 prints the same address

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var x int = 73
7     var xptr *int = &x
8     fmt.Println(x)
9     fmt.Println(&x)
10    fmt.Println(xptr)
11 }

```

Running the above program

```

1 $ go run pointers.go
2 73
3 0xc00001c0a8
4 0xc00001c0a8

```

As one can notice line three and four are the same. In order to get the value of what the pointer is pointing at one can use the `*` operator. This operator operates from the left of the pointer variable. Operating on the right has no meaning if the variable is a pointer and if the variable is any-other datatype then it has the usual meaning of multiplication. Since `*` is doing the opposite of the referencing operator `&` it is called **dereference** operator. The following example illustrates the operation.

! Note

Not to be confused with multiplication operator `*` or the asterisk in front of the pointer data type `*datatype`

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var x int = 73
7     var xptr *int = &x
8     fmt.Println(x)
9     fmt.Println(&x)
10    fmt.Println(xptr)
11    fmt.Println(*xptr)
12 }

```

Result:

```

1 $ go run pointers.go
2 73
3 0xc00001c0a8
4 0xc00001c0a8
5 73

```

! Warning

this is a warning

! tip

this is a tip hint

! Error

this is an error

Sed ut **Table 1** unde **Table 2** omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit

helloMKL

qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur? At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere

My Own Heading

Nam dui ligula, fringilla a, euismo d so dales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus lib ero, pretium at, lob ortis vitae, ul- tricies et, tellus. Donec aliquet, tortor sed accumsan bib endum, erat ligula aliquet magna, vitae ornare o dio metus a mi. Morbi ac orci et nisl hendrerit mollis. Susp endisse ut massa. Cras nec ante. Pellentesque a nulla. Cum so ciis nato que p enatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urn

0.1.3 Tables

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius

modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur? At vero eos et accusamus et iusto odio dignissimos ducimus

Group	One	Two	Three	Four	Sum
Red	1000.00	2000.00	3000.00	4000.00	10000.00
Green	2000.00	3000.00	4000.00	5000.00	14000.00
Blue	3000.00	4000.00	5000.00	6000.00	18000.00
Sum	6000.00	9000.00	12000.00	15000.00	42000.00

Table 1: this is a table

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

his is a Block Quote . Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque pe

Group	One	Two	Three	Sum
Red	1000.00	2000.00	3000.00	10000.00
Green	2000.00	3000.00	4000.00	14000.00
Blue	3000.00	4000.00	5000.00	18000.00
Sum	6000.00	9000.00	12000.00	42000.00

Table 2: this is a table2