

Module 2: Function Types

Topic 1.1: First-class Values

Functions are First-class

- Functions can be treated like other types
 - Variables can be declared with a function type
 - Can be **created dynamically**
 - Can be **passed as arguments** and **returned as values**
 - Can be **stored in data structures**

Variables as Functions

- Declare a variable as a func
- Function is on right-hand side, without ()

```
var funcVar func(int) int
func incFn(x int) int {
    return x + 1
}
func main() {
    funcVar = incFn
    fmt.Print(funcVar(1))
}
```

Functions as Arguments

- Function can be passed to another function as an argument

```
func applyIt(afunct func (int) int,  
            val int) int {  
    return afunct(val)  
}
```

Functions as Arguments

```
func applyIt(afunc func (int) int,  
            val int) int {  
    return afunc(val)  
}  
  
func incFn(x int) int {return x + 1}  
func decFn(x int) int {return x - 1}  
  
func main() {  
    fmt.Println(applyIt(incFn, 2))  
    fmt.Println(applyIt(decFn, 2))  
}
```

Anonymous Functions

- Don't need to name a function

```
func applyIt(afunc func (int) int,  
            val int) int {  
    return afunc(val)  
}
```

```
func main() {  
    v := applyIt(func (x int) int  
                {return x + 1}, 2)  
    fmt.Println(v)  
}
```