-

CommandParserCommand

https://pkg.go.dev/github.com/gogf/gf/v2/os/gcmd@master#Command

```
// Command holds the info about an argument that can handle custom logic.
type Command struct {
        Name          string        // Command name(case-sensitive).
        Usage         string        // A brief line description about its
usage, eg: gf build main.go [OPTION]
        Brief         string        // A brief info that describes what
this command will do.
        Description   string        // A detailed description.
        Arguments     []Argument    // Argument array, configuring how
this command act.
        Func          Function      // Custom function.
        FuncWithValue FuncWithValue // Custom function with output
parameters that can interact with command caller.
        HelpFunc      Function      // Custom help function
        Examples      string        // Usage examples.
        Additional    string        // Additional info about this command,
which will be appended to the end of help info.
        Strict        bool          // Strict parsing options, which means
it returns error if invalid option given.
        Config        string        // Config node name, which also
retrieves the values from config component along with command line.
        parent        *Command      // Parent command for internal usage.
        commands      []*Command    // Sub commands of this command.
}
```

Command3

- Func
- FuncWithValueFunc
- HelpFuncCommand

Func

Func

```
// Function is a custom command callback function that is bound to a
certain argument.
type Function func(ctx context.Context, parser *Parser) (err error)
```

parsererror

Content Menu

```
package main

import (
        "context"

        "github.com/gogf/gf/v2/frame/g"
        "github.com/gogf/gf/v2/net/ghttp"
        "github.com/gogf/gf/v2/os/gcmd"
        "github.com/gogf/gf/v2/os/gctx"
)

var (
        Main = &gcmd.Command{
                Name:        "main",
                Brief:       "start http server",
                Description: "this is the command entry for starting your
http server",
                Func: func(ctx context.Context, parser *gcmd.Parser) (err
error) {
                        s := g.Server()
                        s.BindHandler("/", func(r *ghttp.Request) {
                                r.Response.Write("Hello world")
                        })
                        s.SetPort(8199)
                        s.Run()
                        return
                },
        }
)

func main() {
        Main.Run(gctx.New())
}
```

CommandHelpFuncCommandHelpgcmdh/helpgcmdHelp

go build main.gomain

```
$ ./main -h
USAGE
    main [OPTION]

DESCRIPTION
    this is the command entry for starting your http server
```

CommandCommandCommandAddCommandCommand

```
// AddCommand adds one or more sub-commands to current command.
func (c *Command) AddCommand(commands ...*Command) error
```

```
package main

import (
        "context"
        "fmt"

        "github.com/gogf/gf/v2/os/gcmd"
        "github.com/gogf/gf/v2/os/gctx"
)

var (
        Main = &gcmd.Command{
                Name:        "main",
                Brief:       "start http server",
                Description: "this is the command entry for starting your
process",
        }
        Http = &gcmd.Command{
                Name:        "http",
                Brief:       "start http server",
                Description: "this is the command entry for starting your
http server",
                Func: func(ctx context.Context, parser *gcmd.Parser) (err
error) {
                        fmt.Println("start http server")
                        return
                },
        }
        Grpc = &gcmd.Command{
                Name:        "grpc",
                Brief:       "start grpc server",
                Description: "this is the command entry for starting your
grpc server",
                Func: func(ctx context.Context, parser *gcmd.Parser) (err
error) {
                        fmt.Println("start grpc server")
                        return
                },
        }
)

func main() {
        err := Main.AddCommand(Http, Grpc)
        if err != nil {
                panic(err)
        }
        Main.Run(gctx.New())
}
```

AddCommandhttp/grpchttp/grpcFuncmainFunc

```
$ main
USAGE
    main COMMAND [OPTION]

COMMAND
    http    start http server
    grpc    start grpc server

DESCRIPTION
    this is the command entry for starting your process
```

http

```
$ main http
start http server
```

grpc

```
$ main grpc
start grpc server
```