-Parser

Content Menu

gcmdParse

ParserParser

Parserhttps://pkg.go.dev/github.com/gogf/gf/v2/os/gcmd#Parser

```
func Parse(supportedOptions map[string]bool, option ...ParserOption)
(*Parser, error)
func ParseWithArgs(args []string, supportedOptions map[string]bool, option
...ParserOption) (*Parser, error)
func ParserFromCtx(ctx context.Context) *Parser
func (p *Parser) GetArg(index int, def ...string) *gvar.Var
func (p *Parser) GetArgAll() []string
func (p *Parser) GetOpt(name string, def ...interface{}) *gvar.Var
func (p *Parser) GetOptAll() map[string]string
```

ParserOption

```
// ParserOption manages the parsing options.
type ParserOption struct {
        CaseSensitive bool // Marks options parsing in case-sensitive way.
        Strict        bool // Whether stops parsing and returns error if
invalid option passed.
}
```

```
parser, err := gcmd.Parse(g.MapStrBool{
        "n,name":    true,
        "v,version": true,
        "a,arch":    true,
        "o,os":      true,
        "p,path":    true,
})
```

map,nname-n johnnnamejohn

-f forcefforceforce

```go
func ExampleParse() {
	os.Args = []string{"gf", "build", "main.go", "-o=gf.exe", "-y"}
	p, err := gcmd.Parse(g.MapStrBool{
		"o,output": true,
		"y,yes":    false,
	})
	if err != nil {
		panic(err)
	}
	fmt.Println(p.GetOpt("o"))
	fmt.Println(p.GetOpt("output"))
	fmt.Println(p.GetOpt("y") != nil)
	fmt.Println(p.GetOpt("yes") != nil)
	fmt.Println(p.GetOpt("none") != nil)

	// Output:
	// gf.exe
	// gf.exe
	// true
	// true
	// false
}
```