

gtype(gtype.Int)

```
func NewInt(value ...int) *Int
func (v *Int) Add(delta int) (new int)
func (v *Int) Cas(old, new int) bool
func (v *Int) Clone() *Int
func (v *Int) Set(value int) (old int)
func (v *Int) String() string
func (v *Int) Val() int
```

Content Menu

-
- [JSON/](#)

```
package main

import (
    "github.com/gogf/gf/v2/container/gtype"
    "fmt"
)

func main() {
    // Int
    i := gtype.NewInt()

    //
    fmt.Println(i.Set(10))

    //
    fmt.Println(i.Val())

    // -1
    fmt.Println(i.Add(-1))
}
```

```
0
10
9
```

JSON/

gtypejson/

1Marshal

```

package main

import (
    "encoding/json"
    "fmt"
    "github.com/gogf/gf/v2/container/gtype"
)

func main() {
    type Student struct {
        Id      *gtype.Int
        Name    *gtype.String
        Scores  *gtype.Interface
    }
    s := Student{
        Id:      gtype.NewInt(1),
        Name:    gtype.NewString("john"),
        Scores:  gtype.NewInterface([]int{100, 99, 98}),
    }
    b, _ := json.Marshal(s)
    fmt.Println(string(b))
}

```

```

{"Id":1,"Name":"john","Scores":[100,99,98]}

```

2Unmarshal

```

package main

import (
    "encoding/json"
    "fmt"
    "github.com/gogf/gf/v2/container/gtype"
)

func main() {
    b := []byte(`{"Id":1,"Name":"john","Scores":[100,99,98]}`)
    type Student struct {
        Id      *gtype.Int
        Name    *gtype.String
        Scores  *gtype.Interface
    }
    s := Student{}
    json.Unmarshal(b, &s)
    fmt.Println(s)
}

```

```

{1 john [100,99,98]}

```

