

Be careful with timezones

Especially if you're doing Go in Estonia :)

based on true events

```
package main
```

```
import (  
    "fmt"  
    "time"  
)
```

```
func printTime(input string) {  
    t, _ := time.Parse("15:04", input)  
    t = t.Add(time.Hour)  
    fmt.Println(t.Format("15:04"))  
}
```

```
func main() {  
    printTime("14:35")  
}
```


correct

15 : 35

```
package main
```

```
import (  
    "fmt"  
    "time"  
)
```

```
func printTime(input string) {  
    t, _ := time.Parse("15:04", input)  
    t = t.Add(time.Hour)  
    loc, _ := time.LoadLocation("Europe/Tallinn")  
    fmt.Println(t.In(loc).Format("15:04"))  
}
```

```
func main() {  
    printTime("14:35")  
}
```

```
package main
```

```
import (  
    "fmt"  
    "time"  
)
```

```
func printTime(input string) {  
    t, _ := time.Parse("15:04", input)  
    t = t.Add(time.Hour) GMT+02:00  
    loc, _ := time.LoadLocation("Europe/Tallinn")  
    fmt.Println(t.In(loc).Format("15:04"))  
}
```

```
func main() {  
    printTime("14:35")  
}
```


incorrect

17:14

incorrect

17:14

+01:39 difference 😞


```
package main
```

```
import (  
    "fmt"  
    "time"  
)
```

```
func printTime(input string) {  
    t, _ := time.Parse("15:04", input)  
    t = t.Add(time.Hour)  
    loc, _ := time.LoadLocation("Europe/Tallinn")  
    fmt.Println(t.In(loc).Format("15:04"))  
}
```

```
func main() {  
    printTime("14:35")  
}
```

```
package main
```

correct

```
import (  
    "fmt"  
    "time"  
)
```

```
func printTime(input string) {  
t, _ := time.Parse("15:04", input) t, _ := time.Now()  
t = t.Add(time.Hour)  
loc, _ := time.LoadLocation("Europe/Tallinn")  
fmt.Println(t.In(loc).Format("15:04"))  
}
```

```
func main() {  
    printTime("14:35")  
}
```

Whom to blame?

- `time.Parse?`
- `time.Add?`
- `time.Format?`
- `time.In?`
- `Timezone` itself?

Whom to blame?

- ~~time.Parse?~~
- time.Add?
- time.Format?
- time.In?
- Timezone itself?

Whom to blame?

- ~~time.Parse?~~
- ~~time.Add?~~
- time.Format?
- time.In?
- Timezone itself?

Whom to blame?

- ~~time.Parse?~~
- ~~time.Add?~~
- ~~time.Format?~~
- time.In?
- Timezone itself?

Whom to blame?

- ~~time.Parse?~~
- ~~time.Add?~~
- ~~time.Format?~~
- ~~time.In?~~
- Timezone itself?

```
package main
```

```
import (  
    "fmt"  
    "time"  
)
```

```
func main() {  
    tzName, tzOffset := time.Now().In(time.Local).Zone()  
    fmt.Printf("time.Local is: %s (GMT+%d)\n", tzName, tzOffset/3600)  
  
    t, _ := time.Parse("15:04", "14:35")  
    fmt.Println("T:", t)  
    fmt.Println("T:", t.In(time.Local))  
  
    fmt.Println("Now:", time.Now().In(time.Local))  
}
```


time.Local is: EET (GMT+2)

T: 0000-01-01 14:35:00 +0000 UTC

T: 0000-01-01 16:14:00 +0139 TMT

Now: 2015-02-28 02:41:02.452292128 +0200 EET

time.Local is: EET (GMT+2)

T: 0000-01-01 14:35:00 +0000 UTC

T: 0000-01-01 16:14:00 +0139 TMT

Now: 2015-02-28 02:41:02.452292128 +0200 EET

WTF is that?

time.Local is: EET (GMT+2)

T: 0000-01-01 14:35:00 +0000 UTC

T: 0000-01-01 16:14:00 +0139 TMT

Now: 2015-02-28 02:41:02.452292128 +0200 EET

Brief history of **timezones**

Estonia

timezones in Go

History of timezones

- Before XX century, people used local mean time
- Every city kept it's own meridian and own time
- That become a problem for railways, which had big growth during XIX century
- Humanity had to agree on standard time and meridian

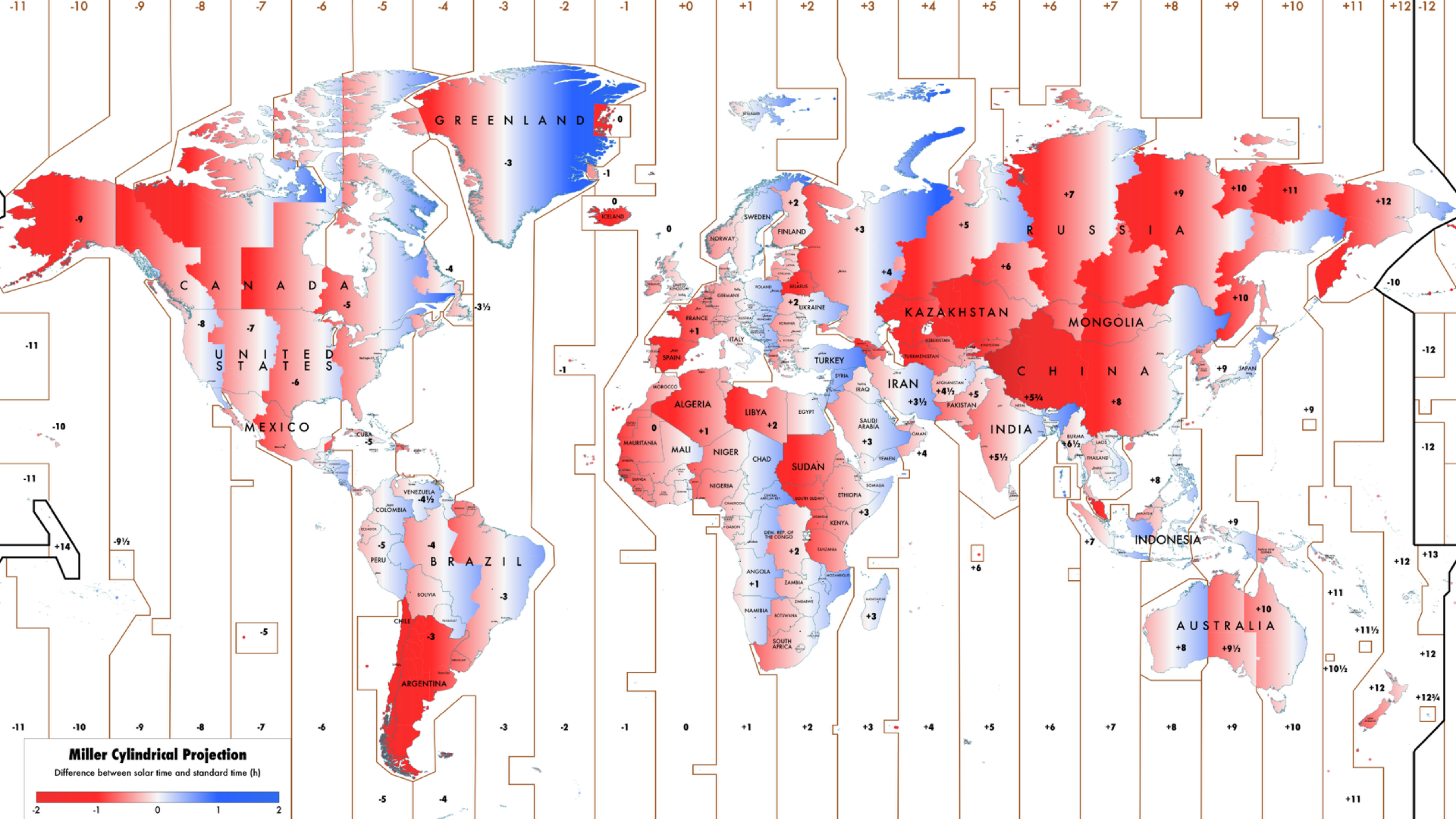


International Meridian Conference

October 1884
Washington, DC



timezones aren't offsets



timezones are weird

timezones are weird

- offsets are often strange
- unusual timezones (like +05:45h in Nepal)
- DST - Daylight Saving Time

timezones change constantly

Time Zone Database

<https://www.iana.org/time-zones>

```
Zone Europe/Madrid -0:14:44 - LMT 1901 Jan 1 0:00s
0:00 Spain WE%sT 1946 Sep 30
1:00 Spain CE%sT 1979
1:00 EU CE%sT
```

```
Zone Africa/Ceuta -0:21:16 - LMT 1901
0:00 - WET 1918 May 6 23:00
0:00 1:00 WEST 1918 Oct 7 23:00
0:00 - WET 1924
0:00 Spain WE%sT 1929
0:00 SpainAfrica WE%sT 1984 Mar 16
1:00 - CET 1986
1:00 EU CE%sT
```

```
Zone Atlantic/Canary -1:01:36 - LMT 1922 Mar # Las Palmas de Gran C.
-1:00 - CANT 1946 Sep 30 1:00 # Canaries T
0:00 - WET 1980 Apr 6 0:00s
0:00 1:00 WEST 1980 Sep 28 1:00u
0:00 EU WE%sT
```

IATA SSIM (1996-09) says the Canaries switch at 2:00u, not 1:00u.

Ignore this for now, as the Canaries are part of the EU.

Brief history of timezones

Estonia

timezones in Go

Estonia

- Population: 1.3 million people
- Area: 45,339 km²
- Small
- But super advanced in IT
- They vote via internet
- They have startup visas
- E-residency program





Estonian timezone

- Location name: **Europe/Tallinn**
- Timezone name: **EET** (Eastern European Time)
- UTC Offset: **+02:00**
- UTC DST Offset: +03:00

So, what's **+0139 TMT?**

Estonian timezones

#	Zone	NAME	GMT	OFF	RULES	FORMAT	[UNTIL]				
Zone	Europe/Tallinn		1:39:00	-	LMT		1880				
			1:39:00	-	TMT		1918	Feb	#	Tallinn	Mean Time
			1:00		C-Eur	CE%sT	1919	Jul			
			1:39:00	-	TMT		1921	May			
			2:00	-	EET		1940	Aug	6		
			3:00	-	MSK		1941	Sep	15		
			1:00		C-Eur	CE%sT	1944	Sep	22		
			3:00		Russia	MSK/MSD	1989	Mar	26	2:00s	
			2:00	1:00	EEST		1989	Sep	24	2:00s	
			2:00		C-Eur	EE%sT	1998	Sep	22		
			2:00		EU	EE%sT	1999	Oct	31	4:00	
			2:00	-	EET		2002	Feb	21		
			2:00		EU	EE%sT					

Estonian timezones

#	Zone	NAME	GMT	OFF	RULES	FORMAT	[UNTIL]			
Zone	Europe/Tallinn		1:39:00	-	LMT		1880			
			1:39:00	-	TMT		1918	Feb	#	Tallinn Mean Time
			1:00		C-Eur	CE%sT	1919	Jul		
			1:39:00	-	TMT		1921	May		
			2:00	-	EET		1940	Aug	6	
			3:00	-	MSK		1941	Sep	15	
			1:00		C-Eur	CE%sT	1944	Sep	22	
			3:00		Russia	MSK/MSD	1989	Mar	26	2:00s
			2:00	1:00	EEST		1989	Sep	24	2:00s
			2:00		C-Eur	EE%sT	1998	Sep	22	
			2:00		EU	EE%sT	1999	Oct	31	4:00
			2:00	-	EET		2002	Feb	21	
			2:00		EU	EE%sT				

Estonian timezones

1:39:00 – LMT	1880
1:39:00 – TMT	1918 Feb
1:00 C–Eur CE%sT	1919 Jul
1:39:00 – TMT	1921 May
2:00 – EET	1940 Aug

Estonian timezones

1:39:00 – LMT	1880	Estonian national awakening
1:39:00 – TMT	1918 Feb	
1:00 C–Eur CE%ST	1919 Jul	
1:39:00 – TMT	1921 May	
2:00 – EET	1940 Aug	

Estonian timezones

1:39:00 – LMT	1880	Estonian national awakening
1:39:00 – TMT	1918 Feb	Estonian Independence (24 Feb)
1:00 C–Eur CE%ST	1919 Jul	
1:39:00 – TMT	1921 May	
2:00 – EET	1940 Aug	

Estonian timezones

1:39:00 – LMT	1880	Estonian national awakening
1:39:00 – TMT	1918 Feb	Estonian Independence (24 Feb)
1:00 C–Eur CE%sT	1919 Jul	The Estonian War of Independence
1:39:00 – TMT	1921 May	
2:00 – EET	1940 Aug	

Estonian timezones

1:39:00 – LMT	1880	Estonian national awakening
1:39:00 – TMT	1918 Feb	Estonian Independence (24 Feb)
1:00 C–Eur CE%sT	1919 Jul	The Estonian War of Independence
		1920 Feb: The Tartu Peace Treaty
1:39:00 – TMT	1921 May	
2:00 – EET	1940 Aug	

Estonian timezones

1:39:00 – LMT	1880	Estonian national awakening
1:39:00 – TMT	1918 Feb	Estonian Independence (24 Feb)
1:00 C–Eur CE%sT	1919 Jul	The Estonian War of Independence
		1920 Feb: The Tartu Peace Treaty
1:39:00 – TMT	1921 May	Accepted to League of Nations
2:00 – EET	1940 Aug	

So, before 1921 the offset for
Europe/Tallinn was +01:39

Brief history of timezones

Estonia

timezones in Go

- Types: Time, Location (each Time has Location)

```
type Time struct {  
    sec int64  
    nsec int32  
    loc *Location  
}
```

```
type Location struct {  
    name string  
    zone []zone  
    tx    []zoneTrans  
    ...  
}
```

```
type zone struct {  
    name    string  
    offset  int  
    isDST   bool  
}
```

- Types: Time, Location (each Time has Location)
- `time.Parse(layout, value string) (Time, error)`
- `time.ParseInLocation(layout, value string, loc *Location) (Time, error)`
- `t.Format(layout string) string`
- `t.In(loc *Location) Time`

- Types: Time, Location

Parses in UTC

- `time.Parse(layout, value string) (Time, error)`
- `time.ParseInLocation(layout, value string, loc *Location) (Time, error)`
- `t.Format(layout string) string`
- `t.In(loc *Location) Time`

- Types: Time, Location

Parses in UTC

- `time.Parse(layout, value string) (Time, error)`

Parses in given timezone (location)

- `time.ParseInLocation(layout, value string, loc *Location) (Time, error)`

- `t.Format(layout string) string`

- `t.In(loc *Location) Time`

- Types: Time, Location

Parses in UTC

- `time.Parse(layout, value string) (Time, error)`

Parses in given timezone (location)

- `time.ParseInLocation(layout, value string, loc *Location) (Time, error)`

Formats in location of t

- `t.Format(layout string) string`
- `t.In(loc *Location) Time`

- Types: Time, Location

Parses in UTC

- `time.Parse(layout, value string) (Time, error)`

Parses in given timezone (location)

- `time.ParseInLocation(layout, value string, loc *Location) (Time, error)`

Formats in location of t

- `t.Format(layout string) string`

Formats in location of loc

- `t.In(loc *Location).Format(layout string) string`

```
package main
```

```
import (  
    "fmt"  
    "time"  
)
```

```
func main() {  
    tzName, tzOffset := time.Now().In(time.Local).Zone()  
    fmt.Printf("time.Local is: %s (GMT+%d)\n", tzName, tzOffset/3600)  
  
    t, _ := time.Parse("15:04", "14:35")  
    fmt.Println("T:", t)  
    fmt.Println("T:", t.In(time.Local))  
  
    fmt.Println("Now:", time.Now().In(time.Local))  
}
```


time.Local is: EET (GMT+2)

T: 0000-01-01 14:35:00 +0000 UTC

T: 0000-01-01 16:14:00 +0139 TMT

Now: 2015-02-28 02:41:02.452292128 +0200 EET

time.Local is: EET (GMT+2)

T: 0000-01-01 14:35:00 +0000 UTC

T: 0000-01-01 16:14:00 +0139 TMT

Now: 2015-02-28 02:41:02.452292128 +0200 EET

Zero date: January 1st, 0000 year

- Type **time.Time** holds not only time, but also a **date**

- Type **time.Time** holds not only time, but also a **date**
- If **Parse()** has only time info, the date is **01/01/00**

- Type **time.Time** holds not only time, but also a **date**
- If **Parse()** has only time info, the date is **01/01/00**
- Timezone rules applies for this year

- Type **time.Time** holds not only time, but also a **date**
- If **Parse()** has only time info, the date is **01/01/00**
- Timezone rules applies for this year
- For **Europe/Tallinn**, the offset before 1921 is **+1:39**

- Type **time.Time** holds not only time, but also a **date**
- If **Parse()** has only time info, the date is **01/01/00**
- Timezone rules applies for this year
- For **Europe/Tallinn**, the offset before 1921 is **+1:39**
- So, the code is **technically correct**

- Type **time.Time** holds not only time, but also a **date**
- If **Parse()** has only time info, the date is **01/01/00**
- Timezone rules applies for this year
- For **Europe/Tallinn**, the offset before 1921 is **+1:39**
- So, the code is **technically correct**
- But result is **surprising**

time.Local is: EET (GMT+2)

T: 0000-01-01 14:35:00 +0000 UTC

T: 0000-01-01 16:14:00 +0139 TMT

Now: 2015-02-28 02:41:02.452292128 +0200 EET

Thanks to Ian Lance Taylor for helping to solve this at golang-nuts

The screenshot shows a Google Groups forum page for the 'golang-nuts' group. The thread title is 'Weird timezone (+139 TMT). Is it Go-specific issue?'. The post is by Ivan Daniluk. It contains Go code for parsing a time string and calculating a timezone offset. A reply by Ian Lance Taylor explains the issue with the 'year zero' in the zoneinfo data for Tallinn.

Post by me (Ivan Daniluk):

Hi all,

I'm in Tallinn (Estonia) right now (EET) and the next code produces strange result for me, both on Linux instance, and on local MacOS X:

```
package main

import (
    "fmt"
    "time"
)

func main() {
    tzName, tzOffset := time.Now().In(time.Local).Zone()
    fmt.Printf("time.Local is: %s (GMT+%d)\n", tzName, tzOffset/3600)

    t, _ := time.Parse("15:04", "14:35")
    fmt.Println("T:", t)
```

Reply by Ian Lance Taylor:

Other recipients: ivan.d...@gmail.com

Note that it might be more informative if you printed out tzOffset before dividing by 3600, since that will drop out the odd minutes offset.

Based on looking at the zoneinfo data, I think this is correct. Note that your parsed time has no date. It is in the year zero. The zoneinfo data for Tallinn says

Zone	Europe/Tallinn	1:39:00	-	LMT	1880
	1:39:00	-	TMT	1918 Feb # Tallinn Mean Time	
	1:00	C-Eur	CE%sT	1919 Jul	
	1:39:00	-	TMT	1921 May	
	2:00	-	EET	1940 Aug 6	
	3:00	-	MSK	1941 Sep 15	
	1:00	C-Eur	CE%sT	1944 Sep 22	

Conclusion

When you work with time...

14:35

When you work with time... always think about timezones...

14:35

When you work with time... always think about timezones...

09:35 10:35 11:35 12:35 13:35 14:35 15:35 16:35 17:35 18:35 19:35 20:35

When you work with time... always think about timezones...
don't forget DST... and historic context of timezones

08:35	09:35	10:35	13:35	14:35				16:35	17:35	18:35	19:35	20:35	21:35
09:35	10:35	11:35	12:35	13:35	14:35	15:35	16:35	17:35	18:35	19:35	20:35		
COT	GYT	SRT	BRST	CVT	UTC	CET	EET	EEST	AMT	TJT	BIOT		
CST	PYT	UYT	FNT	EGT			CEST	IDT	AZT	TMT	BST		
ECT	VET	ADT	GST	AZOT			IST	MSK	SCT	UZT	BTT		
EST	AMT	ART	PMDT					TRT	MUT	MVT	KGT		
PET	AST	BRT								PKT	VOST		
ACT	BOT	CLST											

2017 20:05

19xx 16:21

1884 12:11 16:14 19:55 20:37

09:41 Local Mean Time for many timezones (**UTC is safe**)

0000

PS. *Always* use UTC on the server

PPS. Go handles timezones
extremely well

Thank you