

on the toilet

episode #8

Do you remember January 2nd, 2006?

I don't, but golang certainly cares about it. Let's say you want to convert a string to `Time` struct. You want to figure out how, you open your terminal and type `$ go doc time parse`, and this is what you get:

```
func Parse(layout, value string) (Time, error)
    Parse parses a formatted string and returns the time value it represents.
    The layout defines the format by showing how the reference time, defined to
    be
    Mon Jan 2 15:04:05 -0700 MST 2006
    would be interpreted if it were the value;
    ...
```

Okay, cool. So this means that you can create your own layout:

```
layout := "2||1||2006:::>>>15:04:05<<<:::-07"
t, _ := time.Parse(layout, "12||10||1992:::>>>17:00:00<<<:::-12")
fmt.Println(t) // 1992-10-12 17:00:00 -1200 -1200
```

Or use predefined format :

```
t1, _ := time.Parse(time.RFC3339, "1992-10-12T17:00:00-12:00")
fmt.Println(t1) // 1992-10-12 17:00:00 -1200 -1200
fmt.Println(t1.Equal(t)) // true
```

Where this magic constant **Mon Jan 2 15:04:05 -0700 MST 2006** comes from?
It's just 1234567 put into a date: **01/02 03:04:05PM '06 -0700**.

It's not like nothing happened though. Here are couple of facts from 02.01.2006:

- *"Thirteen U.S. coal miners are trapped after an underground explosion in Upshur County, West Virginia."*
- *"Ugandan presidential candidate Kizza Besigye is released from prison. Besigye was arrested on November 14 on treason and rape charges."*
- *"The leader of the Maoist guerrillas in Nepal issued a statement that his group, the People's Liberation Army, will resume its war with the monarchy after a four-month truce."*

