



Ruby 101

Intermediate Ruby:Time

Intermediate Ruby:

The Time class:

Year

Month

Day

Hour

Min

Sec

Functions:

add, subtract

print

compare

Day of week



Creating a time object:

`Time.new` *year, month, day, hour, min, sec*

`Time.gm` *year, month, day, hour, min, sec*

`Time.local` *year, month, day, hour, min, sec*

`Time.now`

Time represents
number of seconds since Unix Epoch Jan 1, 1970 00:00

`Time.at` *seconds_since_epoch*



Creating a time object:

```
Time.gm 2015, "Feb", 22, 19, 23, 14
```

Parse Time in the Log Parser program

```
def set_properties match_data
  ...
  @time_stamp = Time.gm match_data[6],
    match_data[5], match_data[4], match_data[7],
    match_data[8], match_data[9]
  ...
end
```



Displaying Time:

`.to_i`

-> seconds since epoch

`.to_f`

-> seconds since epoch

`.to_s`

-> "YYYY-MM-DD HH:MM:SS" +
" +/-NNNN" OR " UTC"



Formating Time Display

`.strftime format_string`

-> format according to string

String options:

- %Y -> Year at least 4 digits
- %m -> Month number, 0-padded
- %B -> Month name
- %b -> Abbreviated month name
- %d -> Day of month, 0-padded
- %H -> Hour/24, 0-padded
- %I -> Hour/12, 0-padded
- %p / %P -> AM/PM or am/pm
- %M -> Minutes, 0-Padded
- %S -> Seconds, 0-padded

cf <http://www.ruby-doc.org/core-2.2.0/Time.html#method-i-strftime>

View

```
.strftime "%m-%d %H:%M:%S"  
DD-MM HH:MM:SS -> 14 columns + 1 for space + 1 for next column  
def update_log_file  
  ...  
  total_columns = $stdout.winsize[1] - 44  
  ...  
  row = "\e[K" + entry.time_stamp.strftime("%m-%d %H:%M:%S") +  
        "\e[16G" + entry.ip_address +  
        "\e[#{17 + 16}G" + entry.request.slice(0,  
                                                text_column_size) +  
  ...
```

nb ... we need to add 16 to all the other cursor placements in the row.