

# 64bit PHPの罨

hanawa (a.k.a. id:hnw)  
y at hnw dot jp

第51回PHP勉強会@関東 発表資料

# 自己紹介

- id:hnw
  - 浮動小数点数の人
  - phpallの人
- 好物：楽しいバグ&楽しい仕様
- 趣味：初心者なのに釣れちゃった！

# 釣果

The screenshot shows a web browser window displaying a Hatena Diary entry. The browser's address bar shows the URL `http://d.hatena.ne.jp/hnw/20090227`. The page title is "PHPのsort関数は相当おかしい - hnwの日記". The diary entry is dated "2009年2月27日 (金)" and has the title "PHPのsort関数は相当おかしい". It includes a "追記" (update) from 2009/02/28 15:35 explaining that the original text was misleading. The main text discusses "オヤジギャグ" (old man jokes) and introduces examples of "PHPのsort関数" (PHP's sort function) behaving unexpectedly. A section titled "sort関数の紹介" (Introduction to the sort function) is visible at the bottom, with the first line being "sort — 配列をソートする". The browser's taskbar at the bottom shows various icons, including YSlow, and the system tray displays the date and time as "13年10月15日 火曜日".

PHPのsort関数は相当おかしい - hnwの日記

http://d.hatena.ne.jp/hnw/20090227

Hatena::Diary

日記 検索

最新の日記 記事一覧 日記を書く 管理 ログアウト ヘルプ

hnwの日記

<前の日 | 次の日> [プロフィール]

2009年2月27日 (金) PHPのsort関数は相当おかしい 編集 235 users 38 stars

追記 (2009/02/28 15:35) : ソートする配列の要素が数値または数値形式の文字列のみの場合は、<、==、>が推移律を満たすので、この記事のような矛盾は起こりません。念のため。

オヤジギャグがこらえられなくなったら立派なオヤジだと思います。それはさておき、今日はPHPのsort関数が不思議な挙動をする例を紹介します。

sort関数の紹介

sort — 配列をソートする

説明

YSlow 235 54 14 8 hnw

64bit 環境

使ってまますか



開発環境も

64bit 環境？

# 64bit 環境とは

- CPUが64bit整数レジスタを持つ
- アドレッシングが64bit
- 1プロセスが使えるメモリ空間が広がる
  - 4GB (32bit環境) → 16777216TB

# 64bit 化に必要なもの

- 64bit CPU
- 64bit OS
- 64bit 対応コンパイラ
- 利用ソフトの 64bit 環境での実績



# 現状

- サーバ環境
  - 64bit化の障害は殆どない
  - 4GB以上使いたいニーズ
- ローカルの開発環境（PCなど）
  - まだまだ32bitだらけ？



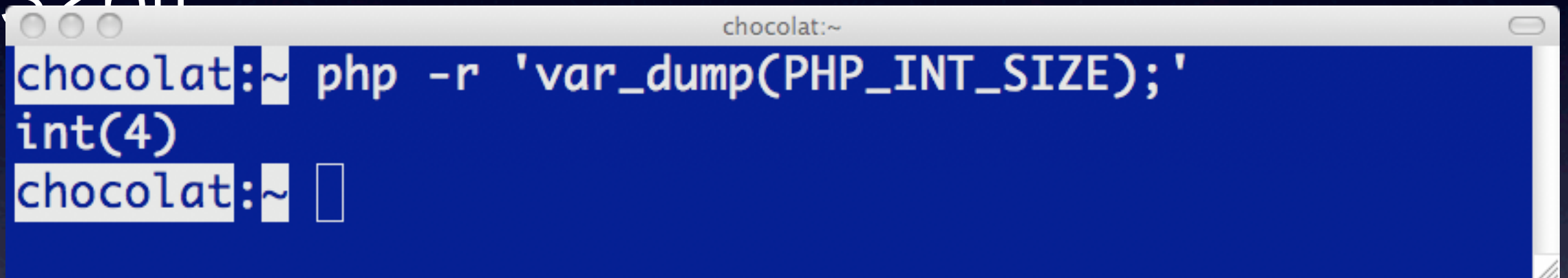
過渡期？

# 一方PHPは

- Cコンパイラ次第
  - PHPのinteger型はCのlong型
    - Unix系64bit環境では64bit
- integerが64bitの環境ではunix timeが64bitに拡張 (PHP5.2.6以降)

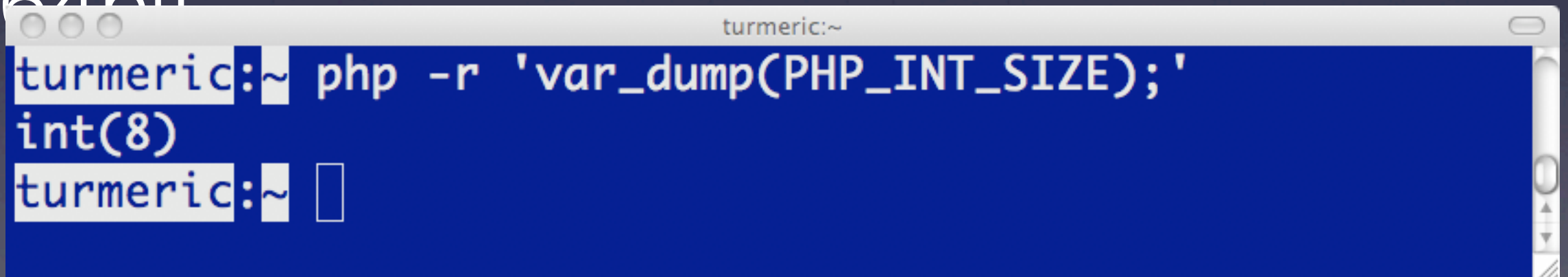
# 整数のサイズの確認方法

- 32bit



```
chocolat:~  
chocolat:~ php -r 'var_dump(PHP_INT_SIZE);'  
int(4)  
chocolat:~
```

- 64bit

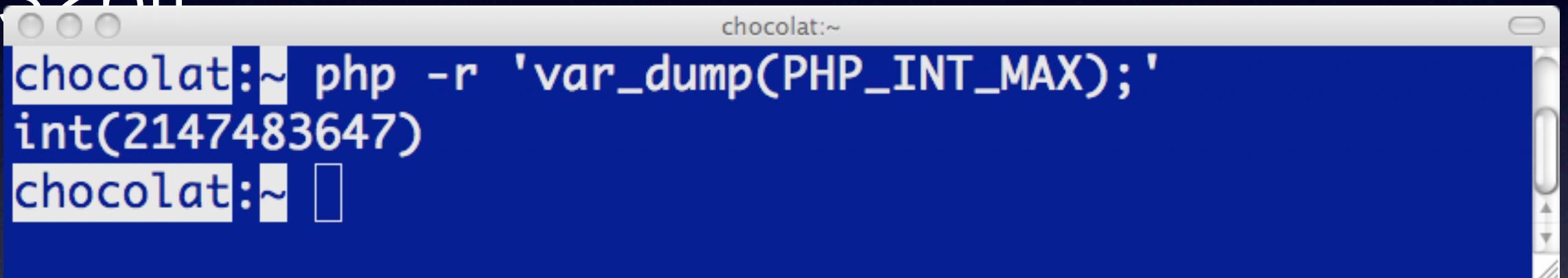


```
turmeric:~  
turmeric:~ php -r 'var_dump(PHP_INT_SIZE);'  
int(8)  
turmeric:~
```



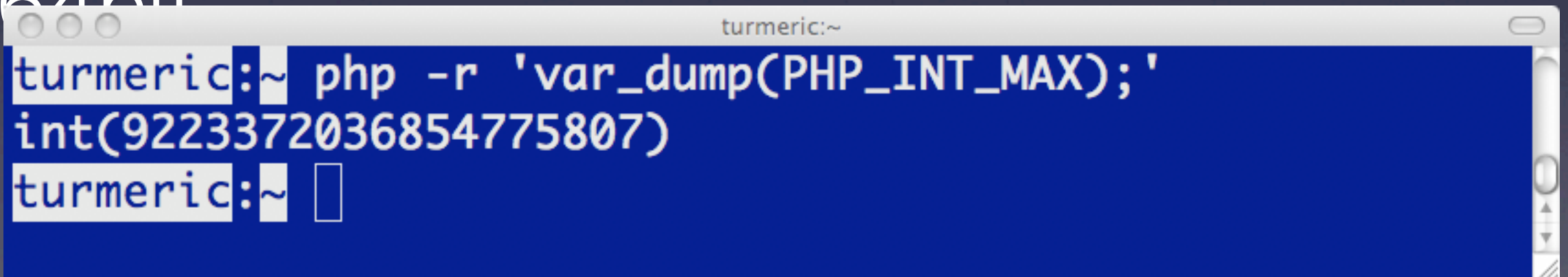
# 最大の整数の確認方法

- 32bit



```
chocolat:~  
chocolat:~ php -r 'var_dump(PHP_INT_MAX);'  
int(2147483647)  
chocolat:~
```

- 64bit



```
turmeric:~  
turmeric:~ php -r 'var_dump(PHP_INT_MAX);'  
int(9223372036854775807)  
turmeric:~
```

# 本題

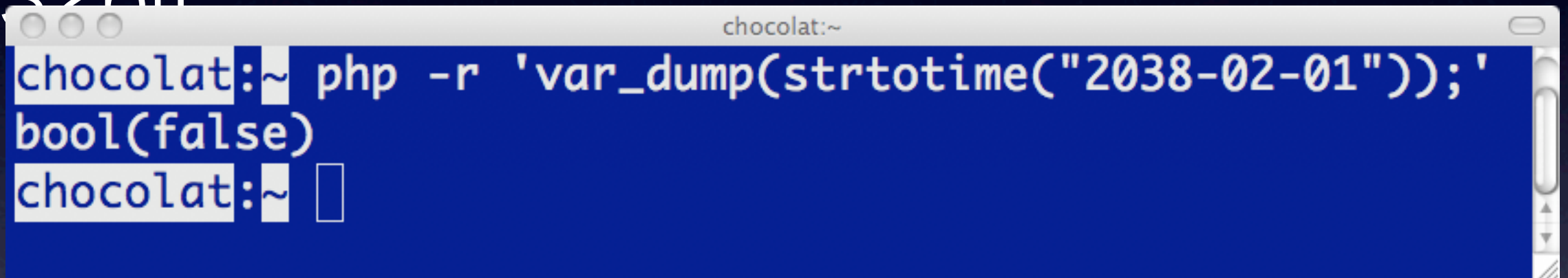
# unix timeが64bitな件

- まあ嬉しいかも
- 罨もあるよ



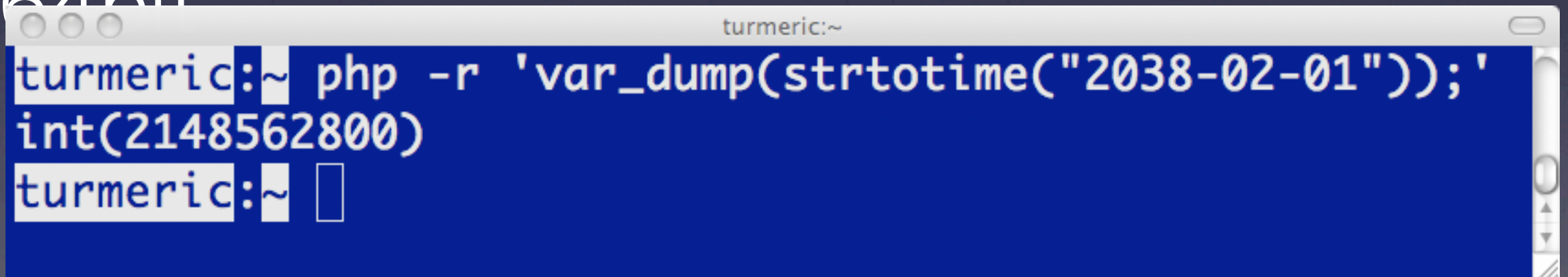
# サヨナラ2038年問題

- 32bit



```
chocolat:~  
chocolat:~ php -r 'var_dump(strtotime("2038-02-01"));'  
bool(false)  
chocolat:~
```

- 64bit



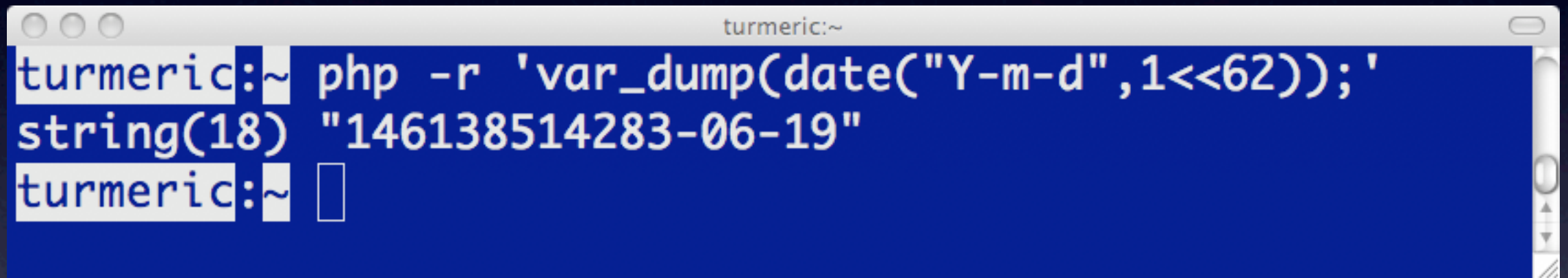
```
turmeric:~  
turmeric:~ php -r 'var_dump(strtotime("2038-02-01"));'  
int(2148562800)  
turmeric:~
```

# サヨナラ2038年問題

- 32bit : 1901年から2038年まで
- 64bit : 1年より前も9999年より後もOK
  - やった！悩みが減ったね！
  - 実は別の悩みが増えてたり

# SQLで使えない日付

- 64bit



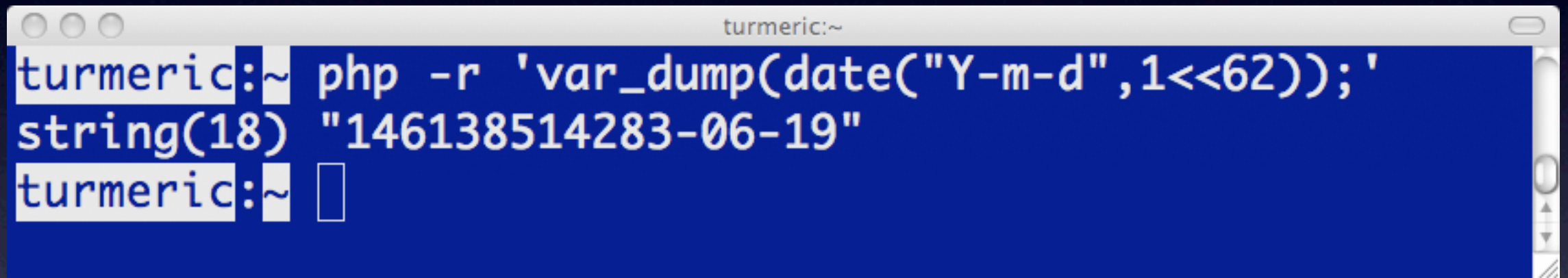
```
turmeric:~  
turmeric:~ php -r 'var_dump(date("Y-m-d",1<<62));'  
string(18) "146138514283-06-19"  
turmeric:~
```

- 西暦1400億年とか言われた！



# SQLで使えない日付

- 64bit



```
turmeric:~  
turmeric:~ php -r 'var_dump(date("Y-m-d",1<<62));'  
string(18) "146138514283-06-19"  
turmeric:~
```

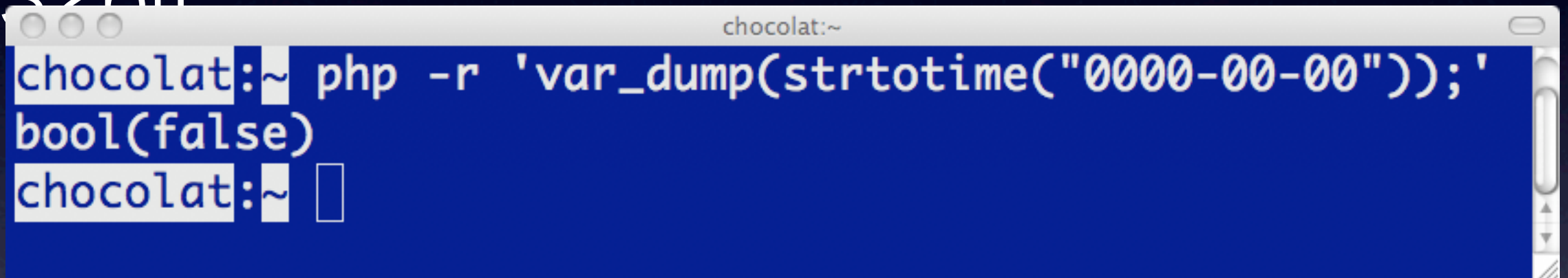
- 西暦1400億年とか言われた！
  - 参考：太陽の寿命=50億年

# SQLで使えない日付

- 64bitだと凄い日付も作れちゃう
  - SQL標準では0001年から9999年まで
- 日付の正当性チェックは  
strptimeの成功/失敗とは別にやるう

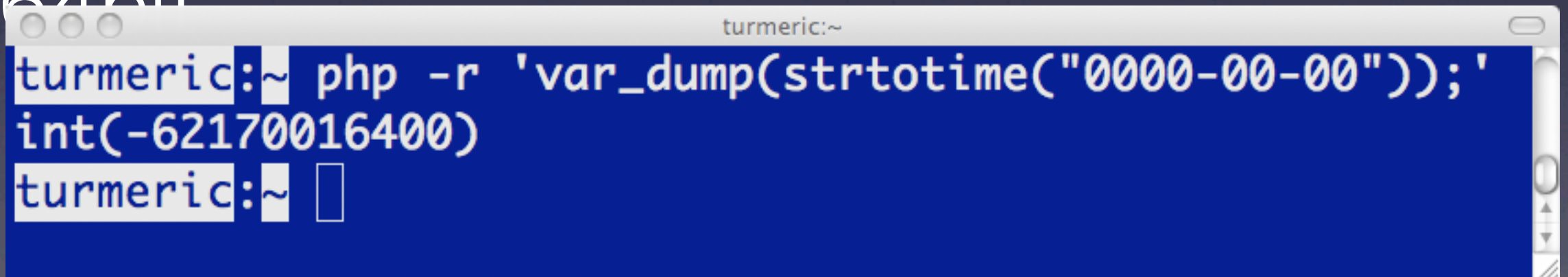
# 0000-00-00問題

- 32bit



```
chocolat:~  
chocolat:~ php -r 'var_dump(strtotime("0000-00-00"));'  
bool(false)  
chocolat:~
```

- 64bit



```
turmeric:~  
turmeric:~ php -r 'var_dump(strtotime("0000-00-00"));'  
int(-62170016400)  
turmeric:~
```



# 0000-00-00問題

- PHPは紀元前1年の-1月-1日とみなす
- そもそもMySQLの悪い習慣
  - 本来的的にはNULLの意味だよね…

# 0000-00-00問題

- PHPは紀元前1年の-1月-1日とみなす
- そもそもMySQLの悪い習慣
  - 本来的にはNULLの意味だよね…
- 0000-00-00撲滅運動を推進したい！

# integerでかすぎ

- 64bit環境のPHP
  - integerもfloatも64bit
- integerの計算でオーバーフローしたら勝手にfloatになる仕様
- アレな気がするけど、実害は思いつかない



# まとめ

- unix timeの64bit化は罨
  - 気をつけよう
- integer が64bitになってるのも少々罨
  - あんまり気をつける点は無いかも



ご清聴  
ありがとうございます  
ございました