

# 超簡単! 英語でバグレポート

田中 哲

2015-11-08

産業技術総合研究所

これだけで大丈夫

Is this intentional?

これは意図的ですか？

# バグレポートの例 (1)

Is this intentional?

```
% ruby -ve 'p [-0.0, 0.0].max'
```

```
ruby 2.3.0dev (2015-11-03 trunk 52430) [x86_64-linux]
```

```
-0.0
```

# バグレポートの例 (2)

Is this intentional?

```
% ruby -ve 'p (-1.0)**Float::INFINITY'
```

```
ruby 2.3.0dev (2015-11-03 trunk 52430) [x86_64-linux]
```

```
1.0
```

# バグレポートのフォーマット

- まえがき
  - 省略可能: 挨拶  
Hi. など。
  - これは意図的ですか?  
Is this intentional?
- 必須: 再現手順
- 推奨: 期待する挙動とその理由  
I expect ...

# バグレポートの例 (1) 期待つき

Hi.

Is this intentional?

```
% ruby -ve 'p [-0.0, 0.0].max'
```

```
ruby 2.3.0dev (2015-10-05 trunk 52033) [x86_64-linux]
```

```
-0.0
```

I expect 0.0.

# バグレポートの例 (2) 期待つき

Is this intentional?

```
% ruby -ve 'p (-1.0)**Float::INFINITY'
```

```
ruby 2.3.0dev (2015-10-05 trunk 52033) [x86_64-linux]
```

```
1.0
```

I expect NaN.

# バグレポートの理屈

- 再現手順
- まえがき
- 期待する挙動

# 再現手順

- 再現手順は最重要  
開発者は手元で再現させて、自分なりの判断を行う
- コピペで再現できるのが最高
- 環境を記述する  
環境に依存する挙動かもしれない  
ruby なら、まず `ruby -v` の結果を書く  
情報が足りなくて尋ねられたら答える  
可能なら、関係ありそうなことは最初から書いておく  
Linux distribution なら `lsb_release -a` など  
問題によってはロケールとか、タイムゾーンとか

# まえがき

実際のところ、たいして重要でない

- 挨拶  
べつになくてもよい  
書きたい気分なら書く
- Is this intentional?
  - 疑問形を使う
    - 実際のところ、バグではないかもしれない
    - 報告者はバグでなかったときに簡単に引き下がる  
「なるほど。理解しました。」
  - 疑問形を使わない場合
    - バグであることがはっきりしている場合
    - Ruby で SEGV するとか

# 期待する挙動

- ドキュメントに書いてある挙動を期待するなら、ドキュメントを引用する
  - 実装とドキュメントの食い違いはどちらかのバグ
- 個人的に期待する挙動を書くかどうか
  - 書いた方が開発者にはわかりやすい  
(開発者が同意するとは限らないけれど)
  - どういう修正になっても気にしないなら、書かないという手もある
  - 報告者にとって意外な修正になると楽しい  
(意外さは笑える)

子供の笑いの中で典型的だと思うのは、第一に何かしら意外な、しかしそれほど恐ろしくはない重大ではない事がらが突発して、それに対する軽い驚愕が消え去ろうとする時に起こるものである。

寺田寅彦, 笑い, 1947

# 返事の例 (1)

> % ruby -ve 'p [-0.0, 0.0].max'

> ruby 2.3.0dev (2015-10-05 trunk 52033) [x86\_64-linux]

> -0.0

> I expect 0.0.

Array#max uses <=> method for comparison.

Unfortunately, it is impossible to distinguish 0.0 and -0.0 using Float#<=>.

## 返事の例 (2)

```
> % ruby -ve 'p (-1.0)**Float::INFINITY'  
> ruby 2.3.0dev (2015-10-05 trunk 52033) [x86_64-linux]  
> 1.0  
> I expect NaN.
```

This behavior is inherited from `pow()` function in C.  
So, it is not a Ruby's problem.

Actually it returns NaN on NetBSD.

# 環境依存問題と開発者の思考

```
> % ruby -ve 'p (-1.0)**Float::INFINITY'  
> ruby 2.3.0dev (2015-10-05 trunk 52033) [x86_64-linux]  
> 1.0  
> I expect NaN.
```

glibc なら pow() が 1.0 を返す

おそらく glibc を使っている

This behavior is inherited from pow() function in C.  
So, it is not a Ruby's problem.

Ruby の実装は  
こうなっている

Actually it returns NaN on NetBSD.

NetBSD の pow() は NaN を返す

# 修正されないときにどうするか

- 気にしない
- 反論する

# 修正されなくても気にしない

- 修正されないのが正しいと納得できれば
- 自分にとって重要な問題でなければ
- 自分が問題を避けられるのであれば
- 修正しない結果の責任は開発元がとるもの

# 修正されないのが正しいと納得

- > Array#max uses <=> method for comparison.
- > Unfortunately, it is impossible to distinguish 0.0
- > and -0.0 using Float#<=>.

I understand. Thank you.

# 反論する

- 自分にとって重要であれば反論する  
手間はかかる
- 根拠を集める
- 方針を決める
- 英語でも日本語でも論理的な文章を記述する必要がある

ここからは bug report というよりは feature request の範疇

# 反論の準備の例: 根拠を集める

- 実装を調べる
  - Ruby の数学関数はたいてい自前で引数の範囲を検査しているのに Float#\*\* ではしていない → 一貫性の不備、環境依存性の露出
- 規格を調べる
  - C99 では 1.0 を返すと書いてある (F.9.4.4, Annex F は normative) → 嬉しくない
  - SUSv4 では pow() が 1.0 を返すと書いてある → 嬉しくない  
ただ MX マークがついているので optional か
- 他の言語を調べる
  - Perl: Rubyと同じく環境依存 (GNU/Linux なら 1.0, NetBSD なら NaN)
  - Python: 1.0 (NetBSD でも 1.0 になる)
  - Java: NaN → 嬉しい
  - ECMAScript: NaN → 嬉しい
- さまざまな環境を調べる
  - GNU/Linux: 1.0      GNU/Hurd: 1.0      AIX: 1.0      NetBSD: NaN
  - FreeBSD: 1.0      Haiku: 1.0                      SunOS: NaN
  - OpenBSD: 1.0      Darwin: 1.0                      Minix: NaN

# 規格についてさらに調べる

- Rationale for International Standard ---  
Programming Language --- C  
「The result of  $\text{pow}(-2, \infty)$  is  $+\infty$ , because all large positive floating-point values are even integers.」  
<http://www.open-std.org/jtc1/sc22/wg14/www/C99RationaleV5.10.pdf>  
無限大を偶数とみなすってということ？
- 検索すると C99 committee は失敗したという見解がいくつかみつける

# 反論の準備の例: 方針を決める

- 自前で引数の範囲を検査をしていなくて環境依存になっているのはよくないと主張可能だろう
- NaN でのほうが良い気がする
  - 1.0 になる環境は多いが、それは C99 がおかしい
  - Java や ECMAScript (JavaScript) はまとも
- 他の値についても検討する
  - ECMAScript の仕様には 23 項目の例が書いてある

# まとめ

- バグレポートには再現手順を書きましょう
- 英語は最小限で済みます  
Is this intentional?  
これは意図的ですか？
- 気楽にバグレポートしましょう
- 反論したい場合は手間がかかります