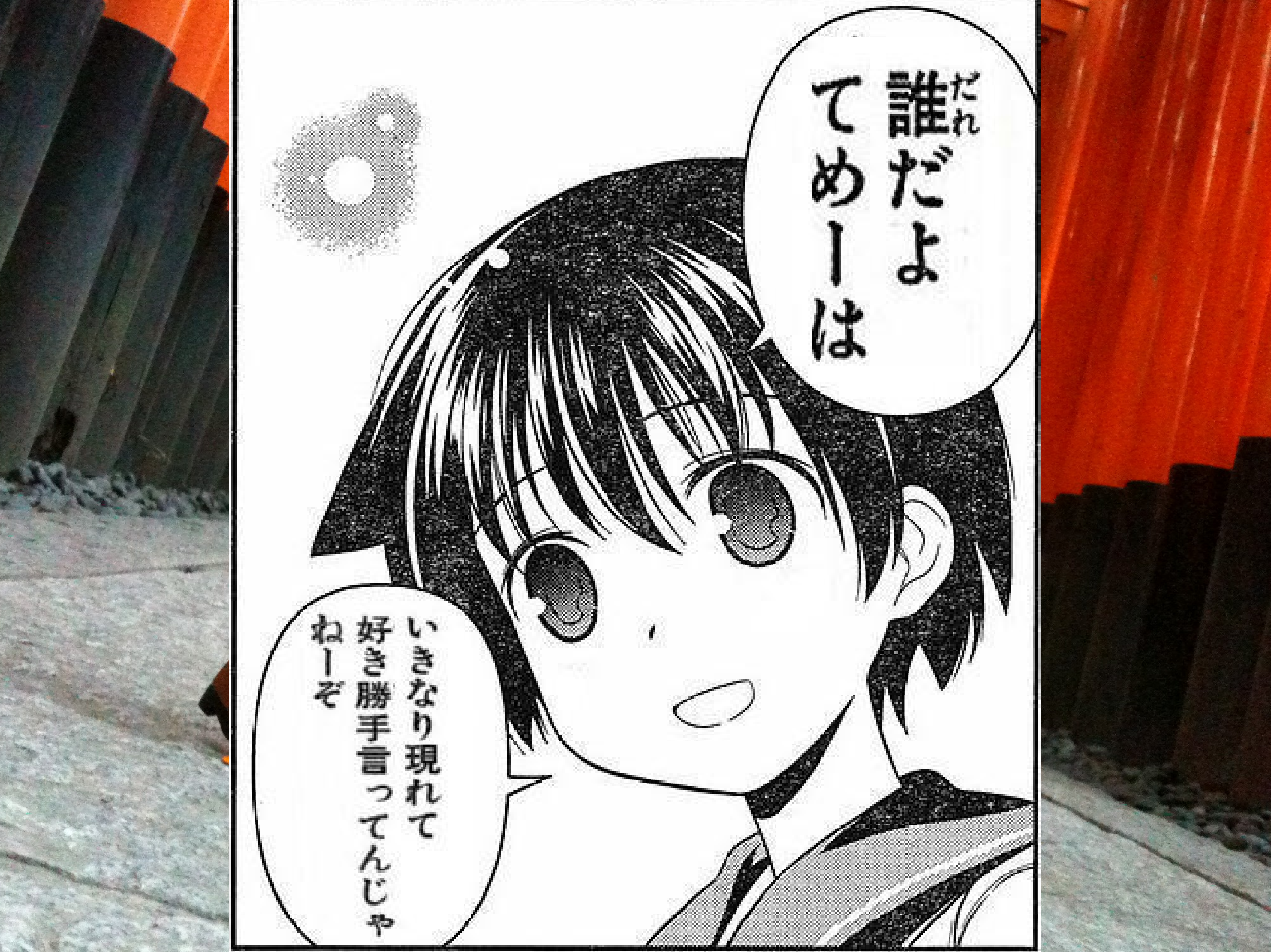
valgrindは実行時メモリチェッ(略

ψ（プサイ）

@tikal

ledyba.org


誰（だれ）だよてめーは
いきなり現れて好き勝手言ってんじゃねーぞ

人間界での所属

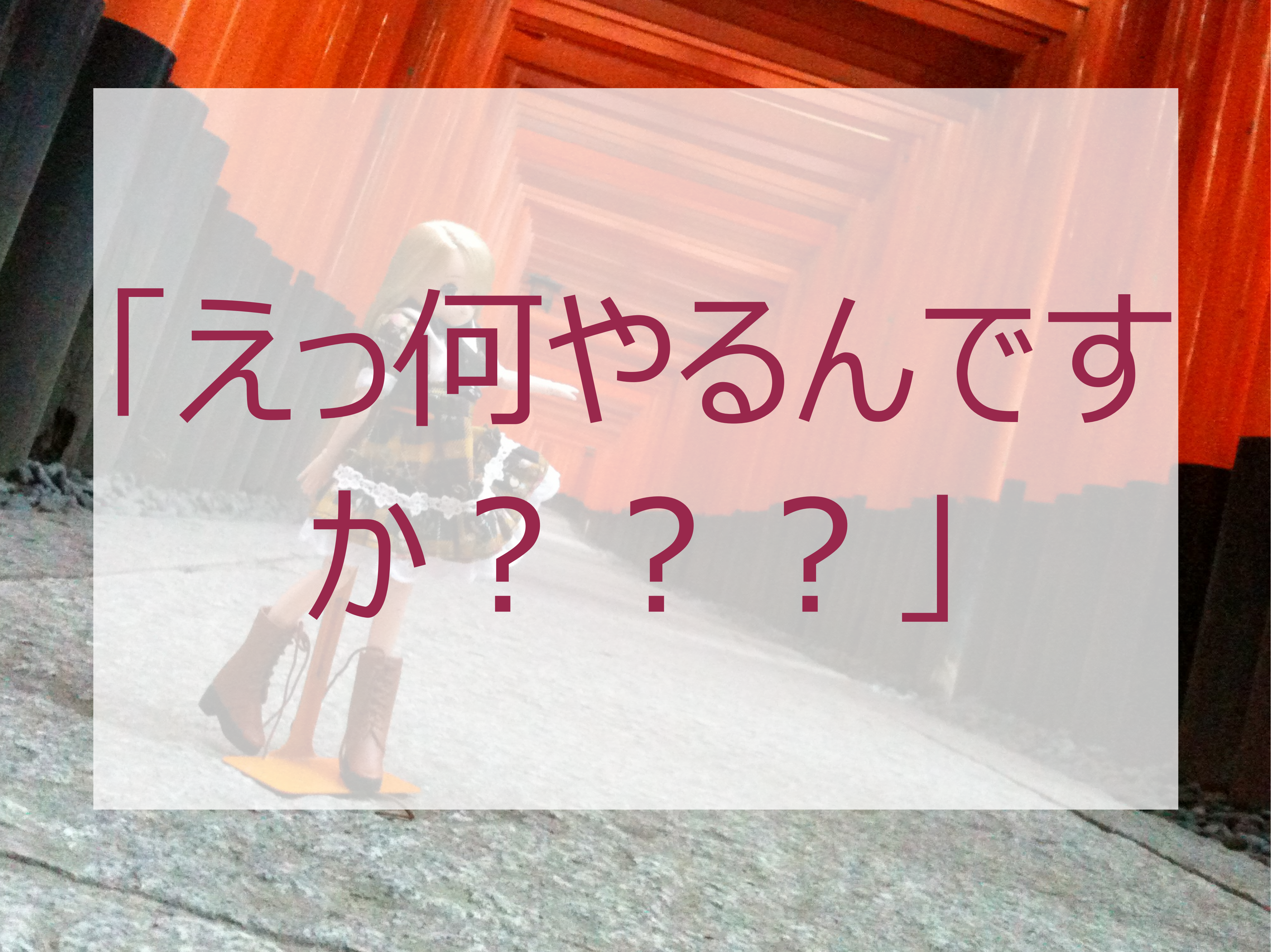
東大
農学部
獣医学科

↓転学部↓

東大

理学部

地球惑星物理学科



「えっ何やるんですか？？？」

天気

地震

FORTRAN77

前学科長が

元参議院議員
東大教授
ロバート・ゲラー
東京大学大学院教授
ロバート・ゲラー

ツイ廃

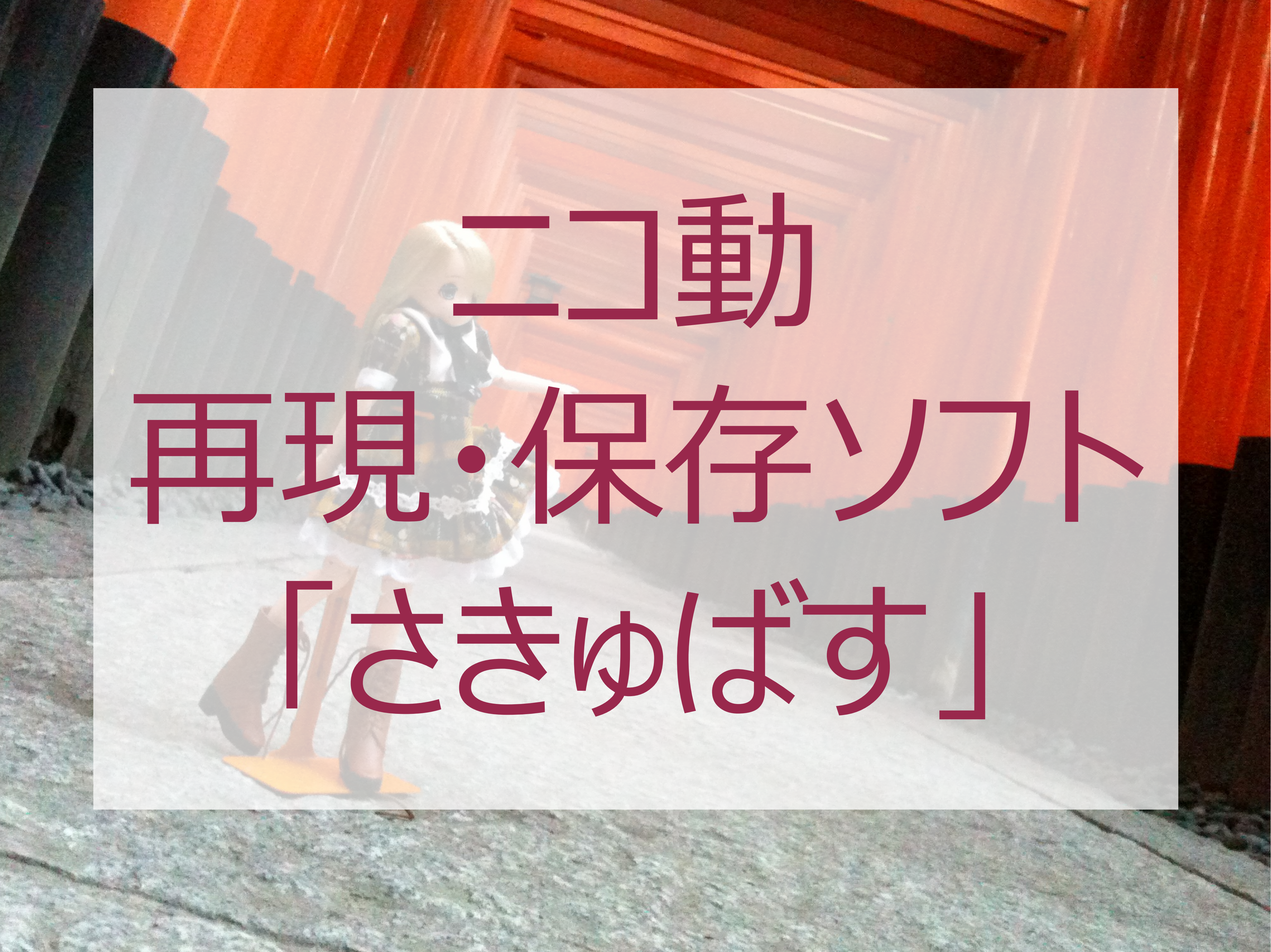
@rjgeller

つらい

東大ニコ研

(略称：UNKO)

1 of
創設メンバー



ニコ動
再現・保存ソフト
「さきゅばす」

未踏2012
スーパー(笑)
クリエイター(笑)

言語とか
作ってます♡

http://donut-lang.org

github/ledyba/Akomachi

github/Saccubus/Saccubus

というのは

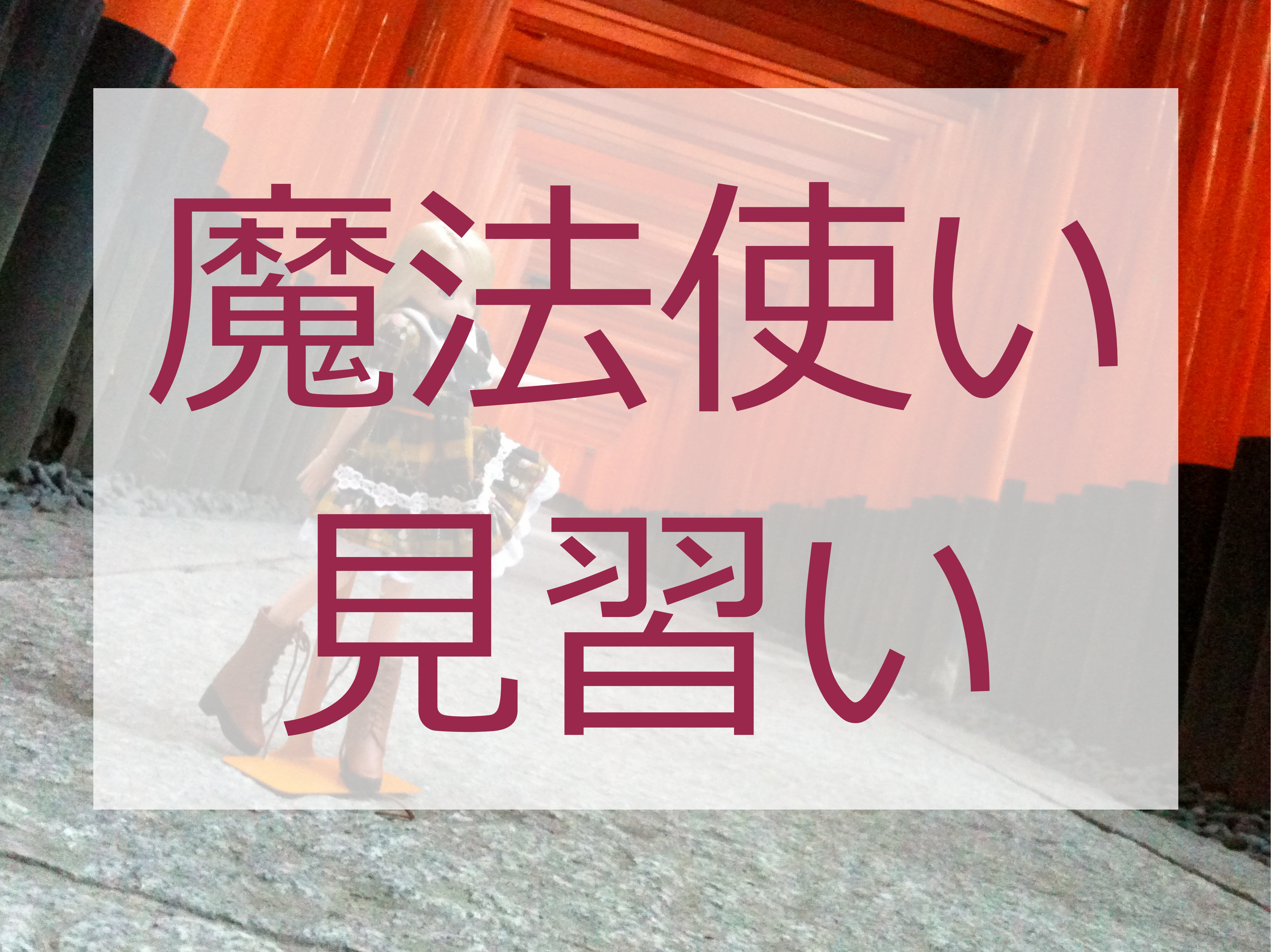
全部

嘘

本当は

魔法界から来た

10万
10歳の



魔法使い
見習い

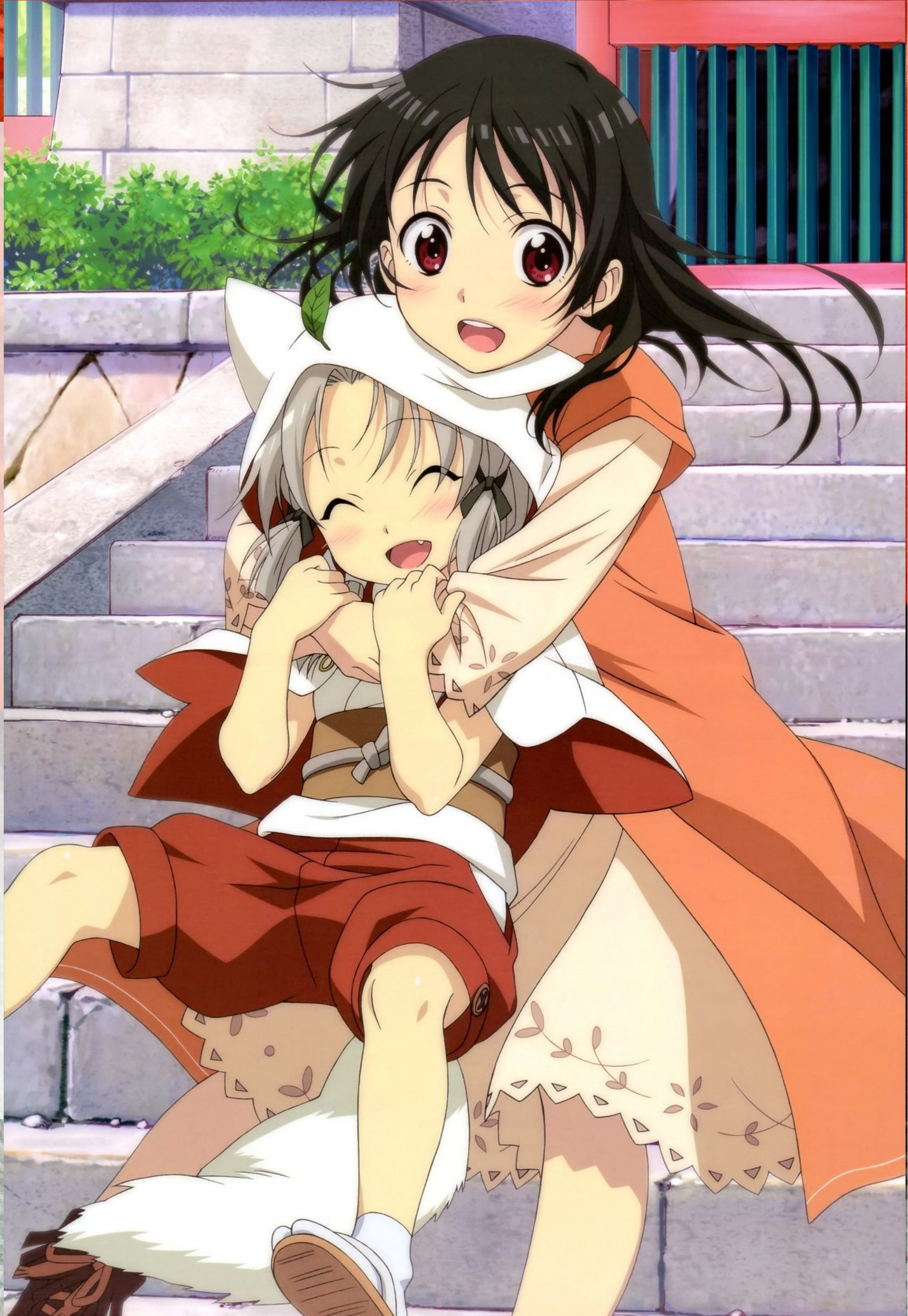
ほ…っ

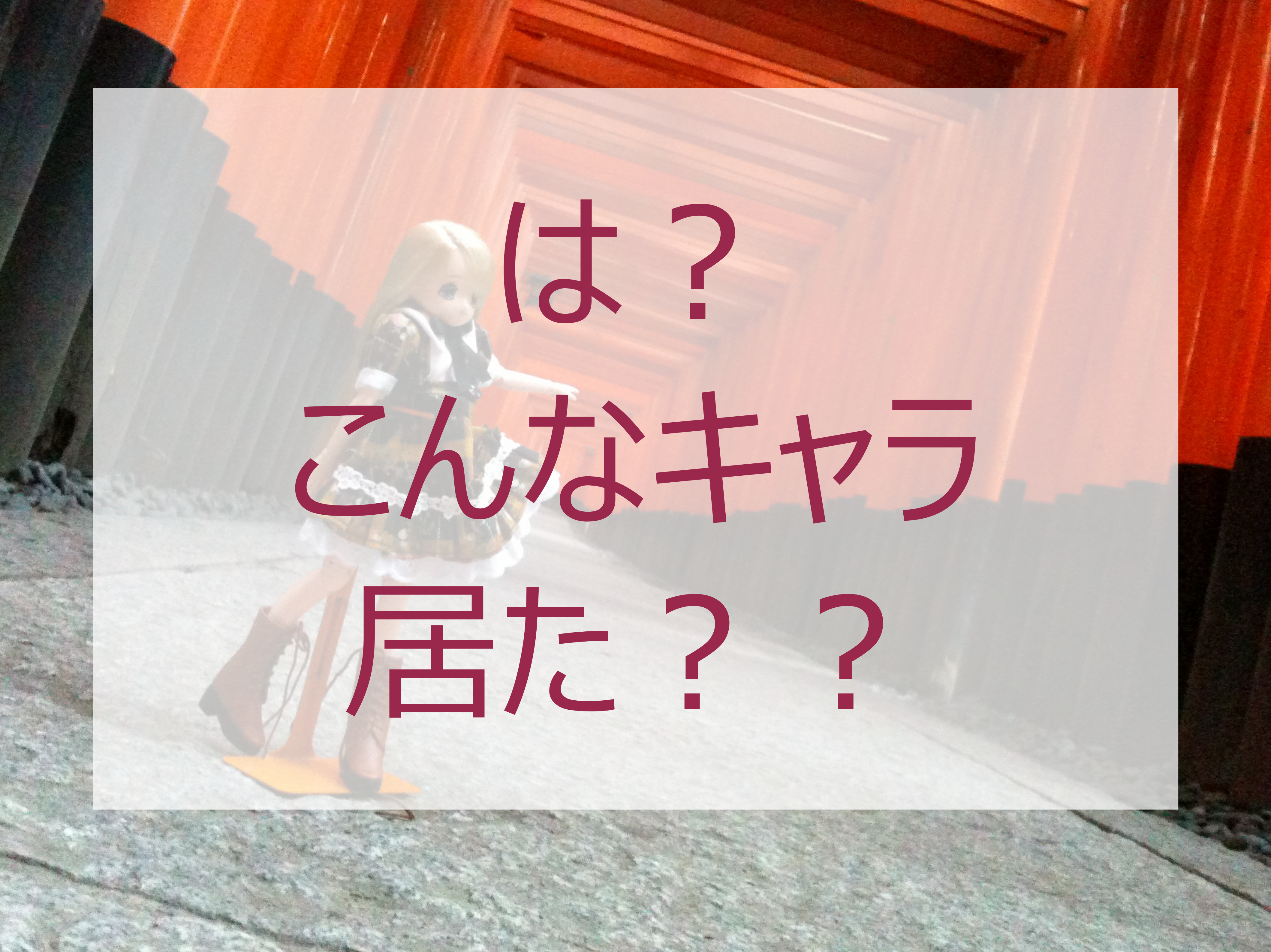
ほんと
だもん！

であと格好

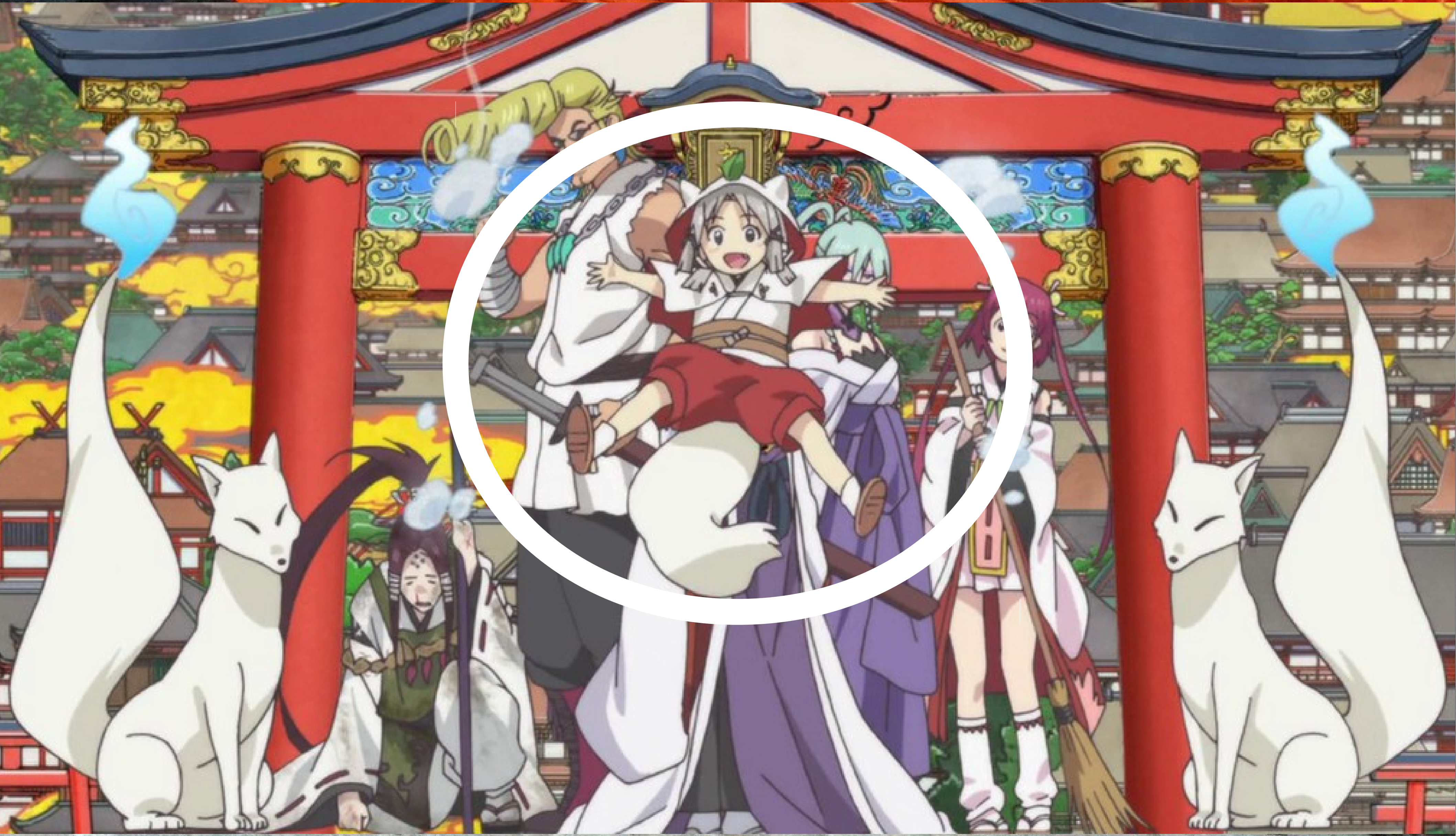
いなり、こんこん、
恋いろは。

コンちゃん





は？
こんなキャラ
居た？？



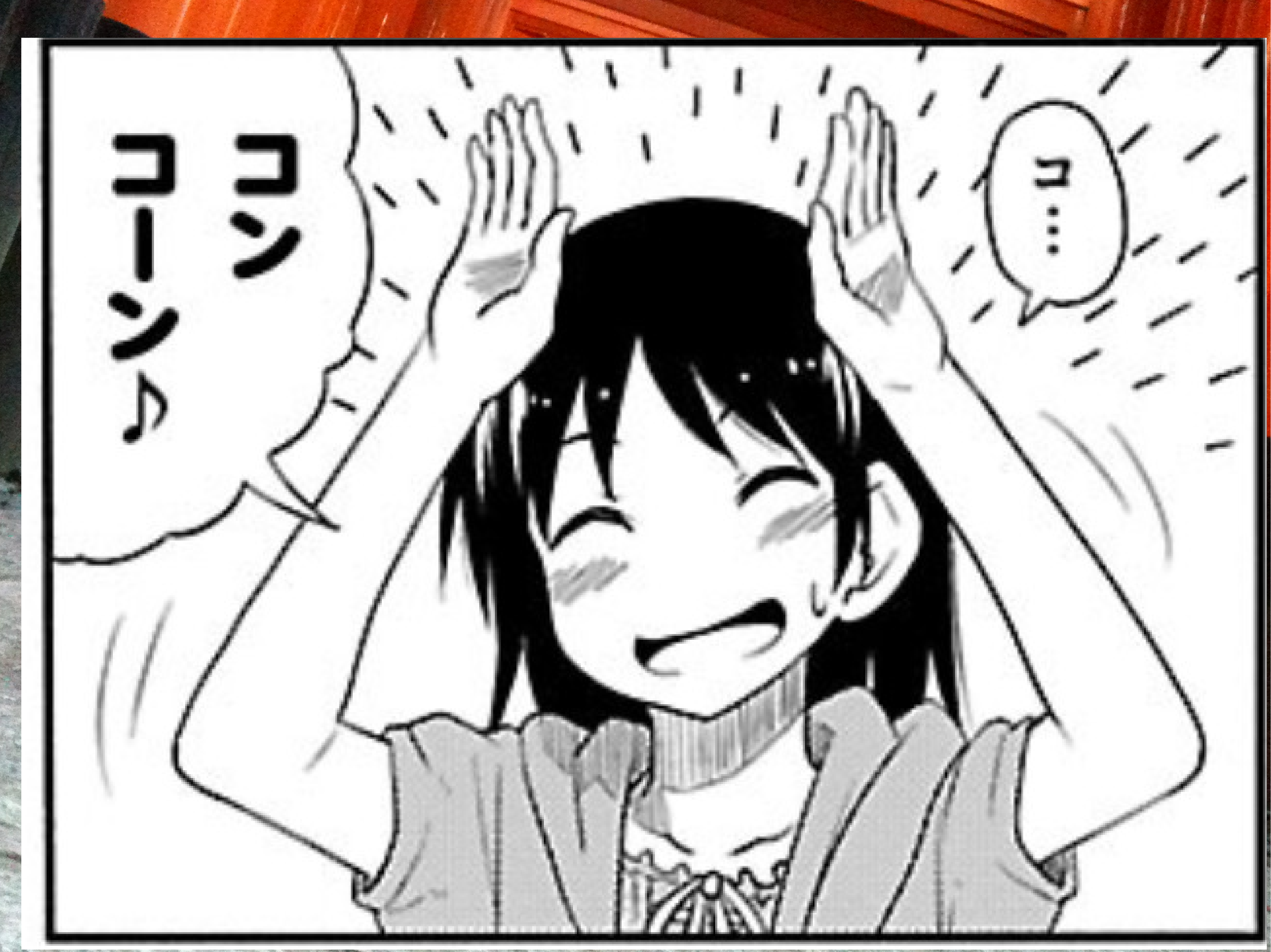
OPで

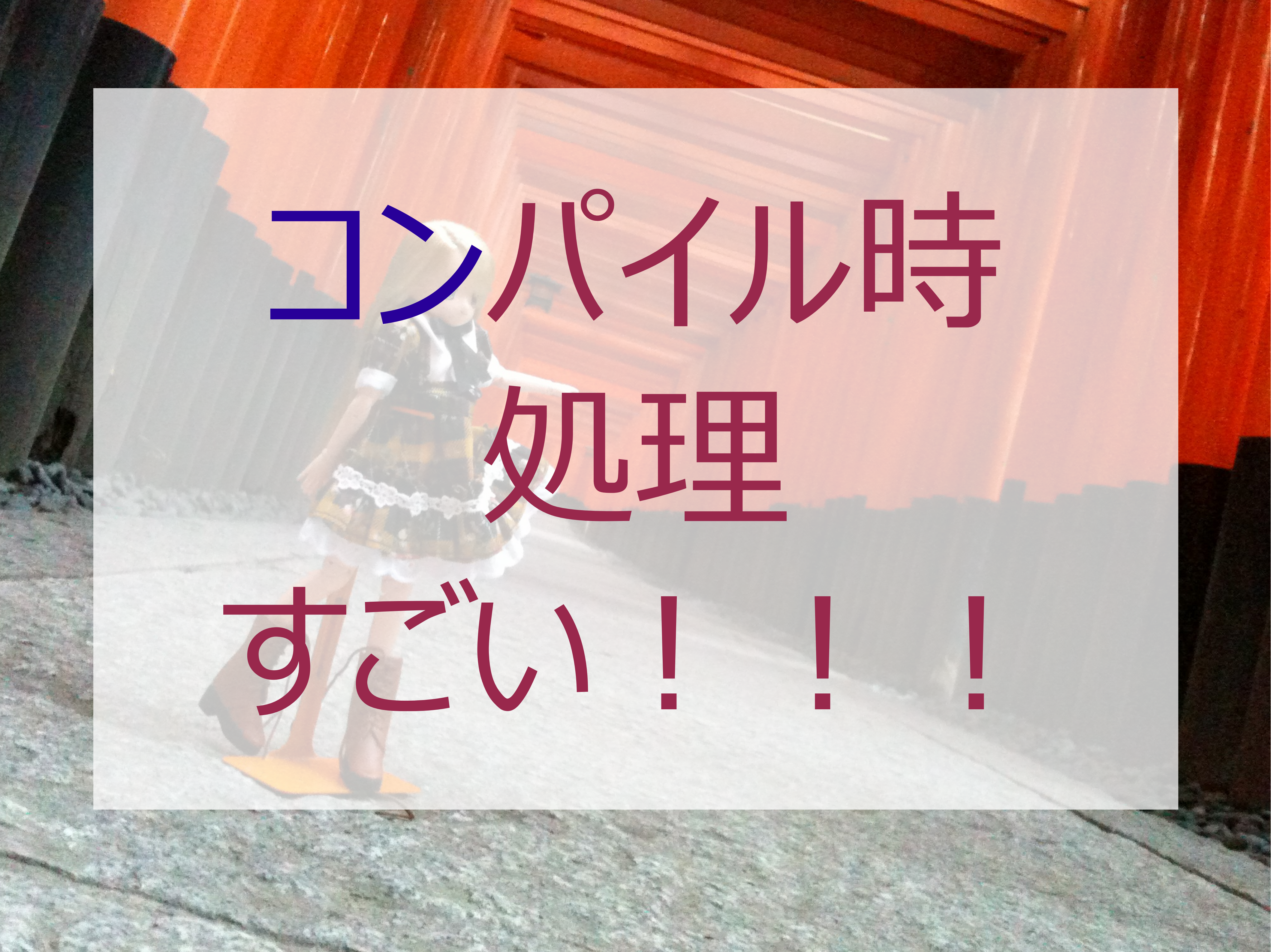
0.5秒

出てくる

背景も
伏見稲荷大社
@京都

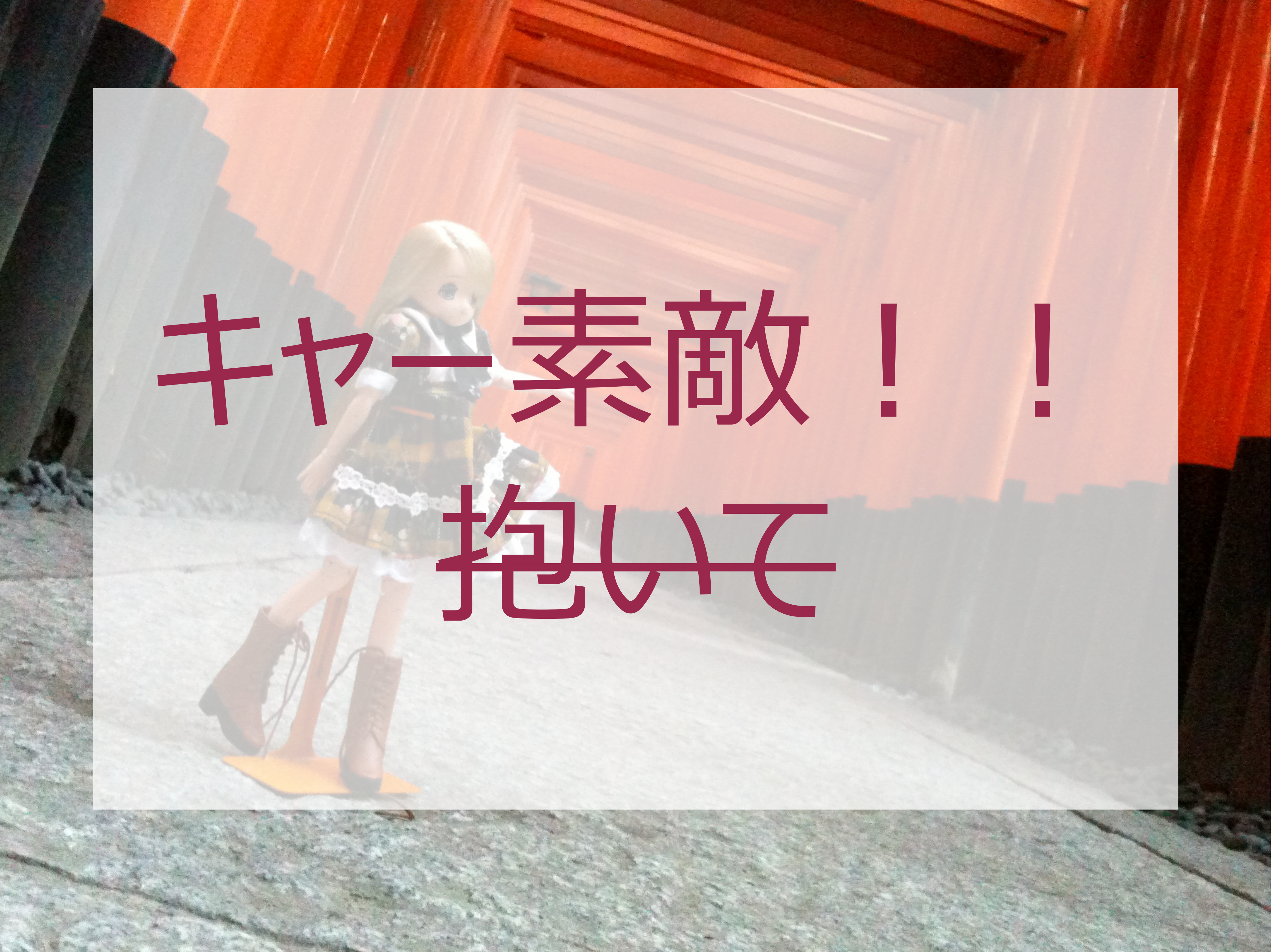
本題します


コ…
コンコーン♪



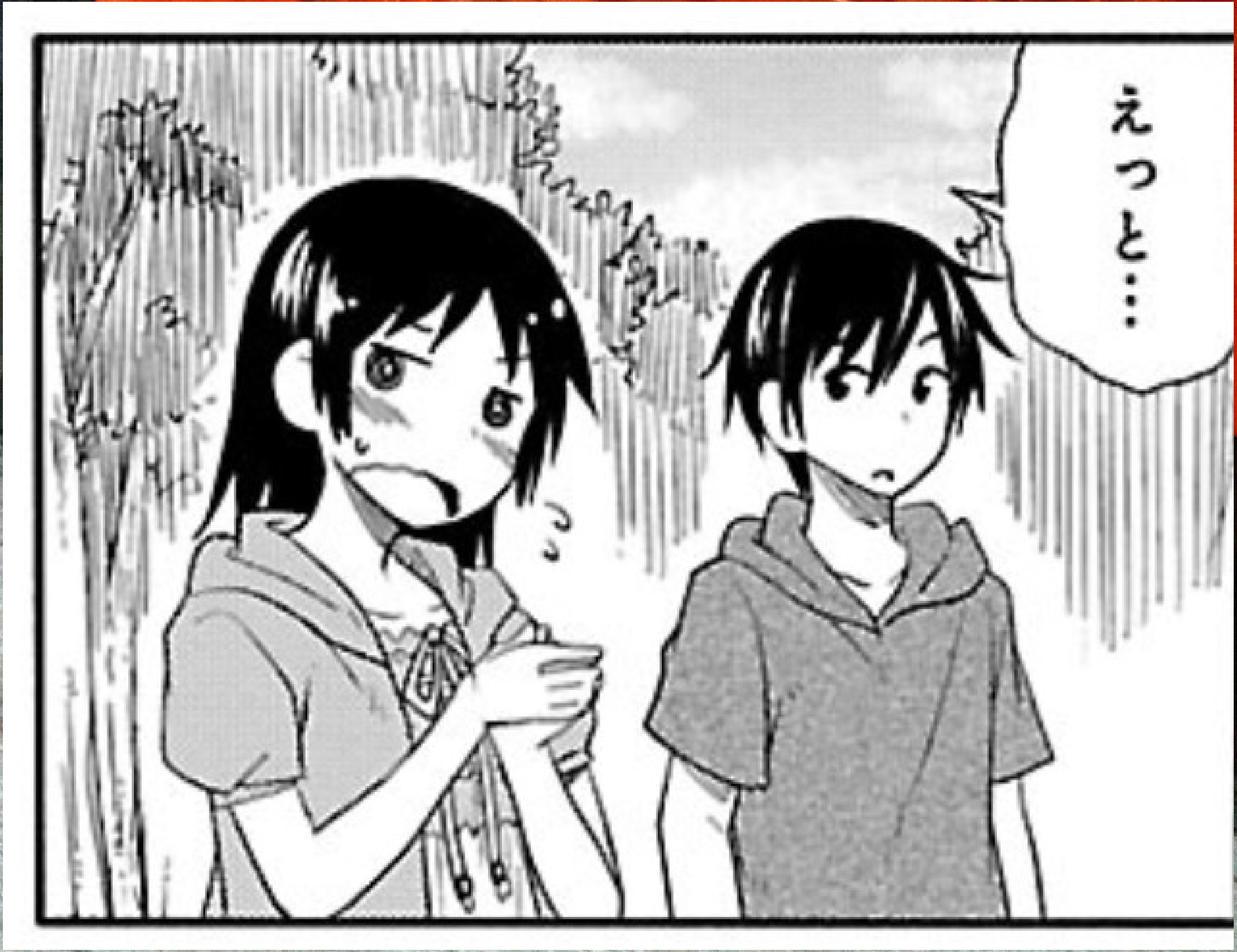
コンパイル時処理すごい！！！

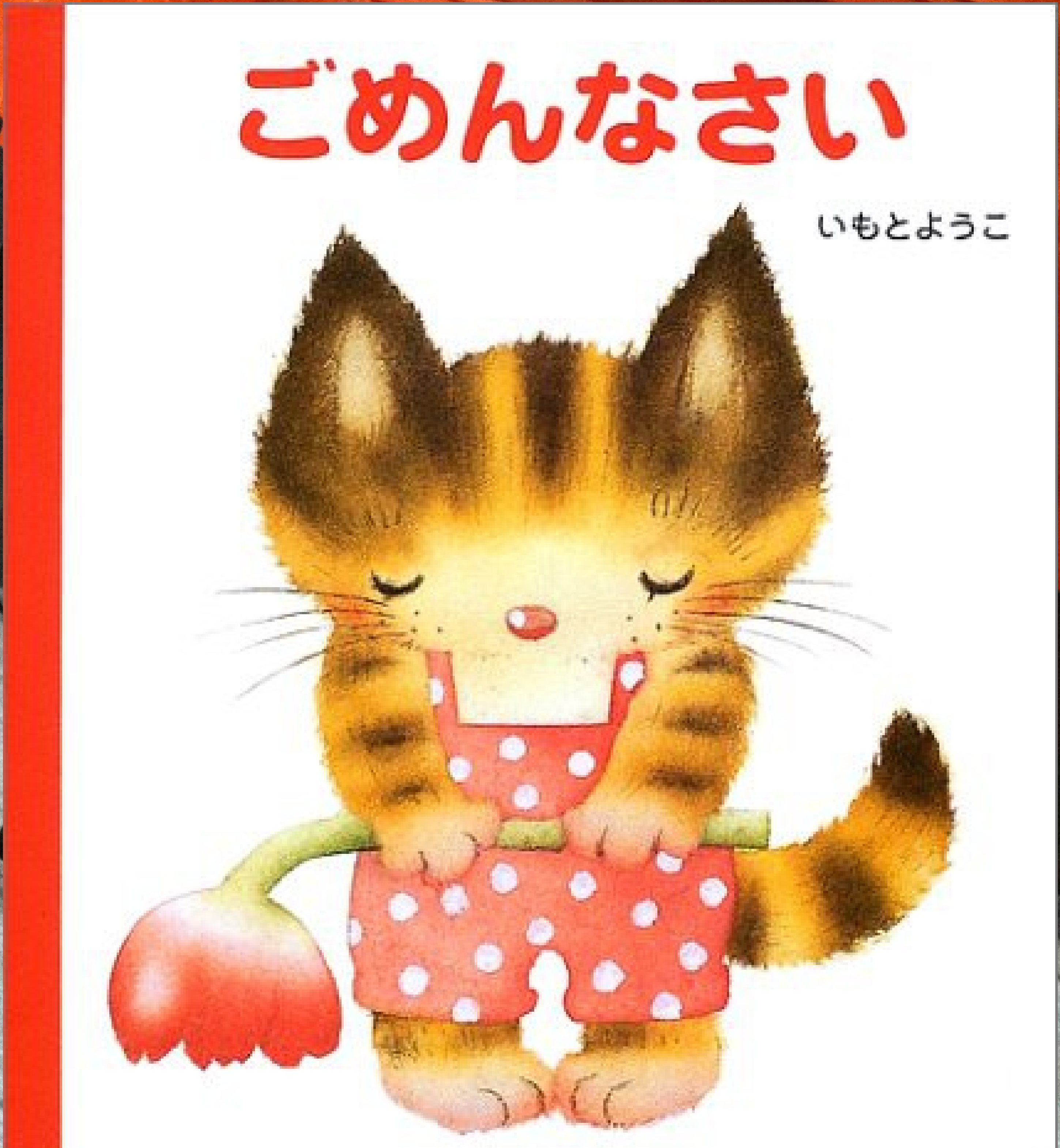
メタプログラミング すごい！！



キャー素敵！！

~~抱いて~~


えっと…



ごめんなさい

いもとようこ

悲しい
お知らせが
あります

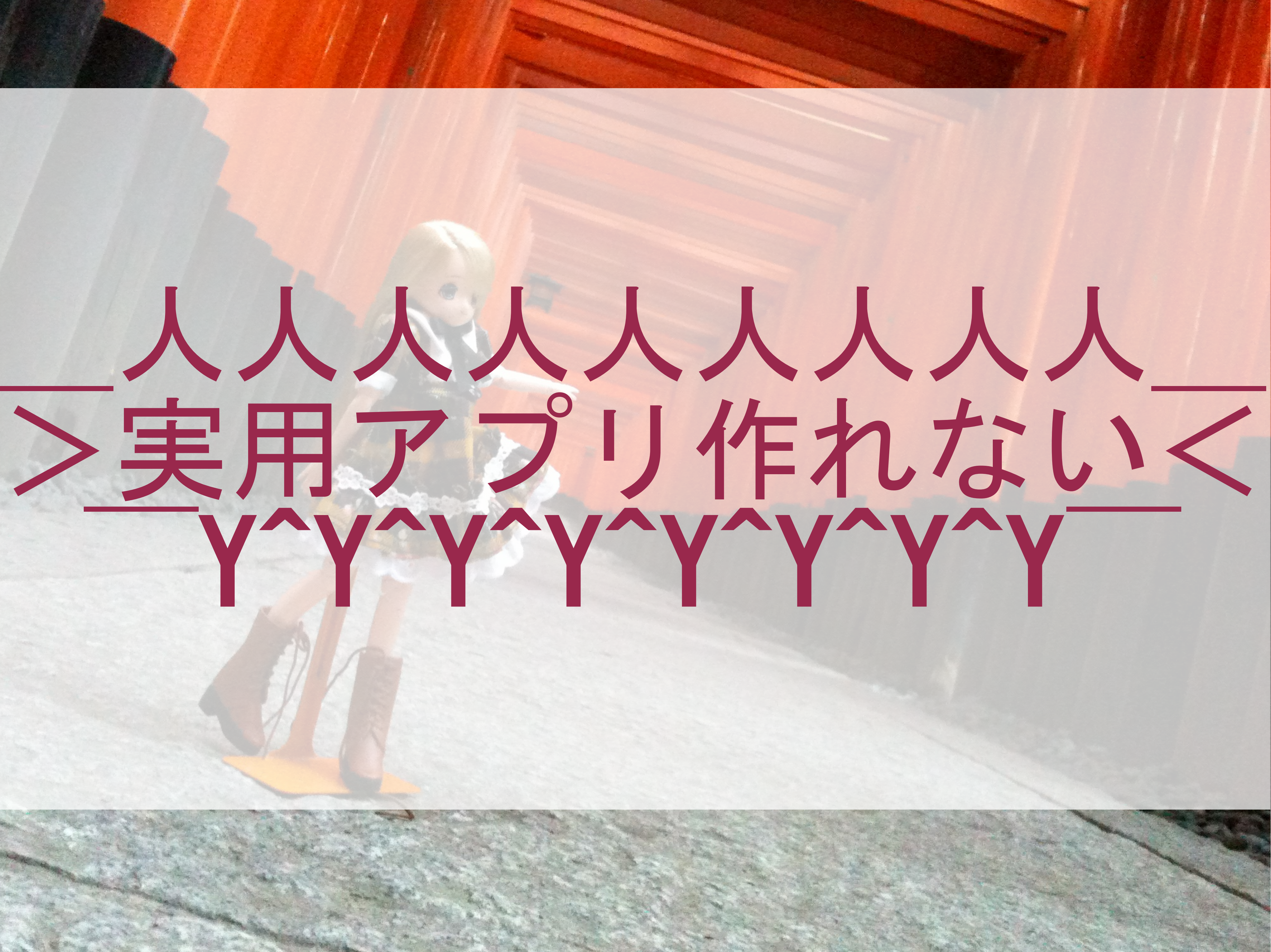
コンパイル時処理

だけでは

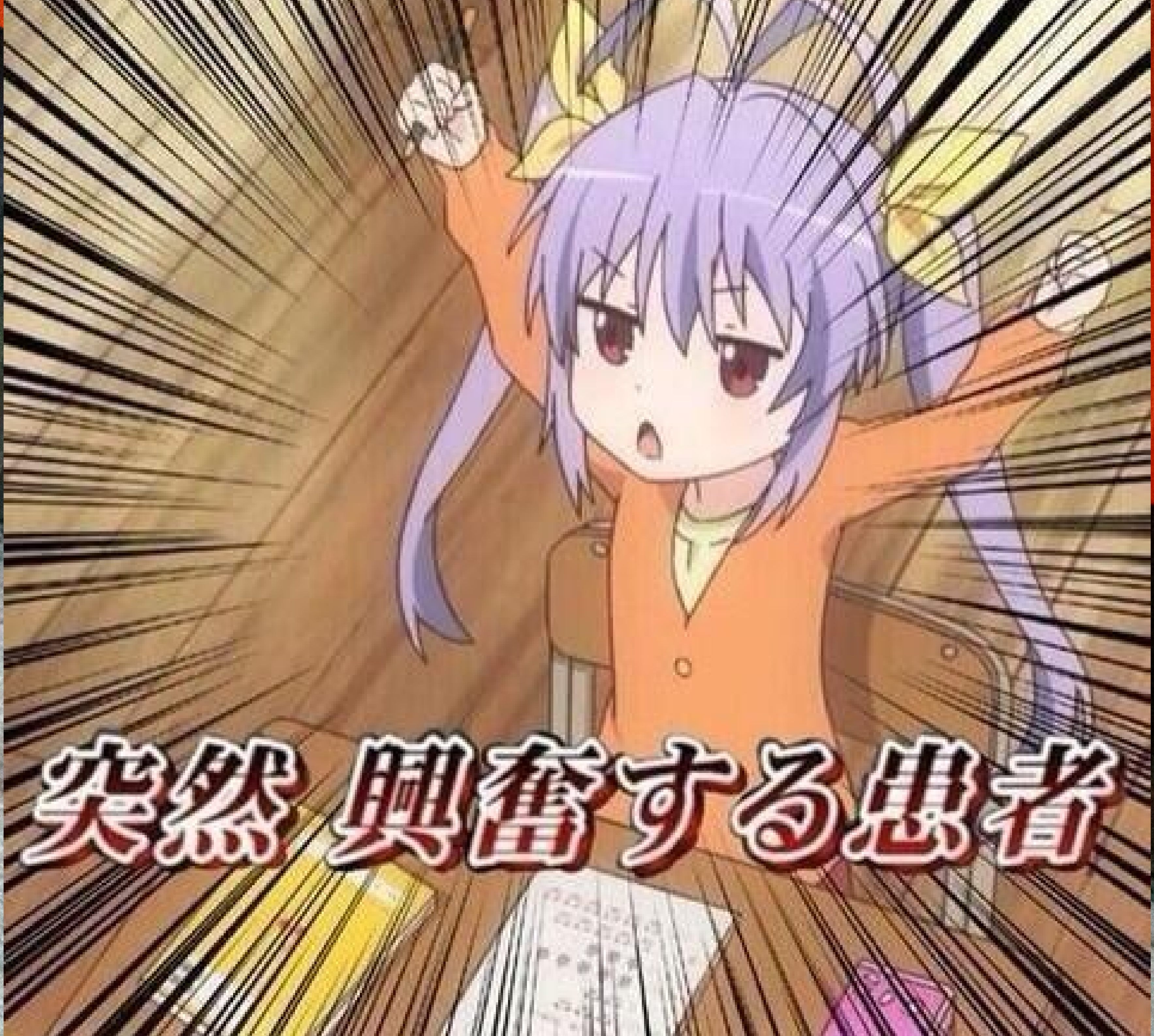
なんと

なんと

なんと



＿人人人人人人人人人＿
＞実用アプリ作れない＜
￣Y^Y^Y^Y^Y^Y^Y^Y￣


突然 興奮する患者

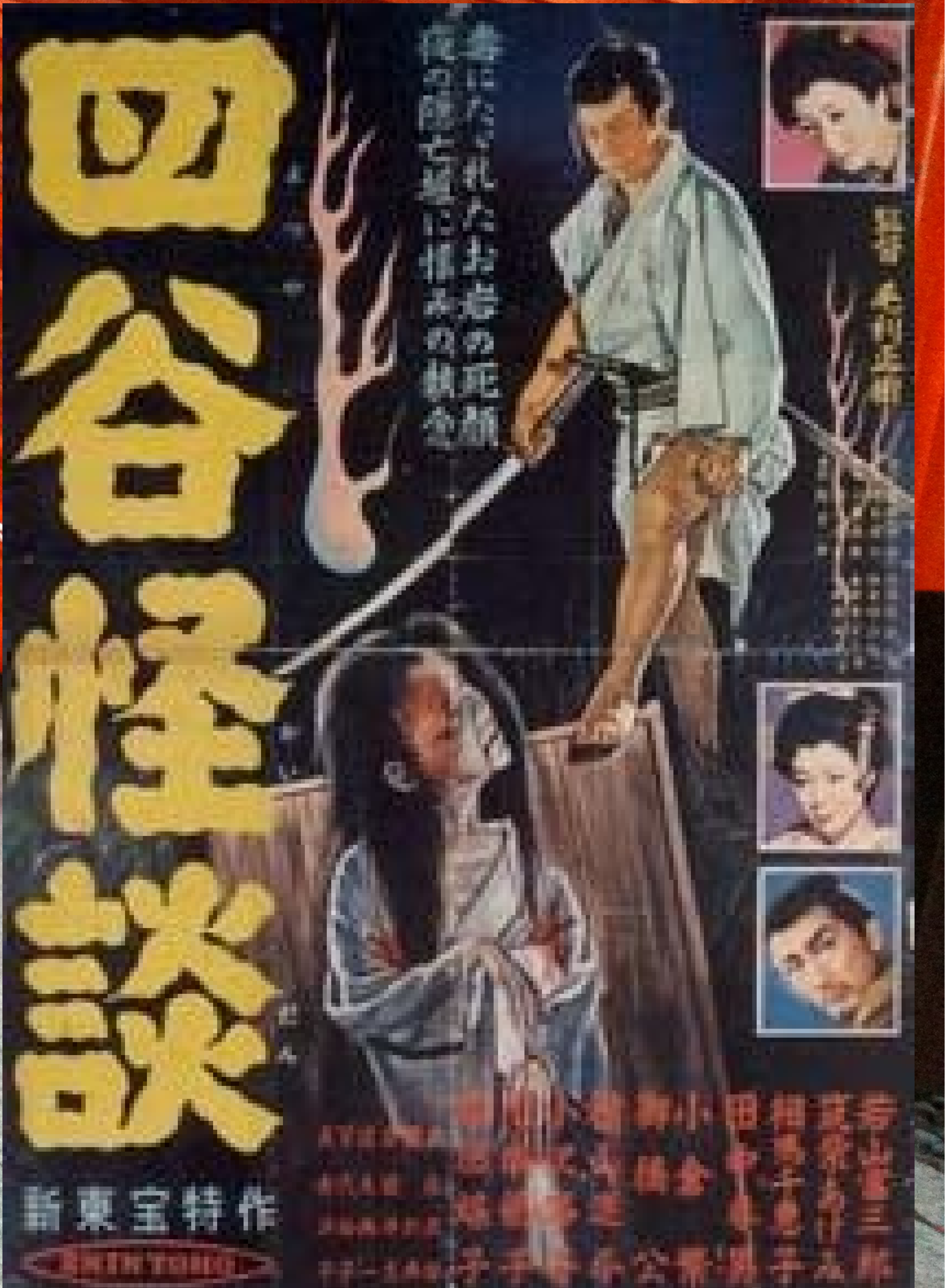
現実世界に生きる我々

実行時からは 逃れられない！

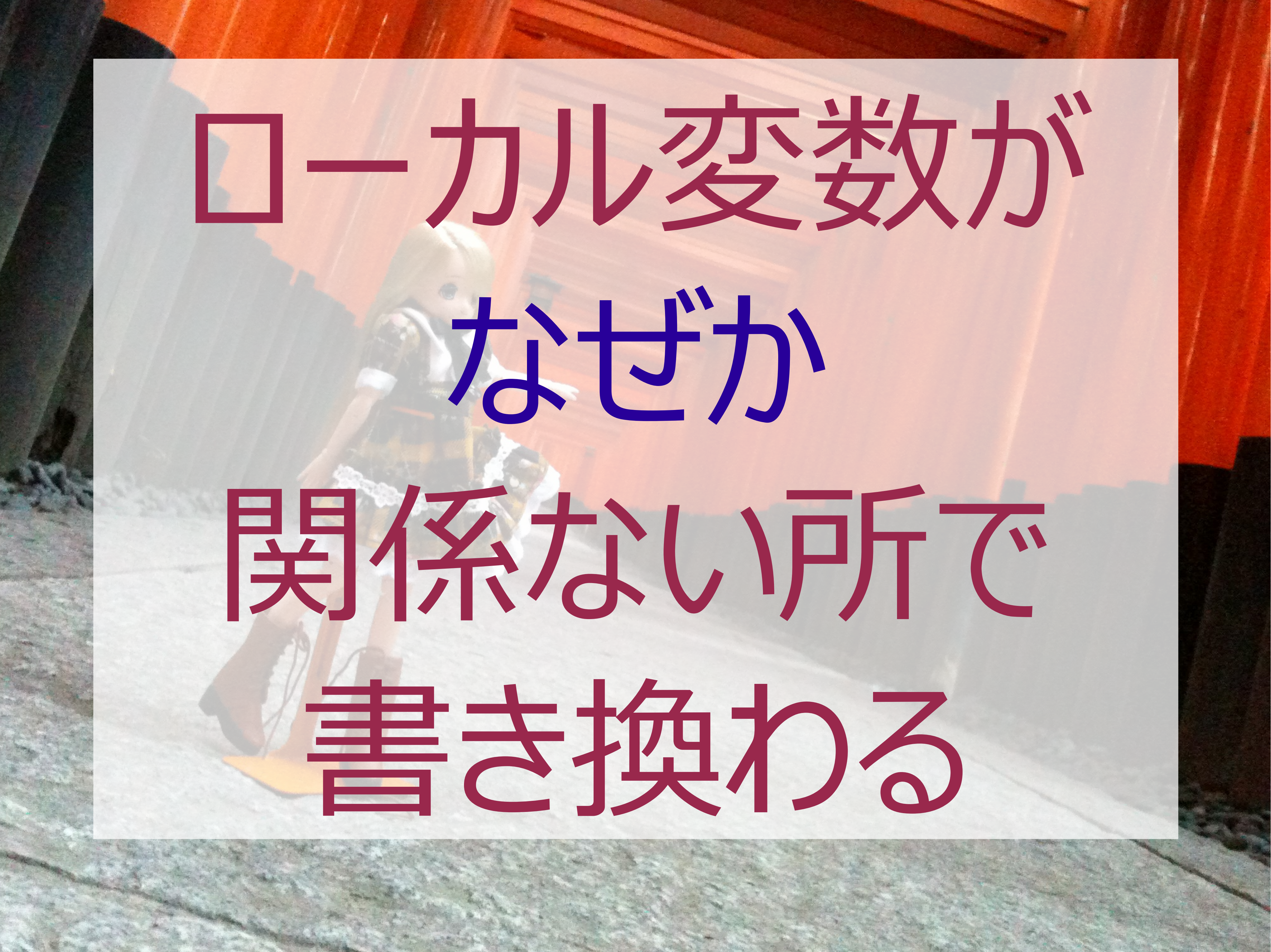
そして付き纏う

実行時エラー
という
″″″怪談″″″

これは
本当に
あった
話です

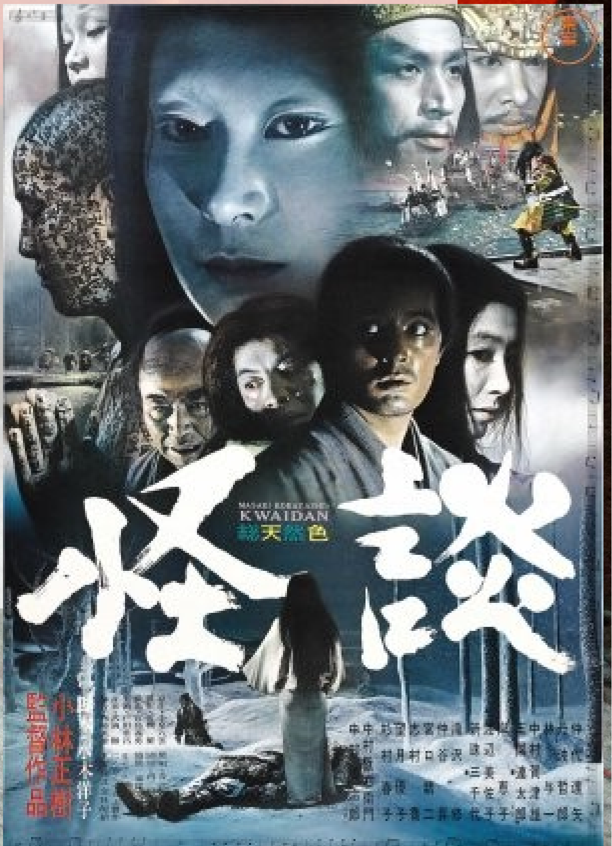


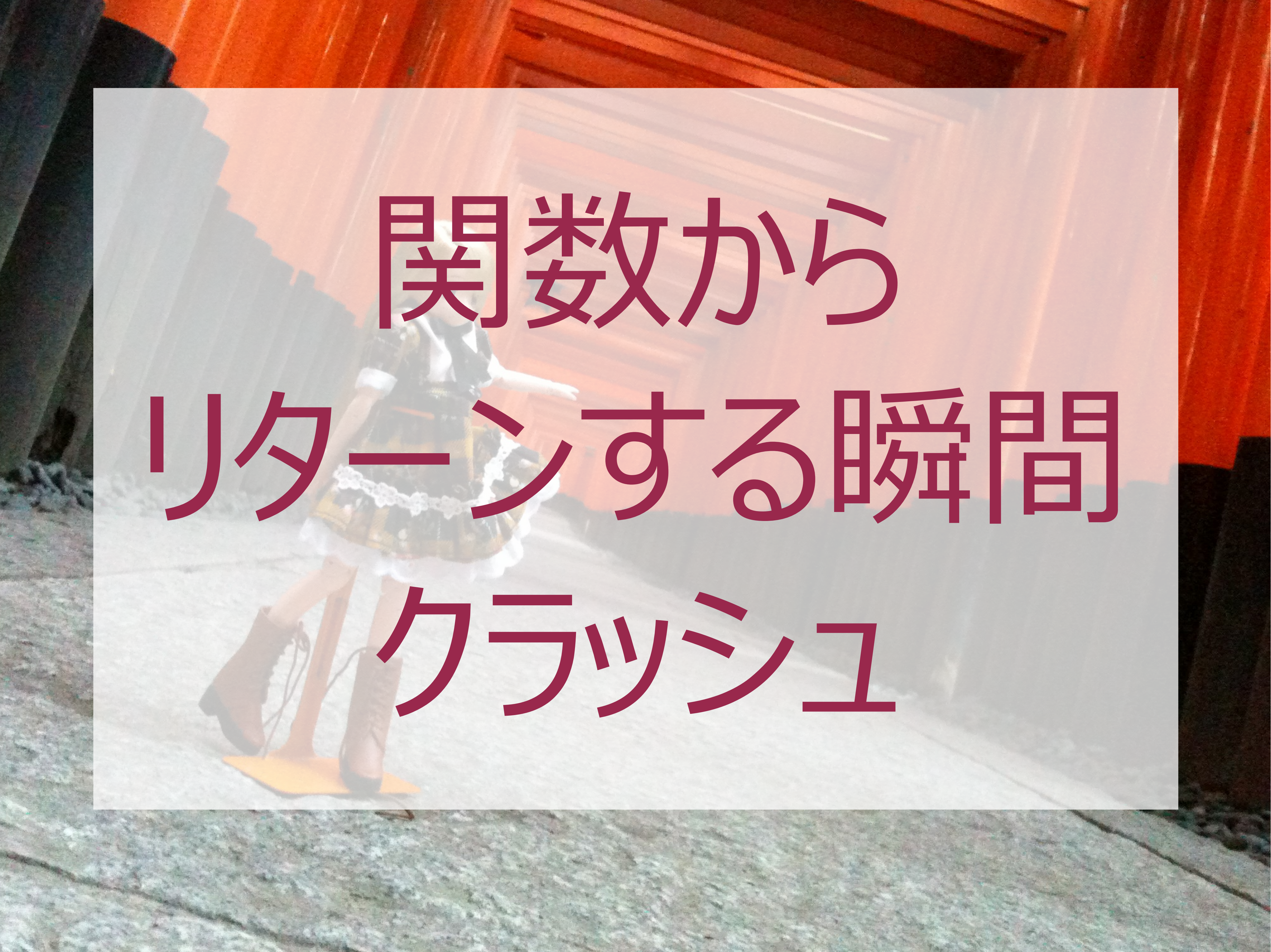
曰く、



ローカル変数がなぜか関係ない所で書き換わる

曰く、


MASAKI KOBAYASHI'S
KWAIDAN
総天然色
怪談
小林正樹
監督作品



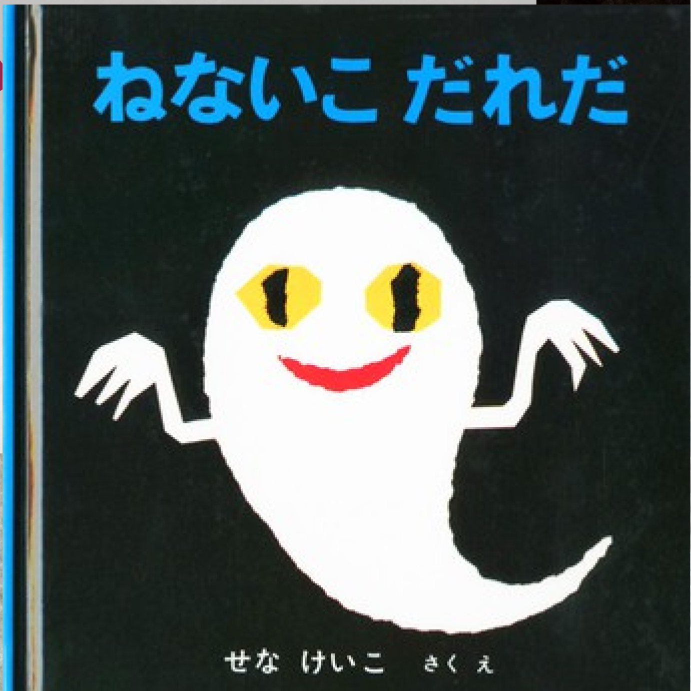
関数からリターンする瞬間クラッシュ

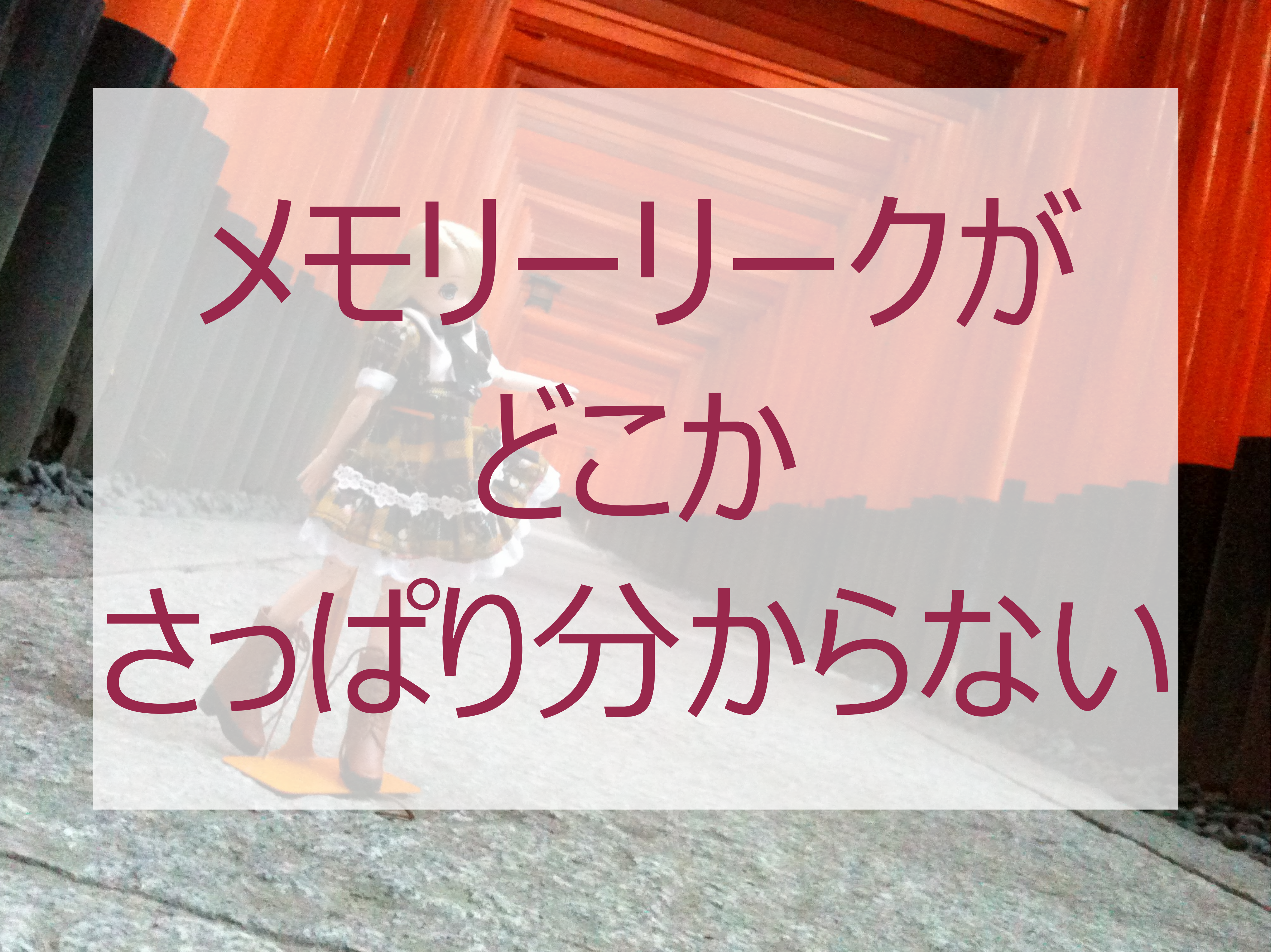
曰く、

DVD VIDEO
日本
怪談劇場
第一巻

new T()
しようとした瞬間
クラッシュ

曰く、

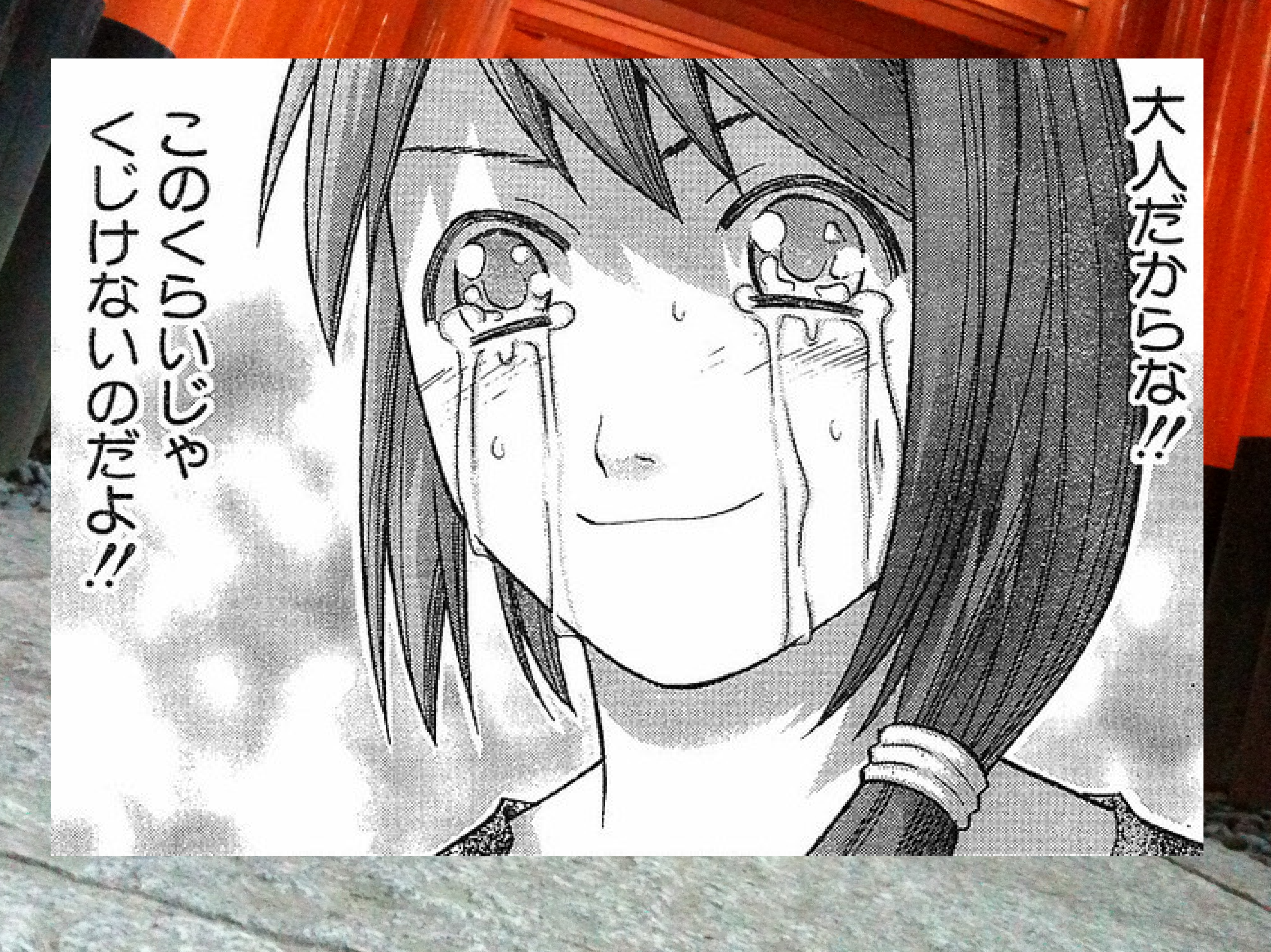

ねないこ だれだ
せな けいこ さく え

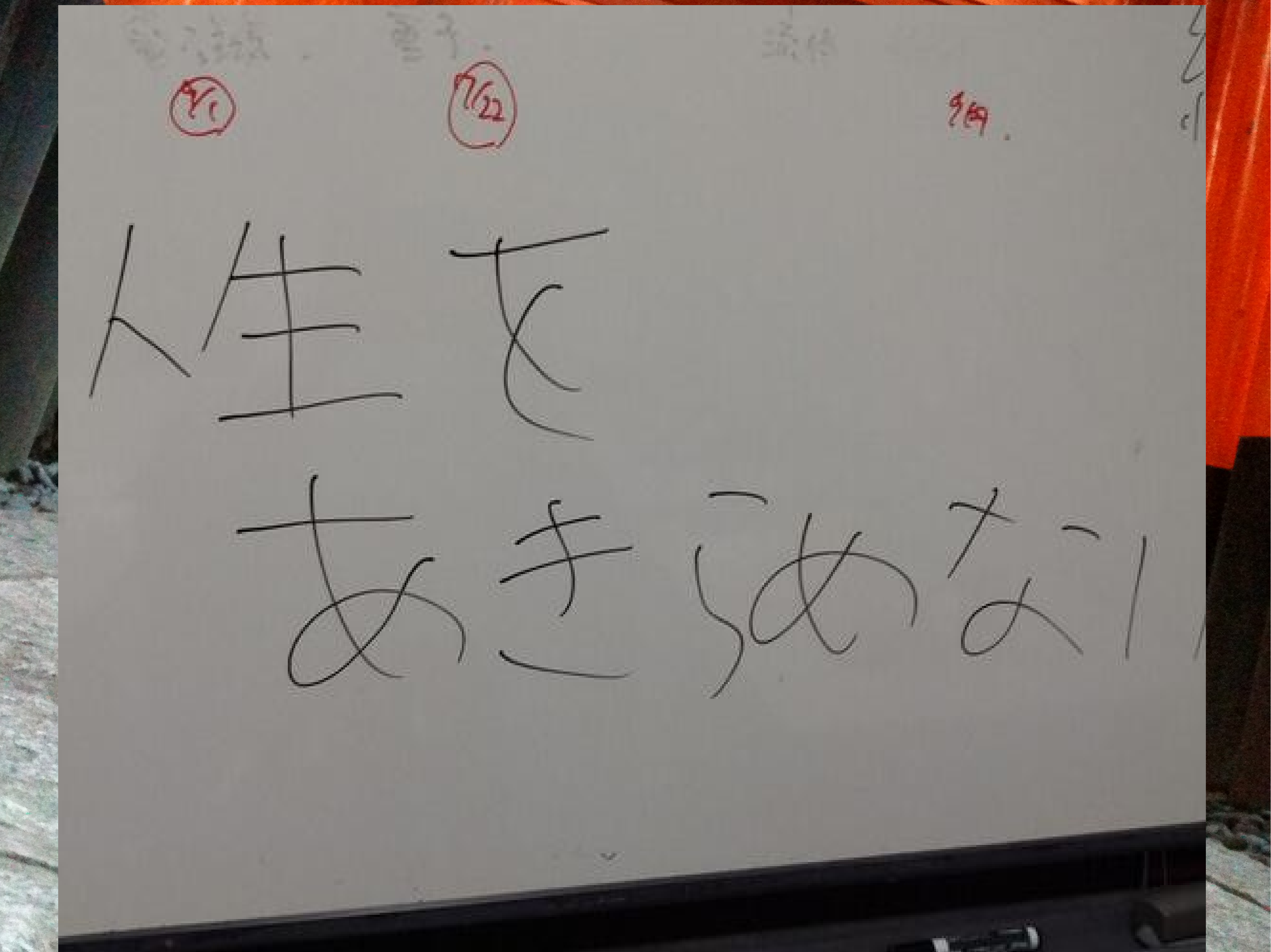


メモリーリークが
どこか
さっぱり分からない

曰く、

オブジェクトが
知らない所で
deleteされてた


大人だからな‼
このくらいじゃ
くじけないのだよ‼



5/1　7/22　9/19.

人生を

あきらめない

実は
SIGSEGVで
落ちる分にはラク

SIGSEGVで死

⇒

SIGSEGVで死

⇒

そこがバグの本丸

そこを
念入りに調べれば
大体直る

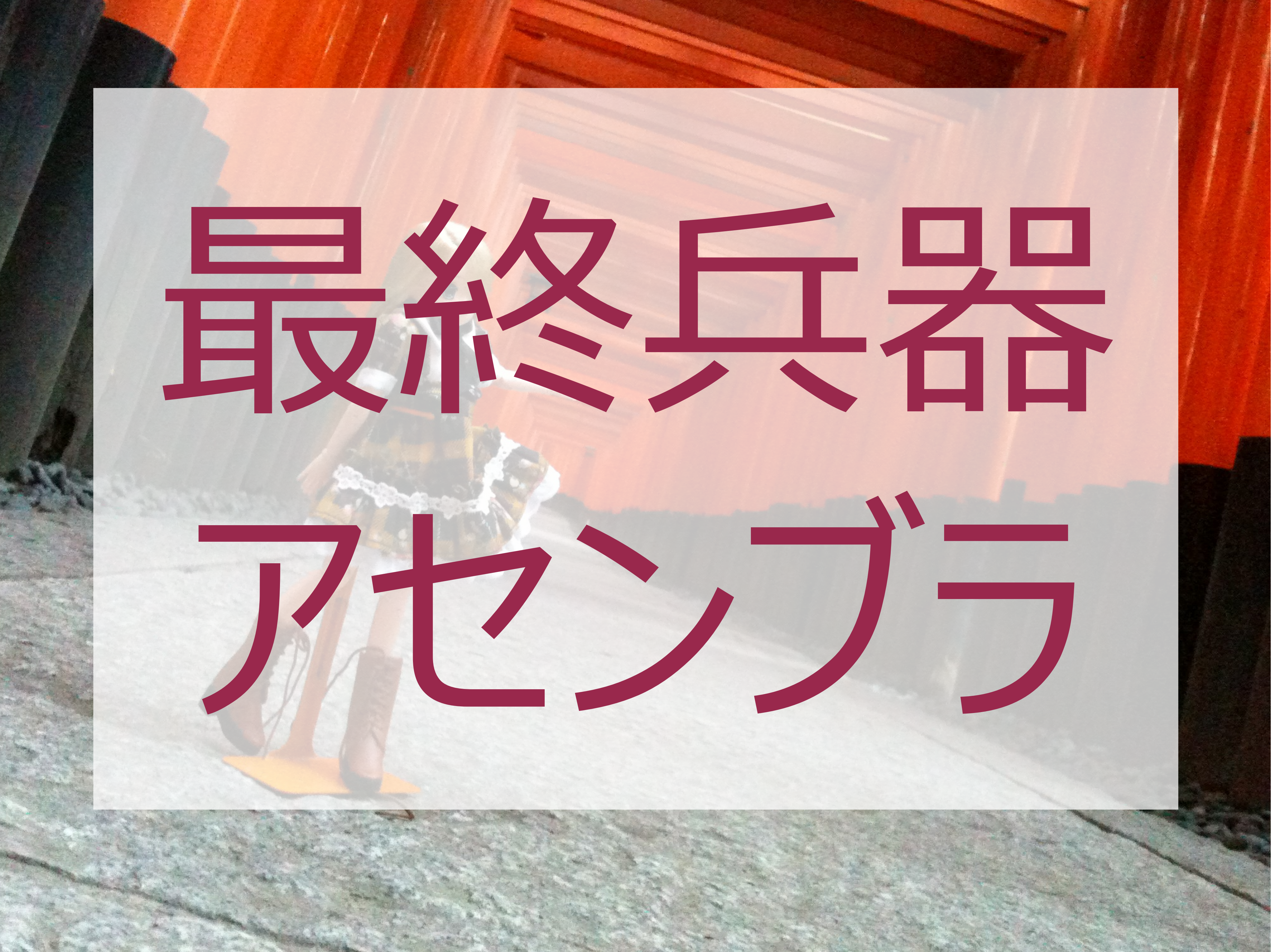
でも

怪談バグ
is
よくわからない

原因と
結果が
離れている！

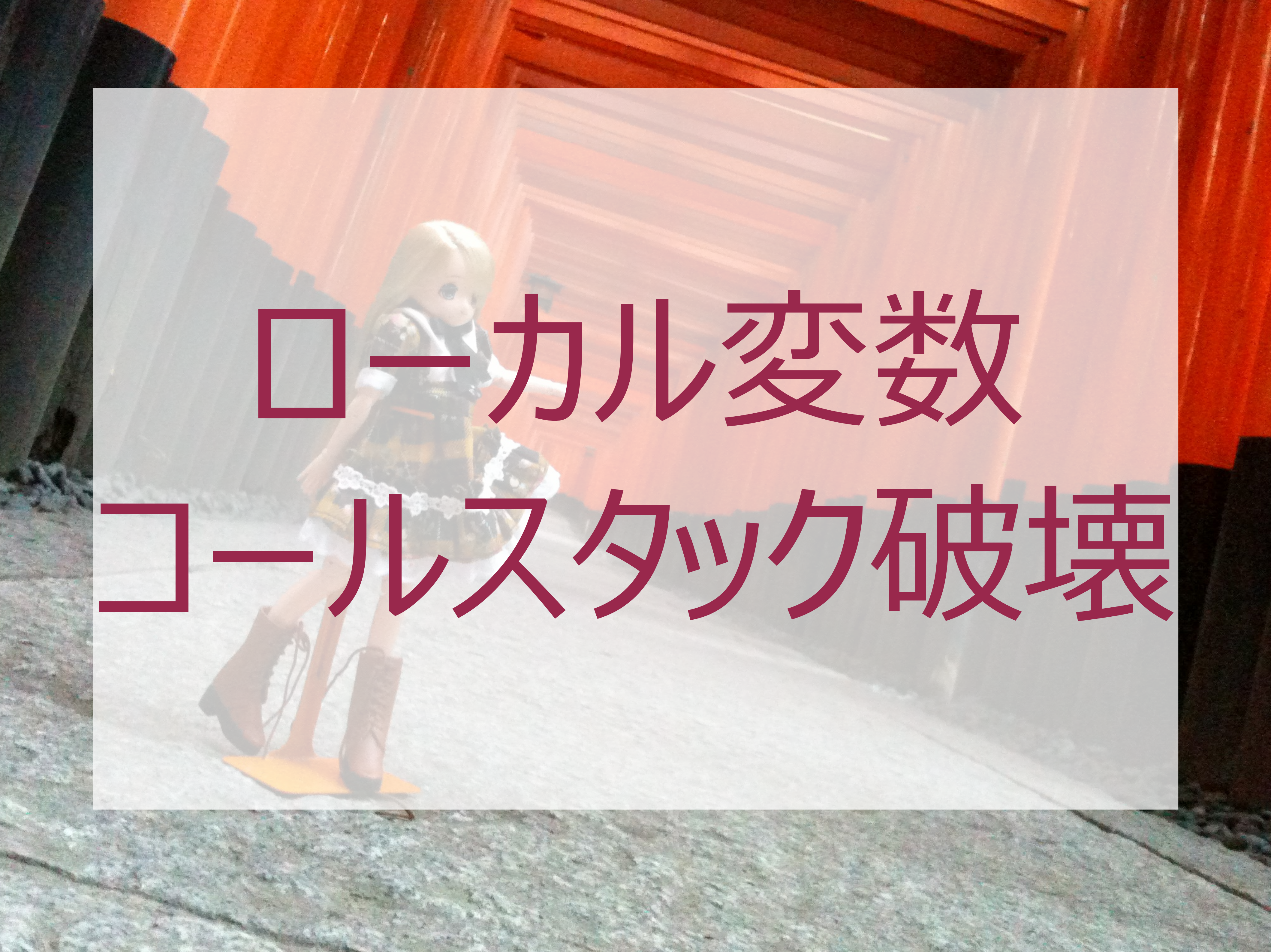
どないすねん

事前にしっかり
確認しましょう。



最終兵器
アセンブラ

```
gdb -S
odjdump -D
```

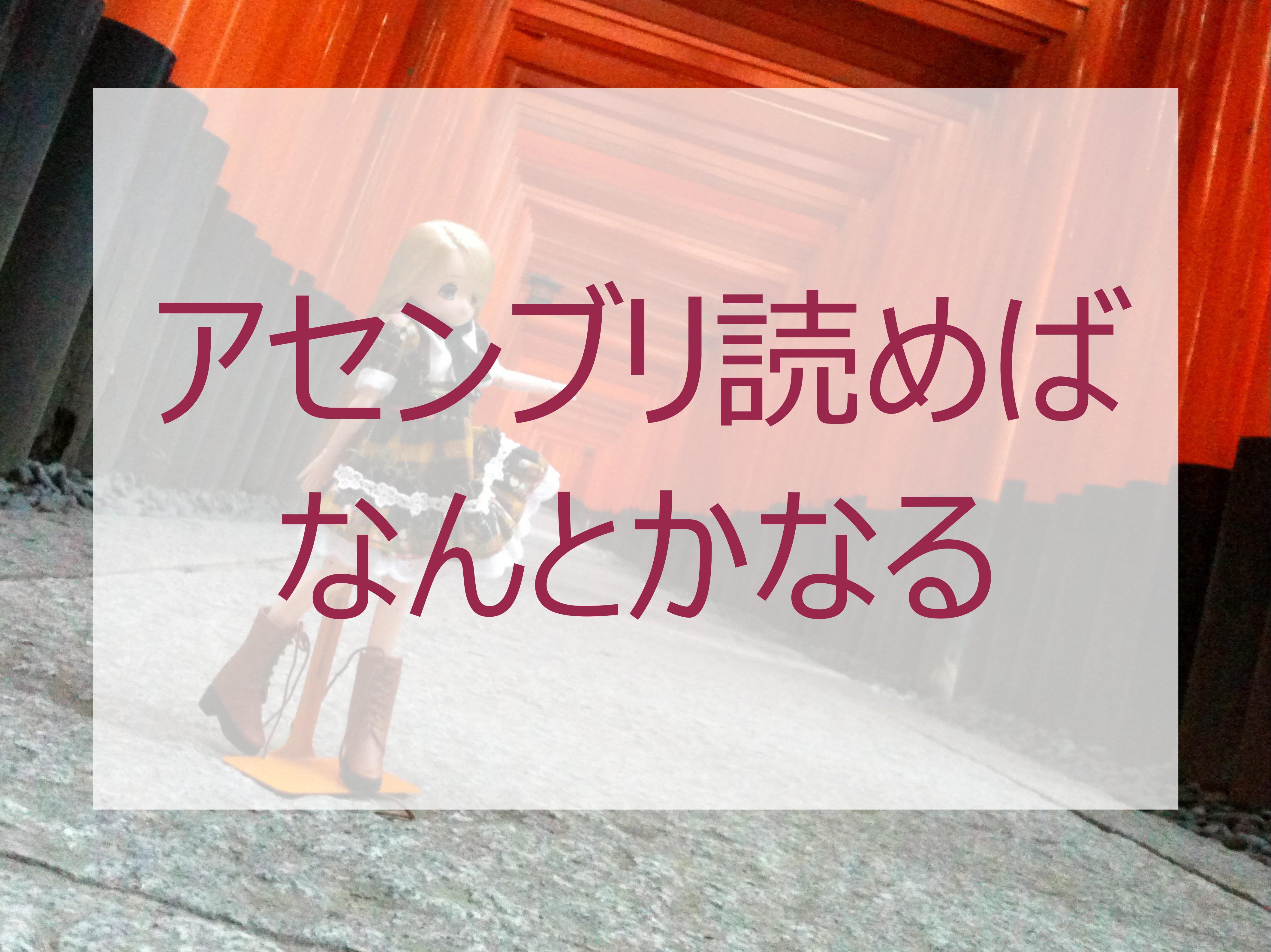


ローカル変数
コールスタック破壊

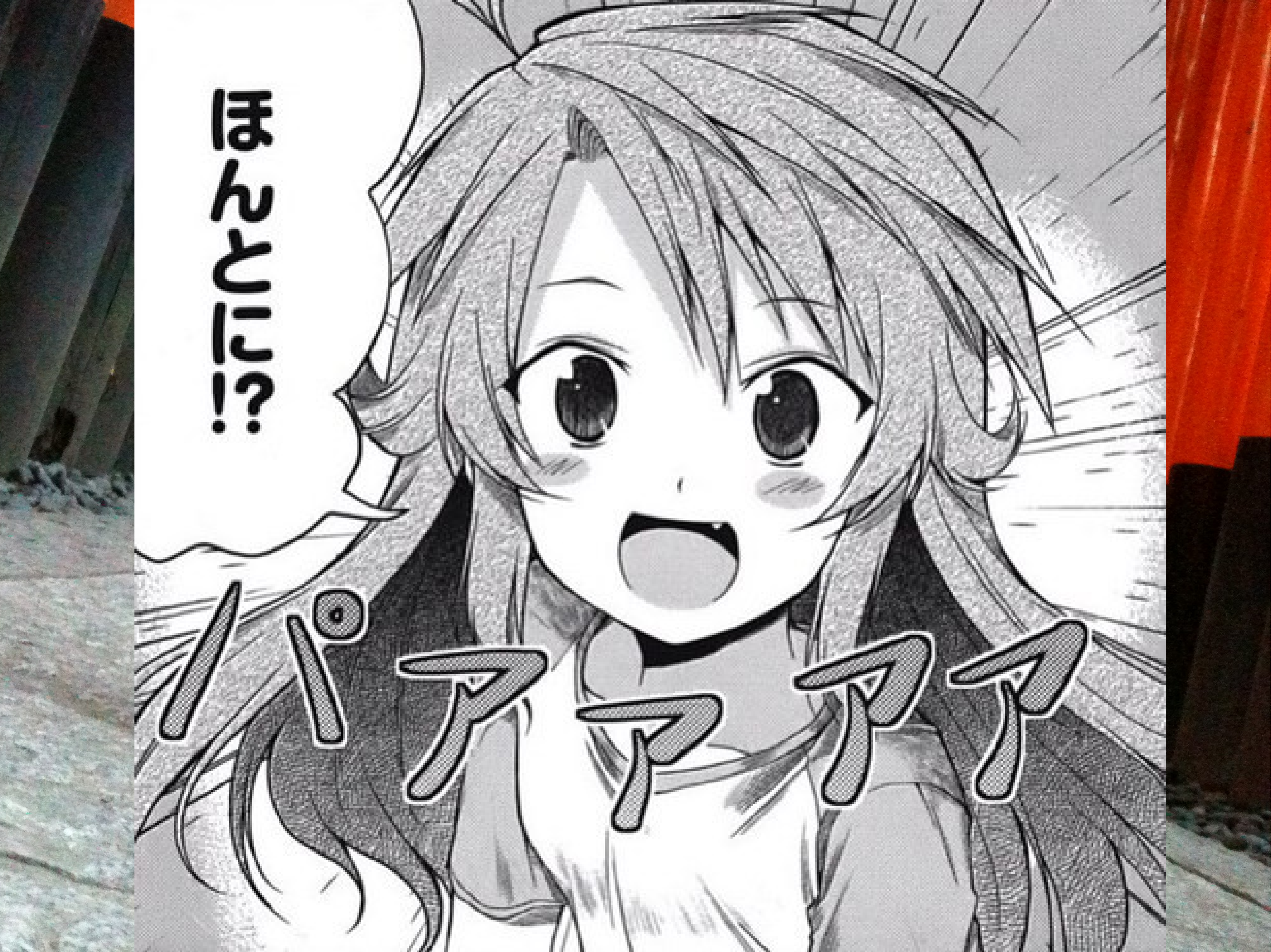
ローカルな変なバグは

最悪

アセンブリ読めば



アセンブリ読めば なんとかなる


ほんとに!?
パァアア

ローカルな

バグは

ローカル

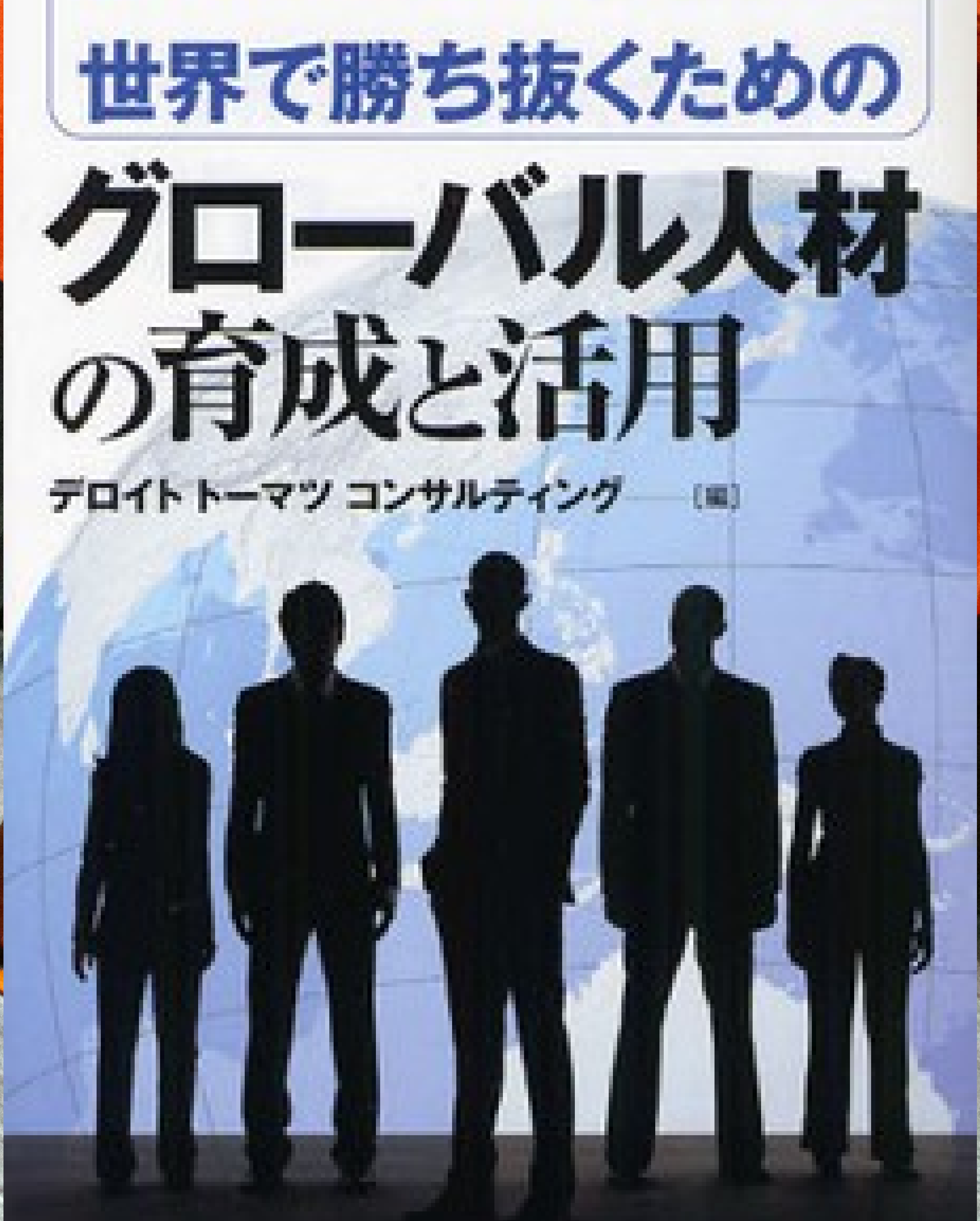
ローカル

⇔

ローカル

⇔

グローバル



世界で勝ち抜くための

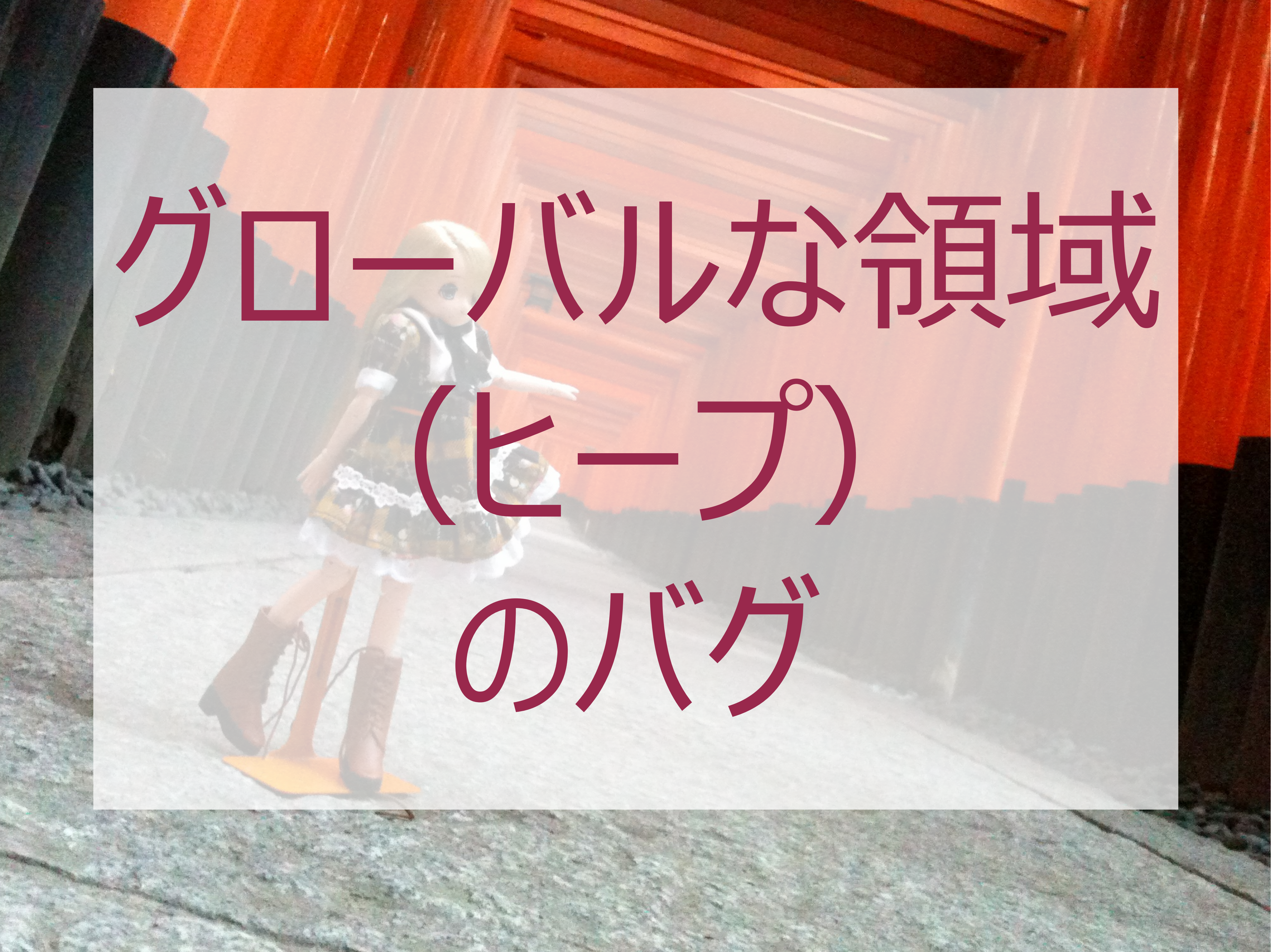
グローバル人材の育成と活用

デロイト トーマツ コンサルティング〔編〕

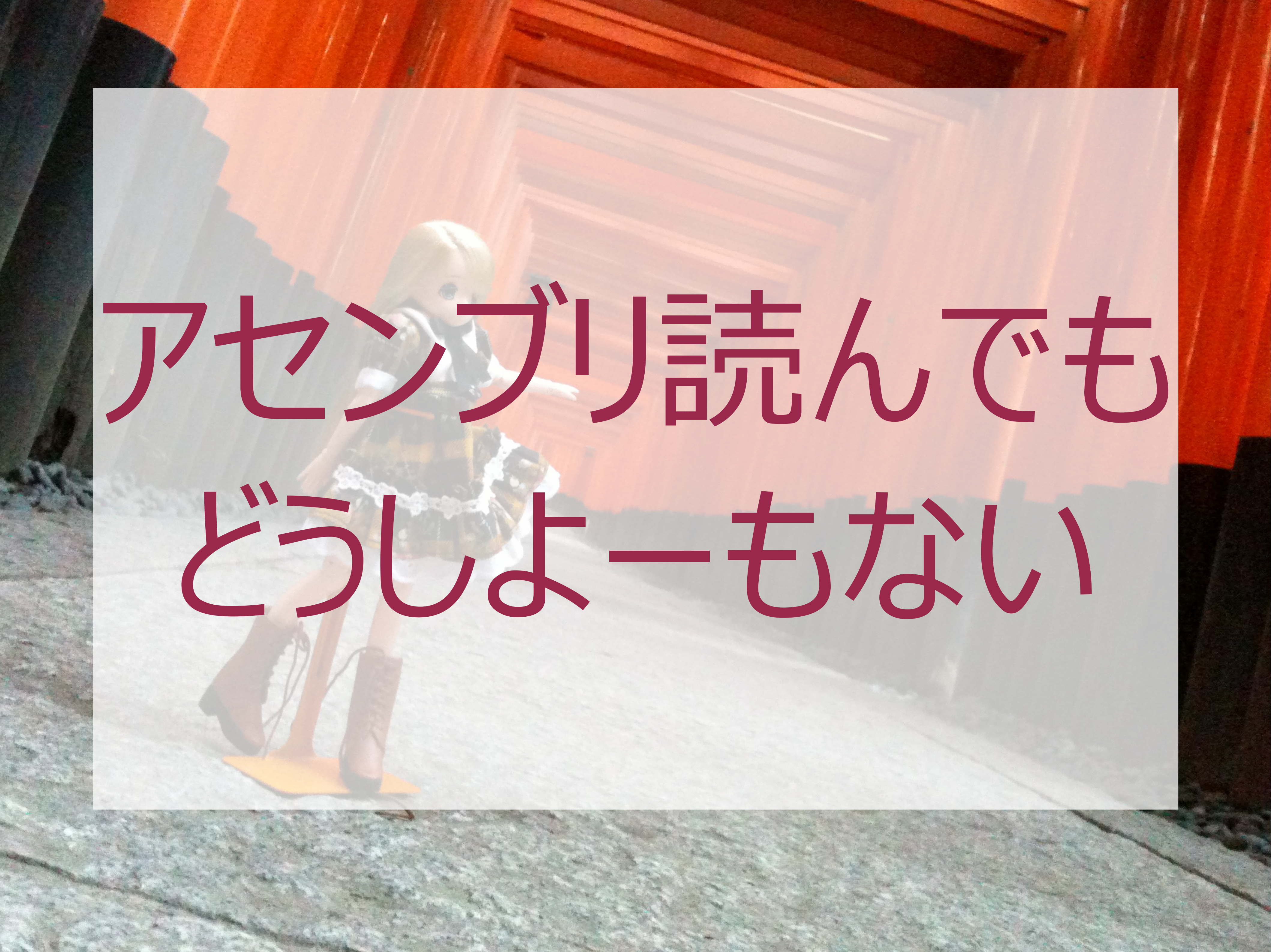
どうにもならない

メモリーリーク

メモリ管理領域の破壊



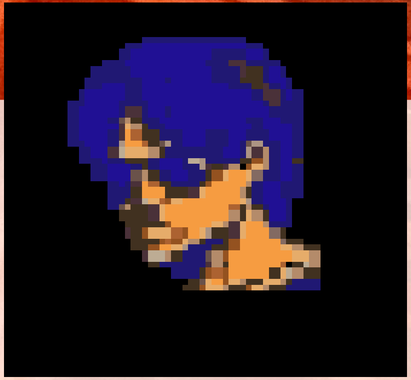
グローバルな領域（ヒープ）のバグ



アセンブリ読んでも どうしよーもない

とくに

cocos2d-x



てめェだよ
てめェー

ごとうきよはる　さんは今どこに？

Aボタン　おしてね。

スマートポインタ使ってない

全手動
retain release

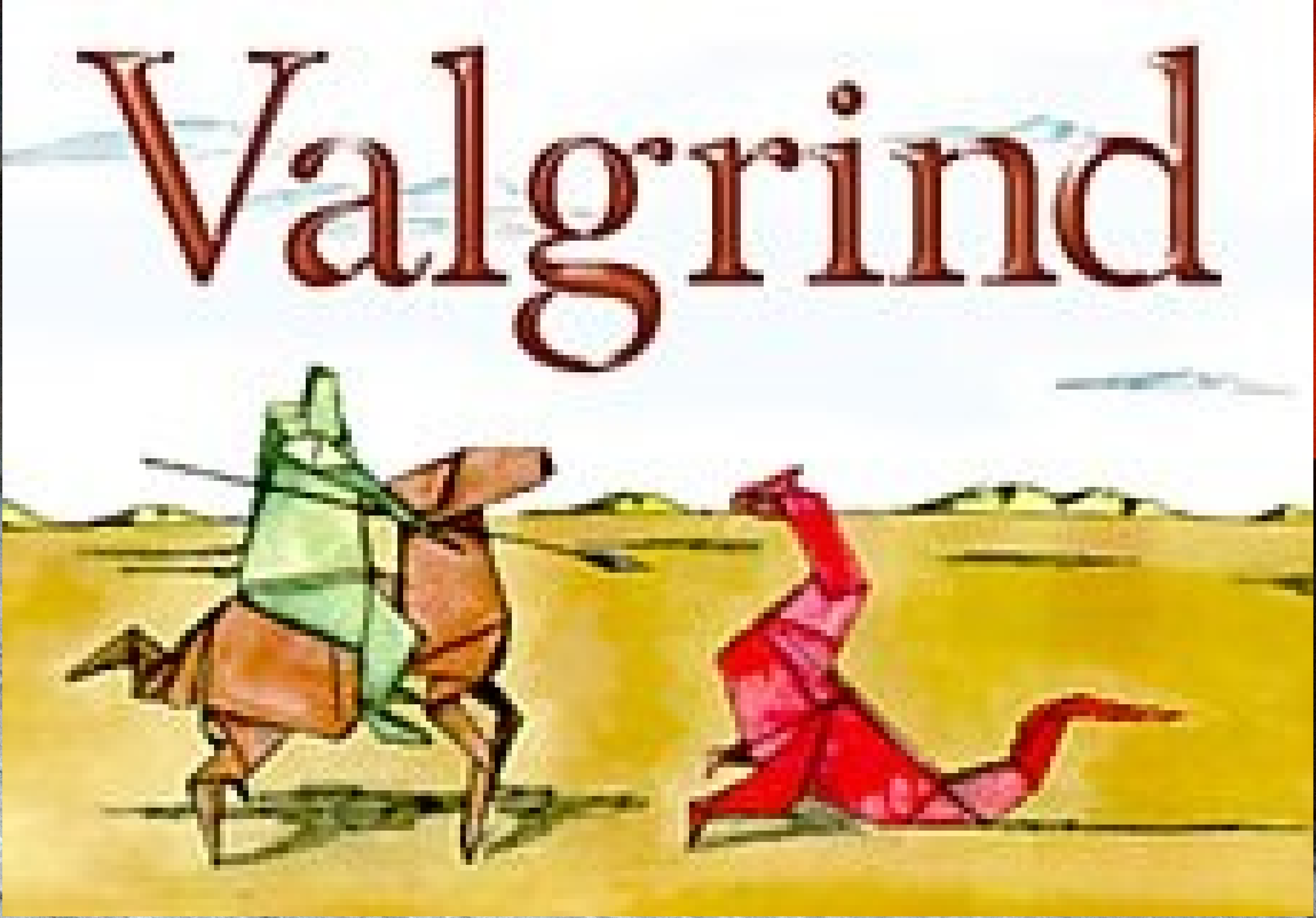
例外安全性 : ZERO

RAII…？

何それ…？

どないすねん（２度目）

そこで

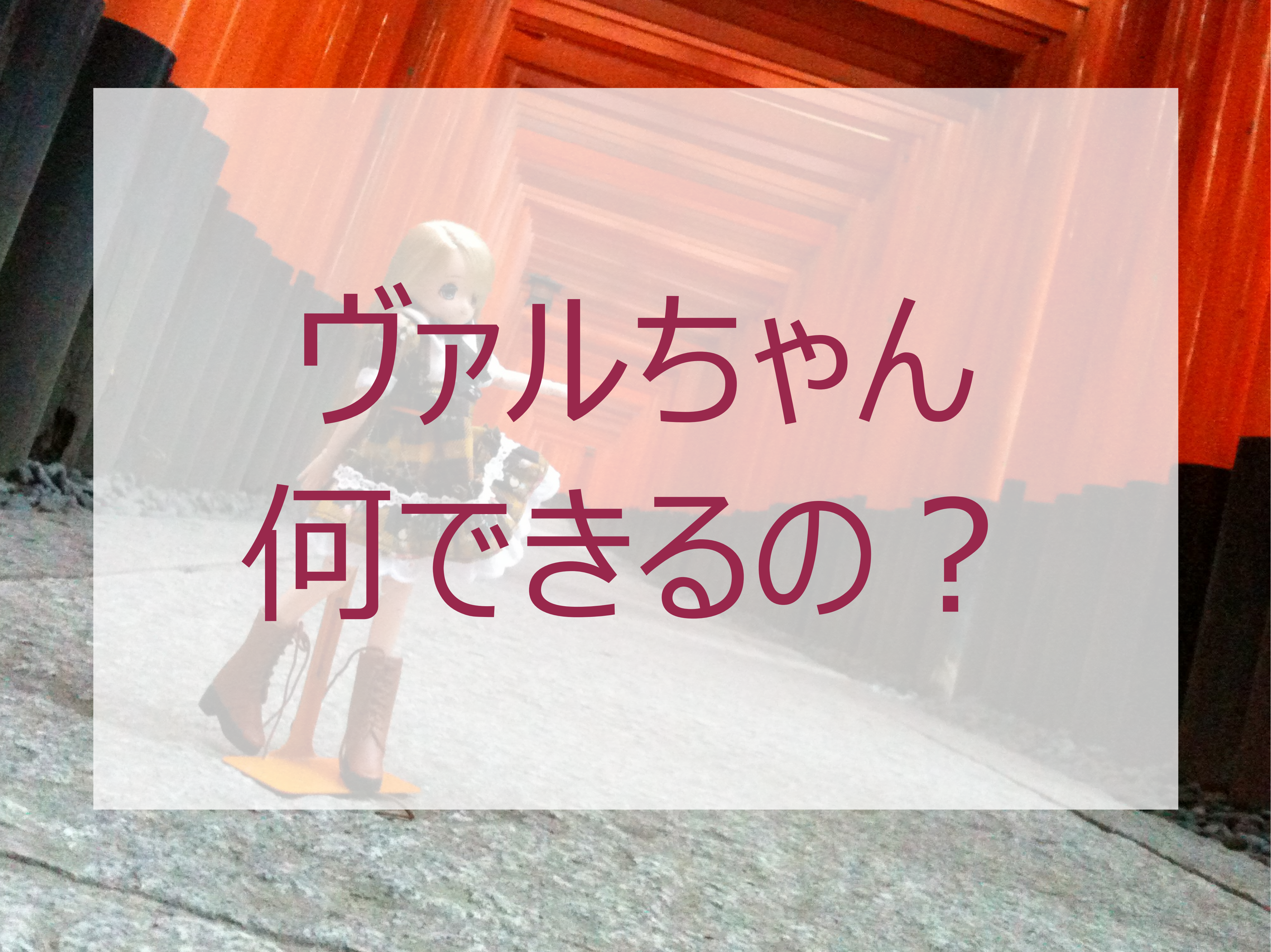

Valgrind

ですよ

valgrind

以下

ヴァルちゃん



ヴァルちゃん
何できるの？

たとえば、

メモリリーク検出

メモリリーク検出（場所付き）

解放済み領域アクセス検出

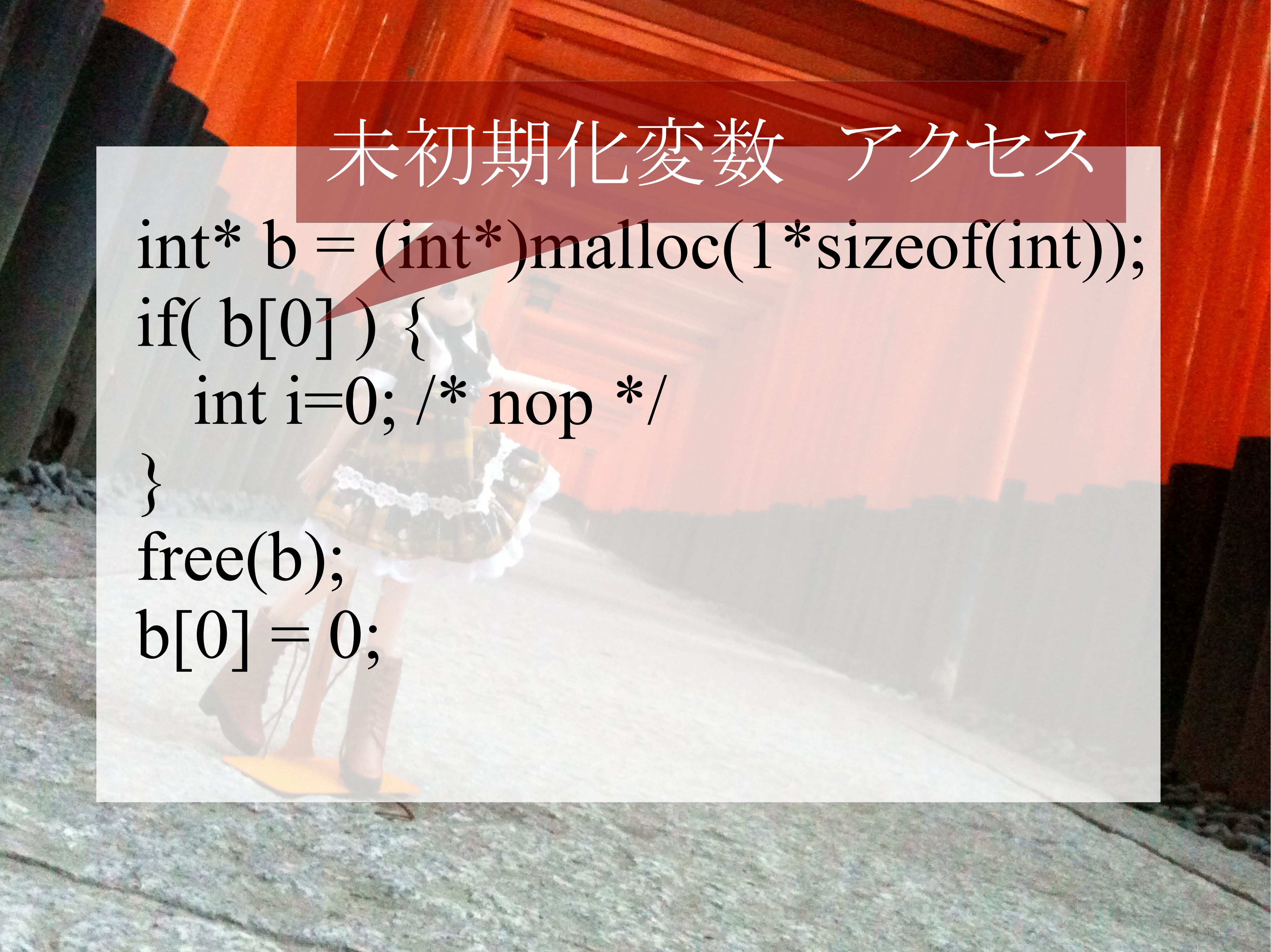
解放済み領域アクセス検出（場所付き）

バッファ
オーバーラン
検出

未初期化メモリへのアクセス検出

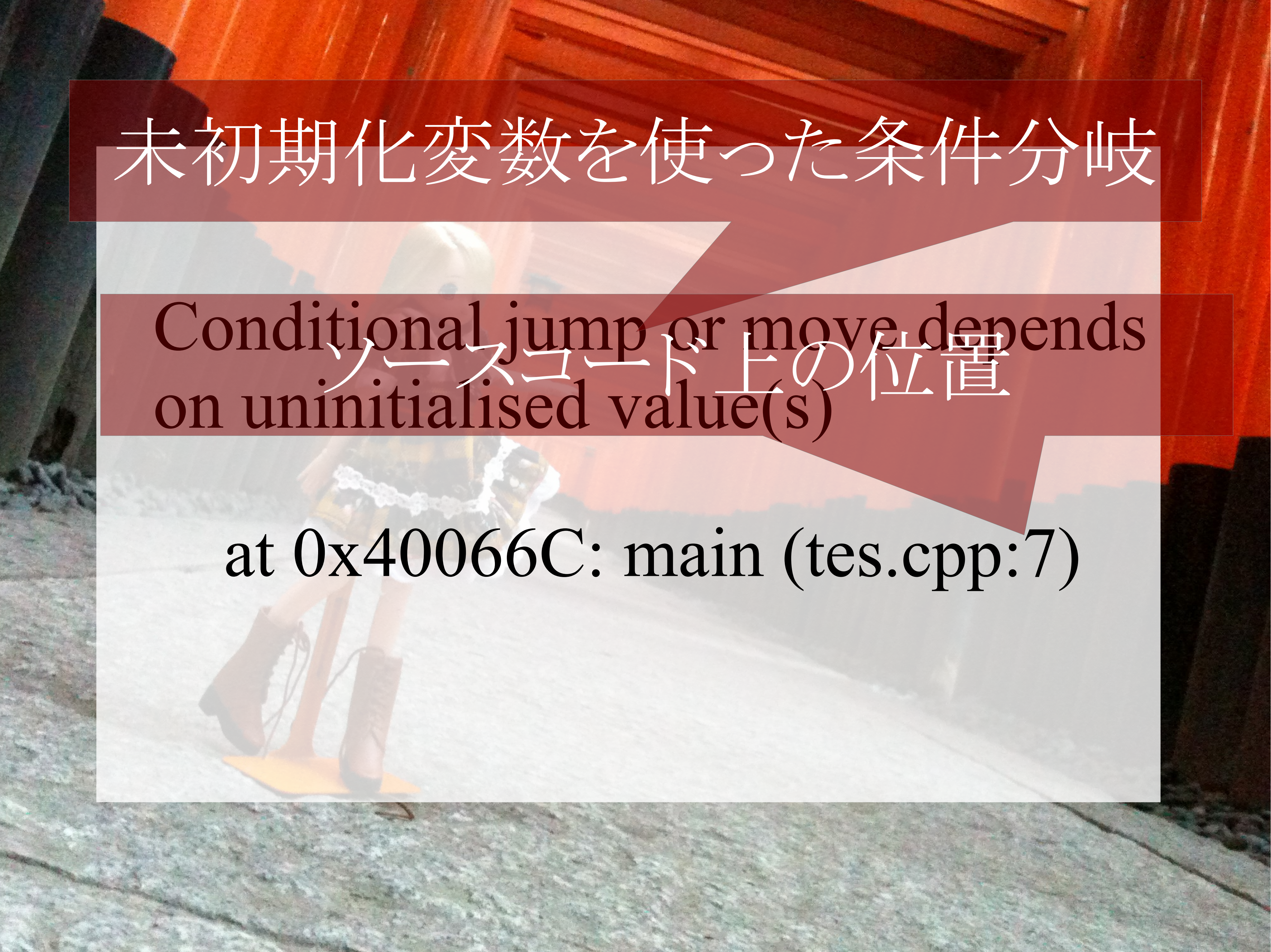
マジで!?

例



未初期化変数　アクセス

```
int* b = (int*)malloc(1*sizeof(int));
if( b[0] ) {
    int i=0; /* nop */
}
free(b);
b[0] = 0;
```

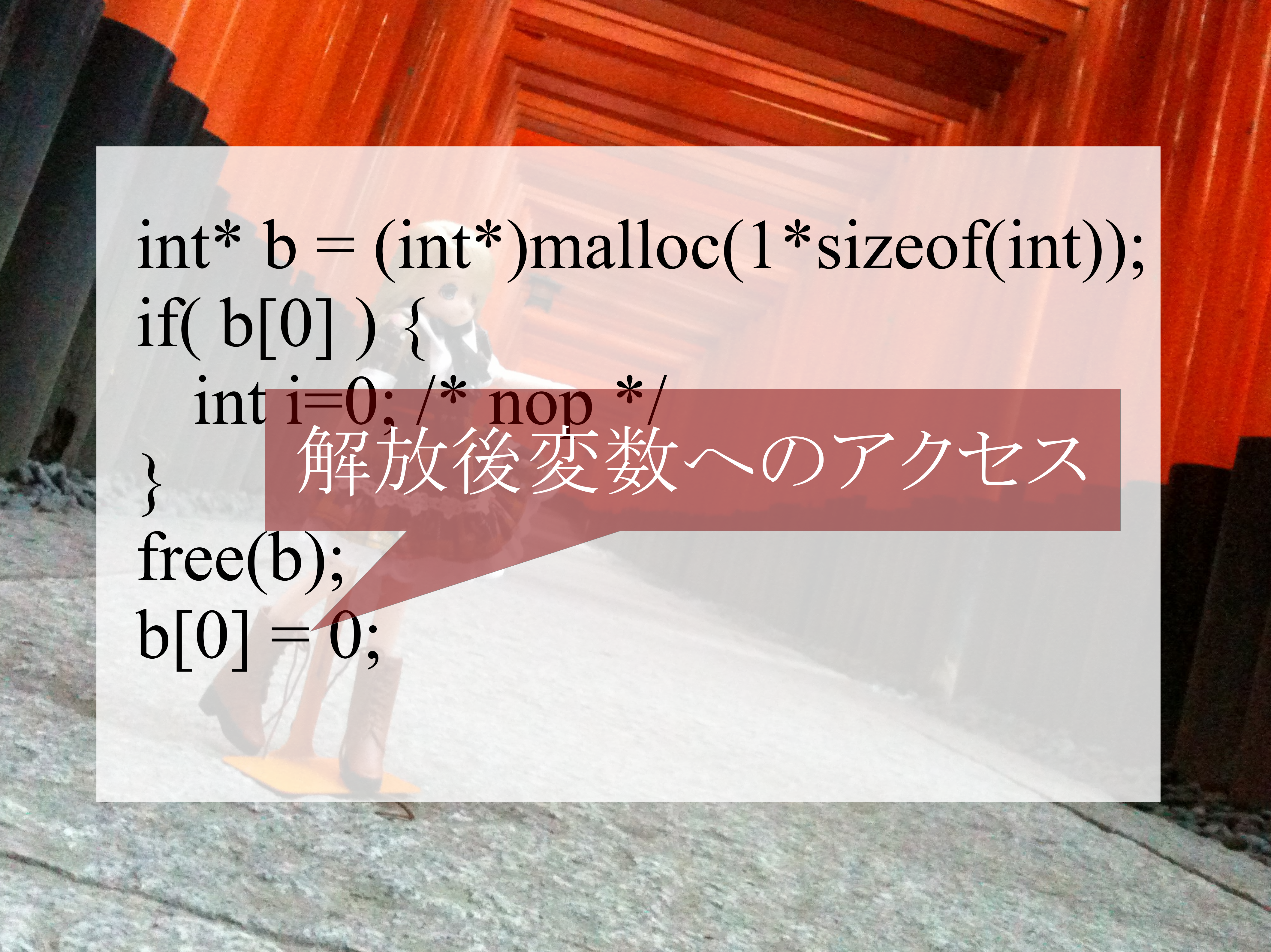


未初期化変数を使った条件分岐

```
Conditional jump or move depends
on uninitialised value(s)
```

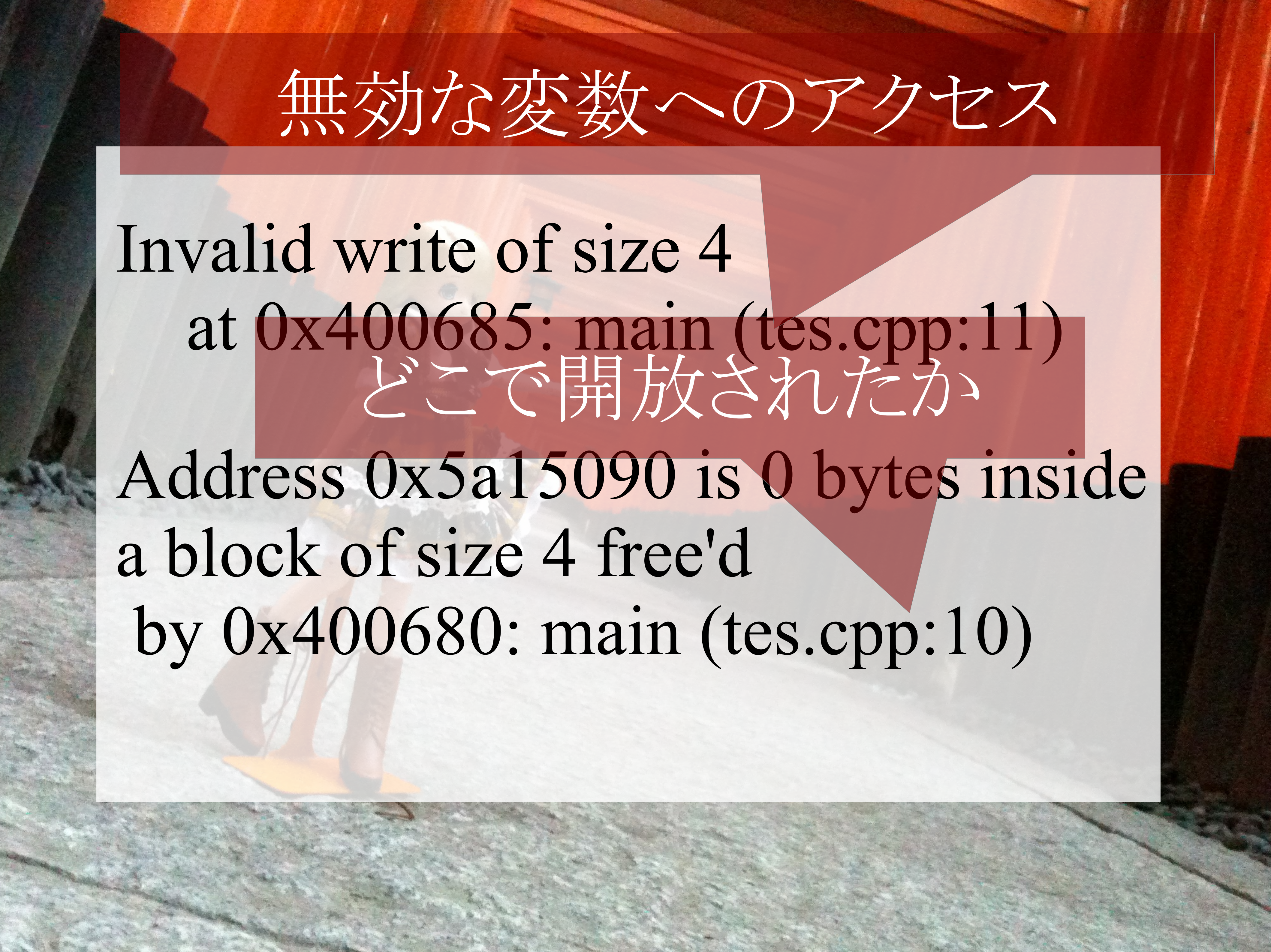
ソースコード上の位置

```
at 0x40066C: main (tes.cpp:7)
```



```
int* b = (int*)malloc(1*sizeof(int));
if( b[0] ) {
   int i=0; /* nop */
}
free(b);
b[0] = 0;
```

解放後変数へのアクセス



無効な変数へのアクセス

```
Invalid write of size 4
    at 0x400685: main (tes.cpp:11)
```

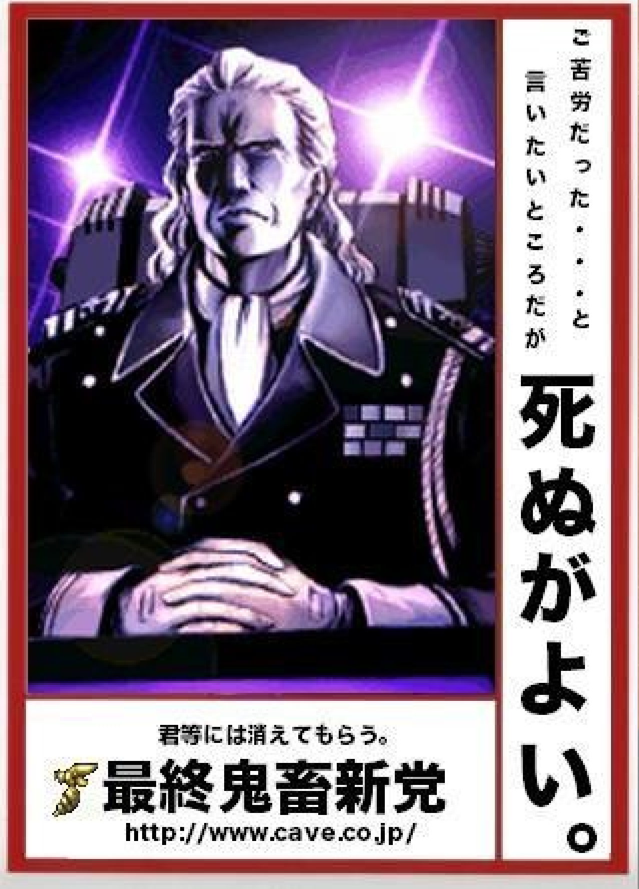
どこで開放されたか

```
Address 0x5a15090 is 0 bytes inside
a block of size 4 free'd
 by 0x400680: main (tes.cpp:10)
```

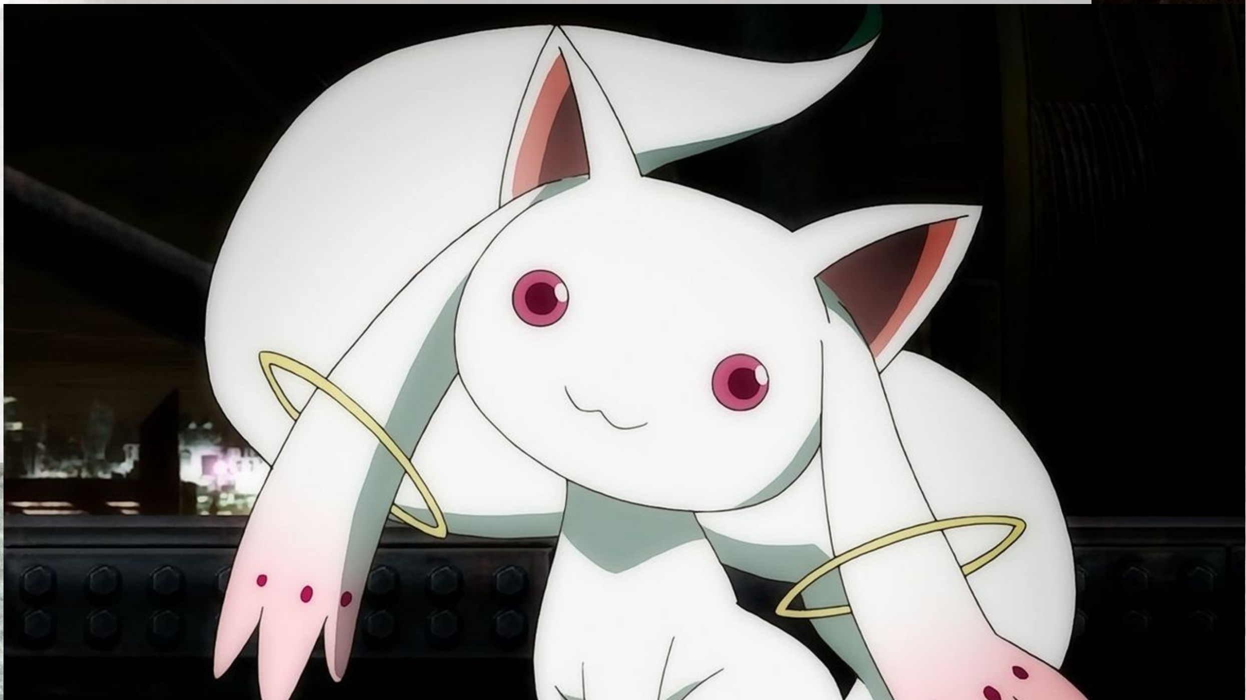
定期的に
ヴァルちゃんして
バグを潰そう♫

きら――ん
完璧だわ!!

と、
言いたい
ところだが

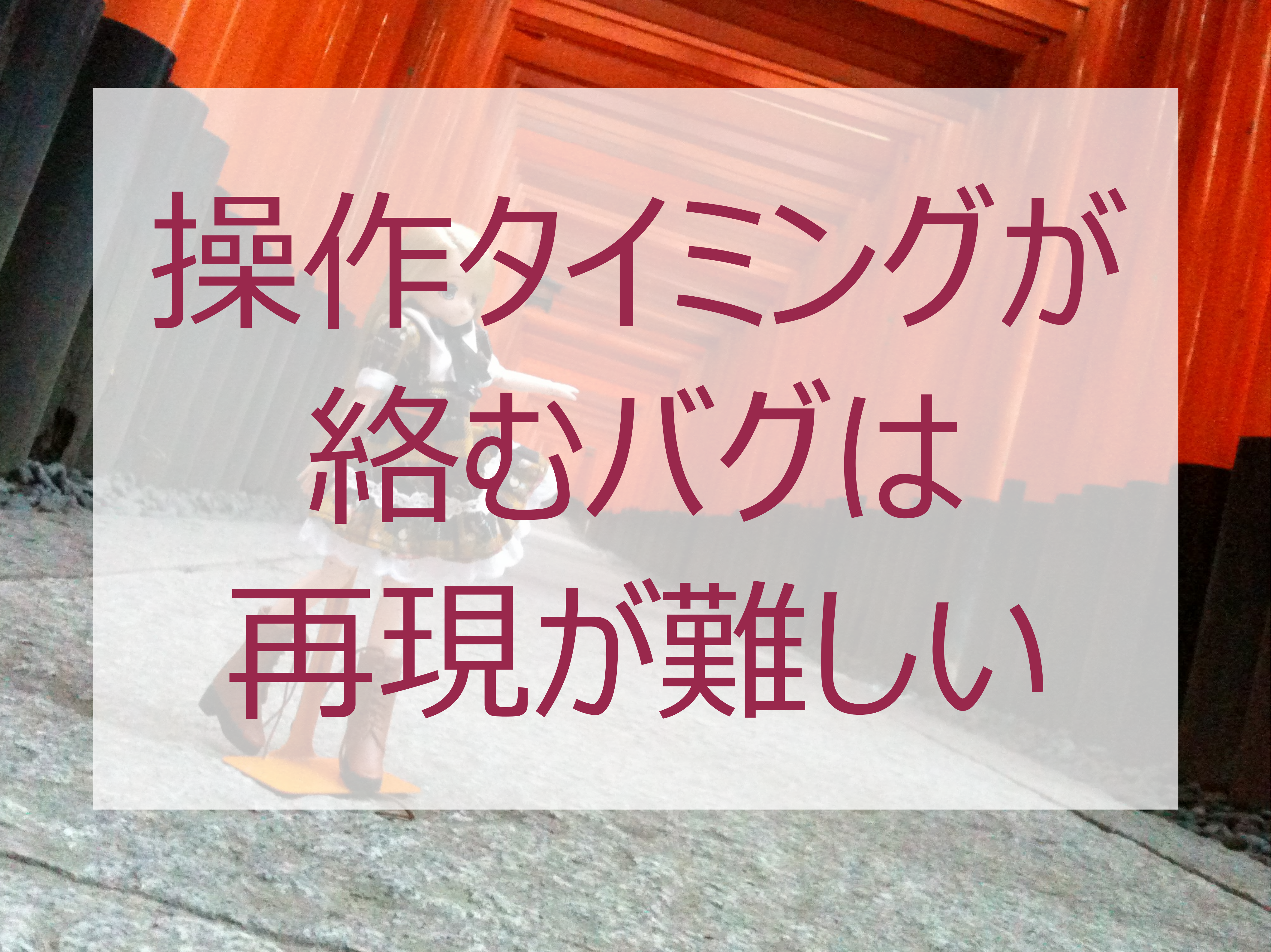

ご苦労だった・・・と
言いたいところだが
死ぬがよい。
君等には消えてもらう。
最終鬼畜新党
http://www.cave.co.jp/

奇跡の代償

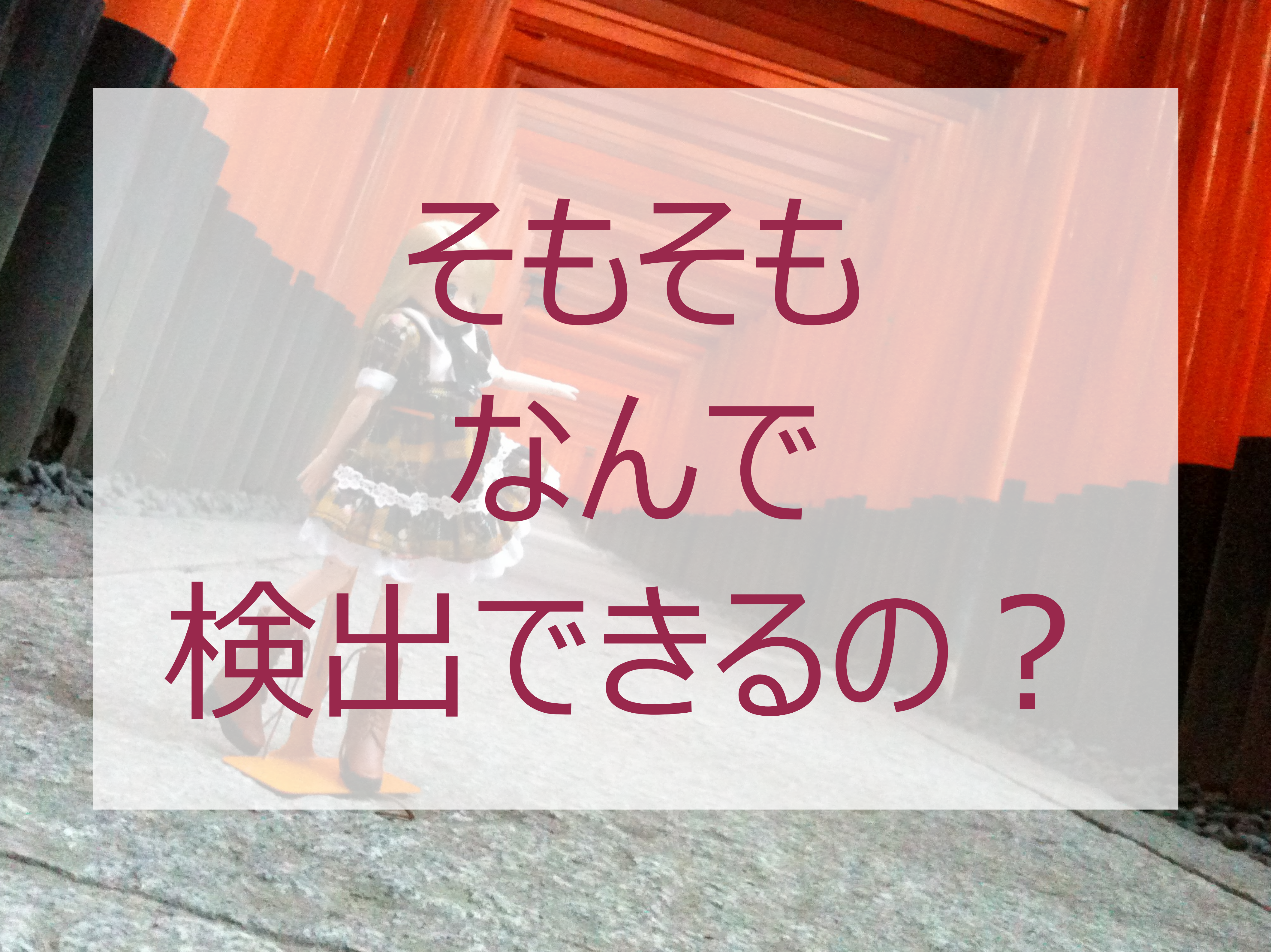


遅い

すごく
遅い



操作タイミングが
絡むバグは
再現が難しい



そもそも
なんで
検出できるの？

x86実行コード

↓

中間形式へ翻訳

全ての
メモリアクセスに
チェックコード挿入

↓

再度x86に
JITコンパイル

遅くない
わけがない

もう一つの問題

誤爆

シングルトン

キャッシュ

メモリーリークとして誤検出

そして

クソblob

はっきり言うと

グラボのドライバ

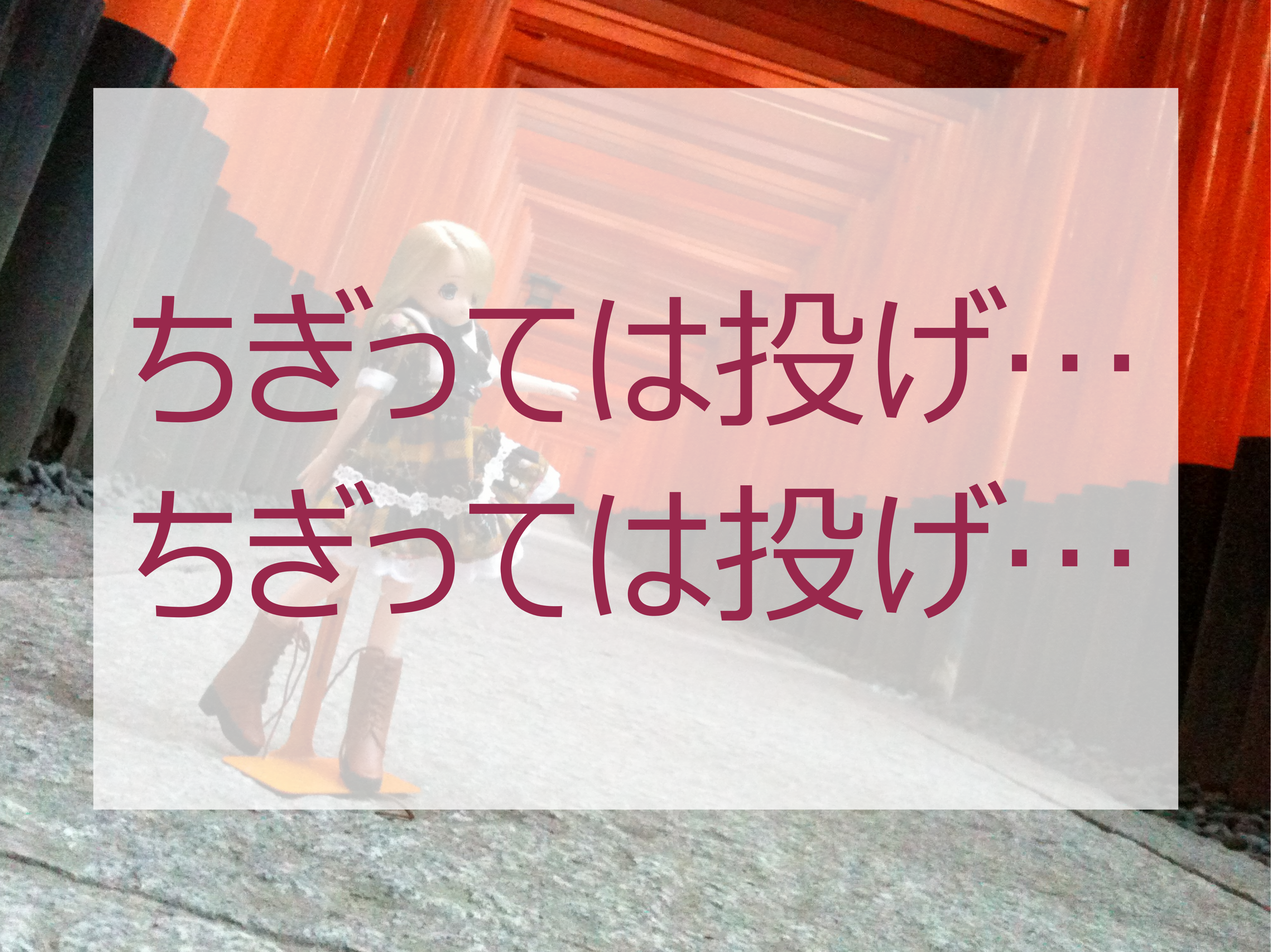
※プライバシー保護のため、モザイク処理を掛けています

クソ**blob**内のクソバグ検出で埋め尽くされる

「無視」
機能はあるので

ひたすら
誤検出を

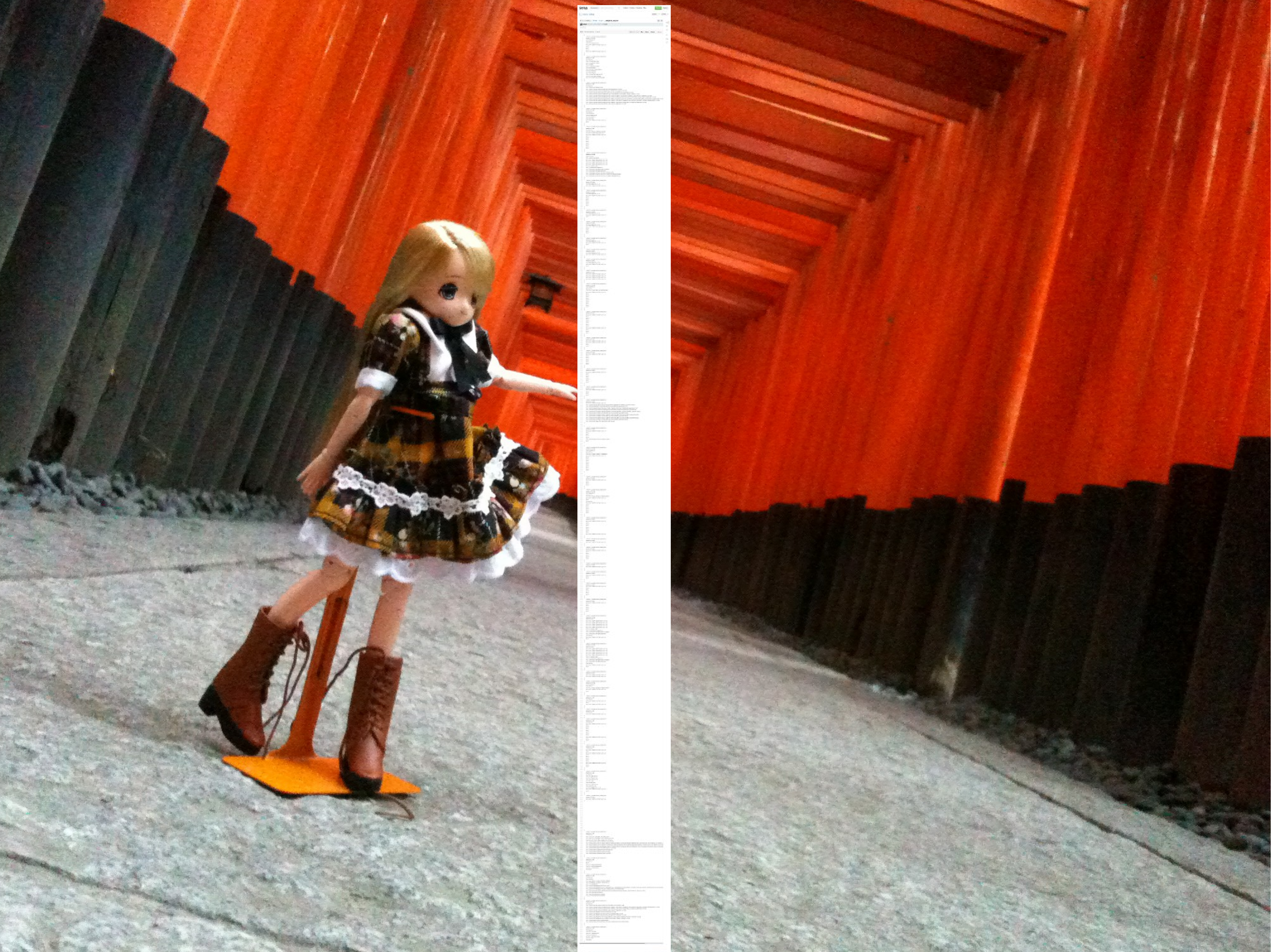
ちぎっては投げ…



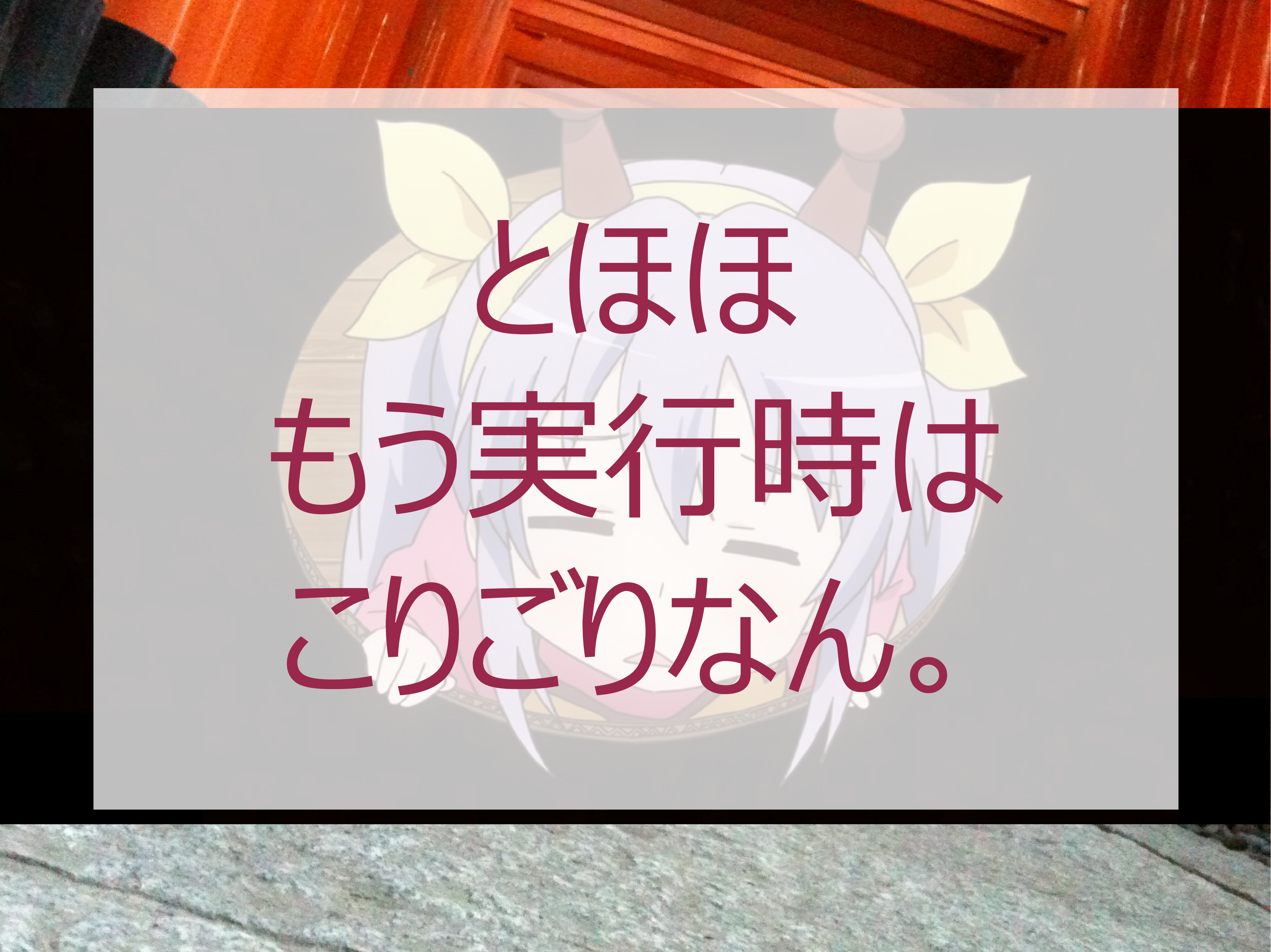
ちぎっては投げ…

ちぎっては投げ…

出来上がった無視ファイル



TOKYO MX



とほほ
もう実行時は
こりごりなん。

終
NHK