

Math::Random::MT::Auto とは

メルセンヌツイスター法による疑似乱数列生成用 [Perl](#) モジュールである。Math::Random::MT から大幅に強化されており、32bit 整数を返す `irand()` や、リスト・配列の並び替えを行う `shuffle()`、正規乱数 `gaussian()`、指数乱数 `exponential()`、ポアソン乱数 `poisson()`、二項乱数 `binomial()` などが実装されている。詳しくは [CPAN のドキュメント](#) を参照。

Math::Random::MT::Auto による `rand()` 関数の置き換え

以下の宣言を `rand()` 関数より前に書くだけ。

```
use Math::Random::MT::Auto 'rand';
```

Math::Random::MT::Auto を使用してマルチスレッドプログラムの子スレッドで利用される乱数列を統一する方法。

```
use strict;
use threads;
use threads::shared;
use Thread::Semaphore;
BEGIN {
    $Math::Random::MT::Auto::shared = 1;
}
use Math::Random::MT::Auto 'rand';
$| = 1;
my $numthreads = 2;

{
    my $semaphore = Thread::Semaphore->new($numthreads);
    for (my $i = 0; $i < 100; $i++) {
        $semaphore->down;
        threads->new(&randomnumber, $i + 1, $semaphore);
    }
    &joinThreads();
}

sub randomnumber {
    my ($i, $semaphore) = @_;
    print($i . ': ' . int(rand(100)) . "\n");
    $semaphore->up;
}

sub joinThreads {
    foreach my $thr (threads->list) {
        if ($thr->tid && !threads::equal($thr, threads->self)) {
            $thr->join;
        }
    }
}
```