

Maple 18 には、Maple の言語とプログラミング機能について、以下の改良点が含まれています。

- [新しい URL パッケージ](#)
- [新しい InertForm パッケージ](#)
- [コンパイラのランタイムライブラリ](#)
- [ランダムツール](#)
- [sort コマンドの改善](#)
- [文字列操作](#)
- [コードエディタの改善](#)
- [インターフェースと Kernelopts](#)
- [entries と indices の indexorder オプション](#)

## ▼ 新しい URL パッケージ

ネットワーク上でのデータのダウンロードとアップロードを可能にする、新しい [URL](#) パッケージが追加されました。URL パッケージでは、http、https、および ftp プロトコルをサポートしています。このパッケージでは、従来 [HTTP](#) パッケージで提供していた機能を拡張しています。

## ▼ 新しい InertForm パッケージ

[InertForm](#) パッケージによって、確実に入力された内容を確認して作業を行うための、簡略化を防いで評価を行うツールを利用できるようになります。

例
$8 \cdot \frac{\sqrt{4}}{4^2} \neq 1$

詳細は、[Maple 18 の InertForm](#) ページを参照してください。

## ▼ コンパイラのランタイムライブラリ

Maple 18 では、コンパイラのランタイムライブラリがスレッドセーフになりました。その結果、スレッドセーフなコンパイル済みプロシージャを複数スレッドで実行できるようになりました。

この操作を行うには、コンパイルする前に `option threadsafe` をプロシージャに追加することで、このプロシージャがスレッドセーフだとコンパイラが認識するようにします。このオプションを追加するだけではスレッドセーフにならないので注意してください。コード作成者は、その責任において、プロシージャがスレッドセーフな形で記述されていることを確認してください。このオプションは、作成されたコードがスレッドセーフであることをコンパイラに知らせるためのものです。

詳細は、[プロシージャのオプション](#)および [Compiler:-Compile](#) を参照してください。

## ▼ ランダムツール

`RandomTools` パッケージで新しい種類の[行列](#)および[ベクトル](#)を使用することで、ランダムな行列およびベクトルを生成できるようになりました。浮動小数および整数の行列およびベクトルを生成する場合、これらの種類は非常に高速です。

さらに、[リスト](#)の種類が更新され、旧バージョンの Maple よりも早く整数と浮動小数のリストが生成されます。

> `RandomTools:-Generate( 'Matrix'( integer, 5, 5 ) );`

$$\begin{bmatrix} -4191464118 & -155509187223 & 410458258399 & -257603501613 & 84608543773 \\ -468934731961 & 263129070744 & 433477878239 & 442400706112 & 108663867602 \\ -47472936313 & -155717390344 & -27268728939 & -467845592812 & -49832113496 \\ -254962893290 & 253906217843 & -264792277647 & 259263763128 & -492986343192 \\ -122890179202 & -172151899801 & -247622562271 & -95359783377 & -290195444176 \end{bmatrix}$$

(1)

> `RandomTools:-Generate( 'Matrix'( polynom( integer( range=0..1), x, degree = 10 ), 4, 4 ) );`

$$\begin{bmatrix} x^{10} + x^9 + x^8 + x^6 + x^4 + x^3 + x^2 + x & x^{10} + x^9 + x^8 + x^3 + x^2 + 1 & x^8 + x^7 + x^5 + x^4 + x^3 + 1 & x^{10} + x^9 + x^5 + 1 \\ x^9 + x^8 + x^7 + x^6 + x^4 + x & x^{10} + x^7 + x^4 + x^2 & x^{10} + x^8 + x^7 + x^6 + x^4 + x^3 + x^2 + 1 & x^{10} + x^9 + x^8 + x^7 + x^5 + x^2 + x + 1 \\ x^{10} + x^8 + x^5 + x + 1 & x^{10} + x^7 + x^5 + x^4 + x^3 + 1 & x^9 + x^7 + x^3 + x + 1 & x^{10} + x^7 + x^3 + x^2 + x \\ x^{10} + x^9 + x^7 + x^6 + x^5 + x^3 + 1 & x^{10} + x^9 + x^8 + x^7 + x^5 + x^4 + x^3 & x^{10} + x^8 + x^6 + x^4 + x & x^7 + x^6 + x^5 + x^4 \end{bmatrix}$$

(2)

> `RandomTools:-Generate( 'Vector'( float, 10, datatype = float[8] ) );`

```
[ 0.156554315800872
  0.468594073777602
  0.897276711016091
  0.366764328242085
  0.169191454027561
  0.155696424011674
  0.419796233035937
  0.539905541967967
  0.511839347390428
  0.896408024321359 ]
```

(3)

```
> RandomTools:-Generate( 'Vector[row]'( float(digits=4), integer(range=5..10) ) );
```

```
[ 0.05206 0.3084 0.005301 0.0001805 0.0007295 0.02227 0.004261 0.7816 0.0005706 ]
```

(4)

```
> time[real]( RandomTools:-Generate( list( integer, 106 ) ) );
```

0.136

(5)

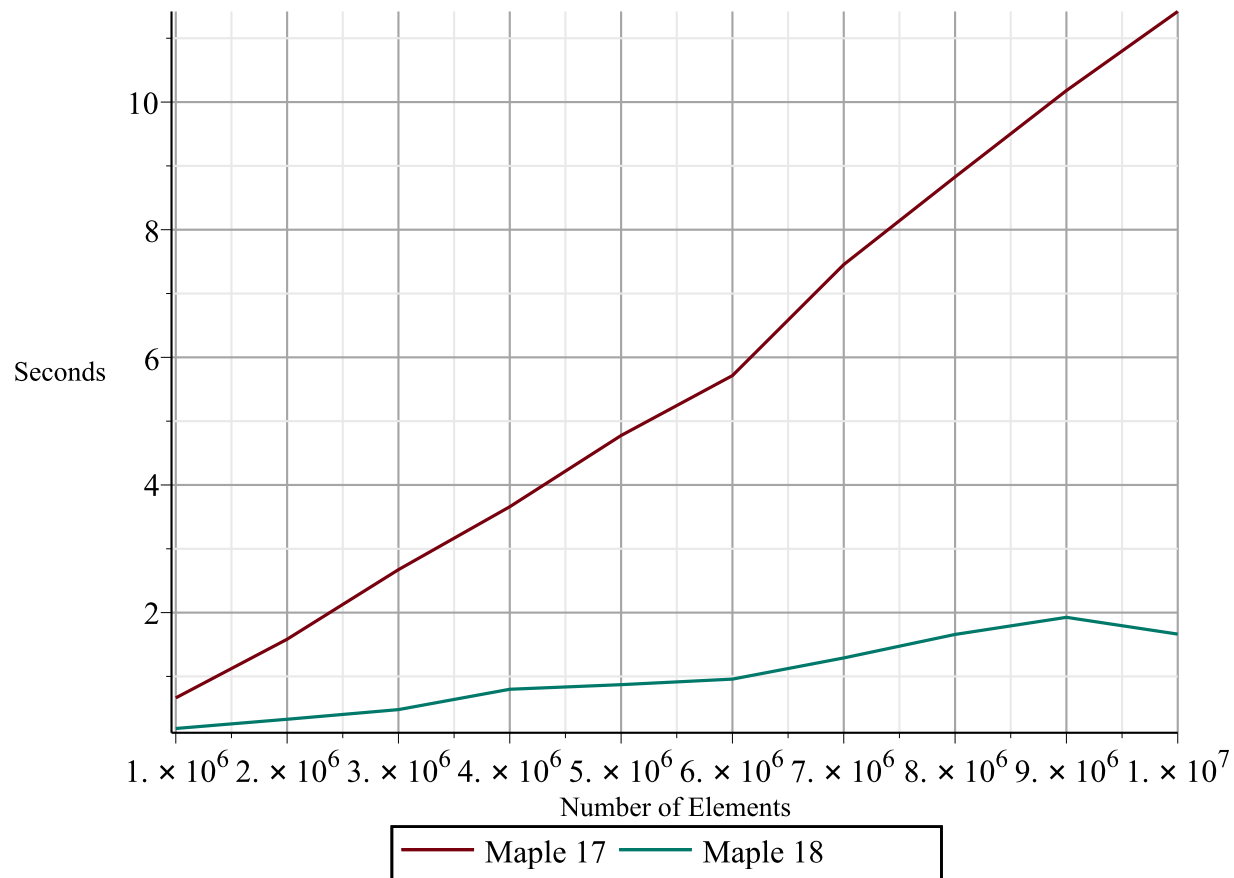
旧バージョンの Maple では、この作業を完了するまでに 5 秒以上かかります。

## ▼ sort コマンドの改善

Maple 18 で改善された sort コマンドにより、自動並列実行や新しいキーソートに加え、折り紙付きの安定ソートによる、より早いソートを行うことができます。

## ▼ 並列ソート

Maple 18 では、[sort](#) コマンドでマルチコアを活用することができます。以下のグラフでは、Maple 17 と Maple 18 で実行されたランダム順列のソートのタイミングの差を示しています。



並列ソートはすべての Maple の組み込みソートアルゴリズムで使われており、自動的にコンピュータ上で利用可能なすべてのコアを活用します。カスタム比較機能がある場合、ソートは直列で行われます。

## ▼ キーソート

Maple 18 では、キーソートをサポートしています。キーソートによって、ユーザはキーに要素をマッピングする関数を指定することができます。入力は、対応するキーをソートすることでソートされます。たとえば、以下のリストでは、各サブリストの最初の要素を比較することでソートできます。

```
> sort( [ [9], [4], [7], [2], [5], [8], [3], [6], [1] ], key = (x → x[1]) );
[[1], [2], [3], [4], [5], [6], [7], [8], [9]]
```

(6)

旧バージョンの Maple では、カスタム比較関数でこれを行うことができます。キーソートではキー関数を  $O(n)$  回だけコールするのに対し、比較関数では  $O(n \cdot \log(n))$  回コールされます。このため、キーソートのほうが早くソートすることができます。さらに、デフォルトの Maple 比較関数で基本的なソート操作を実行した場合、ソートを並列で行うことができます。

以下の例では、比較関数でソートした場合とキー関数でソートした場合の時間の差 (秒) を示します。

```
> L := [ seq( [i], i in combinat:-randperm( 3·105 ) ) ]:
> time[real]( sort( L, (x, y) → x[1] ≤ y[1] ) );
11.040
```

(7)

```
> time[real]( sort( L, key = (x → x[1]) ) );
1.003
```

(8)

## ▼ 安定ソート

Maple 18 のソート機能は安定しています。安定ソート機能では、入力と等しい要素が相対的に同じ順序で出力に残されていることを保証します。これは、カスタム比較機能またはキーソート機能を使用してソートする場合に非常に重要です。以下のリストは、リストの最初の要素に従ってソートされています。最初の要素が等しいリストは、入力リストで指定されているとおり、相対的に同じ順序になります (2 番目の要素を参照)。

```
> sort( [[1, 1], [2, 1], [3, 1], [1, 2], [2, 2], [3, 2], [1, 3], [2, 3], [3, 3]], (x, y) → (x[1] ≤ y[1]) );
[[1, 1], [1, 2], [1, 3], [2, 1], [2, 2], [2, 3], [3, 1], [3, 2], [3, 3]]
```

(9)

安定ソートは、すべての内蔵された Maple ソートアルゴリズム、および [ソート](#) ドキュメントに示されたガイドラインに従うすべてのカスタム比較機能で利用できます。デフォルトで、比較機能は厳密ではない (同等か小さい) 必要があります。厳密な比較機能 (小さい) を使用する場合は、**厳密なオプション** を使用して指定する必要があります。

```
> sort( [ [1, 1], [2, 1], [3, 1], [1, 2], [2, 2], [3, 2], [1, 3], [2, 3], [3, 3] ], strict = ((x, y) → x[1] < y[1]) );
[[1, 1], [1, 2], [1, 3], [2, 1], [2, 2], [2, 3], [3, 1], [3, 2], [3, 3]]
```

(10)

## ▼ 文字列操作

[StringTools](#) パッケージには、3 つの新しいコマンドがあります。

- [DifferencePositions](#) : 文字列を文字のリストに変換します。
- [IsDerangement](#) : 文字列が別の文字列の完全順列であるか検証します。
- [Snarf](#) : 文字列から特定の文字から成る接頭辞を抽出します。

```
> StringTools:-DifferencePositions( "edit", "tide" )
```

1, 2, 3, 4

(11)

```
> StringTools:-IsDerangement( "edit", "tide" )
```

true

(12)

```
> StringTools:-Snarf( "tiDE", 'lower' );
```

"ti"

(13)

## ▼ コードエディタの改善

コードエディタが使いやすく改善されました。詳細は、[埋め込みコンポーネントの改良](#)を参照してください。

## ▼ インターフェースと Kernelopts

新しいインターフェース変数 `worksheetdir` から、アクティブなワークシートのディレクトリへのパスが返されます。詳細は、[インターフェース](#)を参照してください。

2 つの新しいカーネルオプション `gctotaltime` および `gctotaltime[real]` から、ガベージコレクションの処理にかかる時間が、CPU 時間および実際の時間で返されます。詳細は、[kernelopts](#) を参照してください。

## ▼ entries と indices の indexorder オプション

[entries](#) と [indices](#) プロシージャでは、新しいオプション `indexorder` を使用しています。indexorder オプションによって、返された要素がインデックスの値でソートされます。テーブルのインデックスに基づいてテーブル全体をソートする `entries` コマンドに最も有用です。例

```
> t := table( [1. = "one", 2. = "two", 3. = "three", 4. = "four", 5. = "five", 6. = "six", 7. = "seven", 8. = "eight", 9. = "nine", 10. = "ten" ] );
```

```
t := table([2. = "two", 6. = "six", 7. = "seven", 5. = "five", 3. = "three", 10. = "ten", 4. = "four", 1. = "one", 8. = "eight", 9. = "nine"]);
```

 (14)

```
> entries(t);
```

```
["two"], ["six"], ["seven"], ["five"], ["three"], ["ten"], ["four"], ["one"], ["eight"], ["nine"]
```

 (15)

```
> entries(t, indexorder);
```

```
["one"], ["two"], ["three"], ["four"], ["five"], ["six"], ["seven"], ["eight"], ["nine"], ["ten"]
```

 (16)

## ▼ 参照

[Maple 18 の新機能](#)

## ▶ Pages That Link to This Page