

# Maple 14 におけるプログラミングおよび接続性に対する変更

Maple 14 にはプログラミング機能および他ツールへの接続性に対して以下の強化がなされています。

## ▼ バイト配列

- 配列で元のデータ型と異なるデータ型を表示できるように [ArrayTools\[Alias\]](#) コマンドが拡張されました。 [readbytes](#) と併用して混合形式のバイナリファイルを読むときに便利です。たとえば、 [readbytes](#) を使用してファイルを整数 1 バイトの配列に読み込みます。先頭 100 バイトがファイルのヘッダーであると判明したら、残りのバイトは倍精度 (float[8]) の浮動小数点数として表示できます。

## ▼ データ圧縮

- [StringTools](#) パッケージにはデータ圧縮関連の新規コマンドが 2 つ ([Compress](#) および [Uncompress](#)) 追加されました。これらのコマンドは文字列やバイト配列を含むデータの集合に対して zlib 圧縮または解凍を実行します。

## ▼ ベンチマーク

- [CodeTools](#) パッケージには新規のコマンド [Usage](#) が追加されました。このコマンドは、 [time](#) に似ており、コマンドの実行にかかった時間とメモリ使用量を測定しますが、出力がより柔軟に制御できるようになっています。

```
> with(CodeTools):  
> Usage(ifactor(32!+1));  
memory used=10.11MiB, alloc change=8.12MiB, cpu time=0.17s, real time=  
0.18s  
      (61146083) (652931) (2889419049474073777) (2281) (3.1)
```

```
> Usage(ifactor(33!+1),output='all');  
Record(realtime=0.191, cputime=0.183, bytesused=12125768, bytesalloc=9959648,  
output=(67) (143446529) (175433) (101002716748738111) (50989)) (3.2)
```

- [CodeTools\[CPUTime\]](#) および [CodeTools\[RealTime\]](#) は、それぞれオプション `output=[cputime,output]` および `output=[realtime,output]` が指定されている [Usage](#) の呼出手順のショートカットとして使用できます。

```
> CPUTime(ifactor(34!+1));  
0.003, (4379593820587205958191075783529691) (67411) (3.3)
```

## ▼ ListTools

- [ListTools](#) パッケージには新規のコマンド [FindMaximalElement](#)、 [FindMinimalElement](#)、 [SelectFirst](#) および [SelectLast](#) が追加されました。これらは一般的なリスト操作を行うタスクをコマンド化したものです。

```
> with(ListTools):  
> SelectFirst(type, [1., 1/2, 1, 5.1, 2, 11], integer);  
1 (4.1)
```

```
> FindMaximalElement([[1,a,4],[2,b,3],[3,c,2],[4,a,1]], (x,y)  
->lexorder(x[2],y[2]));  
[3, c, 2] (4.2)
```

## MATLAB からの行列のインポートおよび MATLAB への行列のエクスポート

- [ImportMatrix](#) および [ExportMatrix](#) コマンドは MATLAB のバイナリファイルをインポートおよびエクスポートできるように変更されました。データ圧縮なしの MATLAB バージョン 6 とデータ圧縮ありの MATLAB バージョン 7 のファイルがサポートされています。

## コマンドラインバージョンの Maple 向け ExcelTools

- コマンドラインバージョンの Maple 環境で作業しているときも、スタンダードワークシートインターフェイスを動かす必要なく Excel のスプレッドシートからデータを取り込んだり、`rtable` のデータを Excel のスプレッドシートにファイル変換することができるようになりました。ExcelTools パッケージの使用に関する詳細については、[ExcelTools](#) を参照してください。

## インプレース (上書き) 代入

- `subs` コマンドはインプレースで配列や行列、ベクトルに代入できるようになりました。この機能は、コマンド名にインデックスとして `inplace` を追加して指定します。

```
> A := <w,x;y,z>;
```

$$A := \begin{bmatrix} w & x \\ y & z \end{bmatrix} \quad (7.1)$$

```
> subs[inplace](x=Pi,A);
```

$$\begin{bmatrix} w & \pi \\ y & z \end{bmatrix} \quad (7.2)$$

```
> A;
```

$$\begin{bmatrix} w & \pi \\ y & z \end{bmatrix} \quad (7.3)$$

## 匿名プロシージャに対するサポートの向上

- 再帰呼び出しを容易にする、新しい特別パラメータ `thisproc` がプロシージャ内から利用できるようになりました。現在実行中のプロシージャ名を指す `procname` と異なり、`thisproc` はそのプロシージャ自体を指すため、匿名プロシージャ内から再帰呼び出しを行うために使用できます。
- 新しいプロシージャオプション `option procname` を指定すると、そのプロシージャの呼び出し元プロシージャの名前を引き継ぎ、エラー報告を行うことができます。すなわち、入れ子になっている内のプロシージャで発生した例外を外側の包含しているプロシージャで発生したかのように見せることができます。

## タスクプログラミングモデル

- [Task Programming Model](#) には新しい関数が追加されました。タスク内で `Return` 関数を使用すると、タスクモデルの実行を中断し、`Start` から特定の値を返すことができます。

## 外部呼び出しにおけるタスクモデル

- 外部コードから [Task Programming Model](#) が利用できるように外部呼び出しの API に関

数が追加されました。新しく利用可能になった関数は次の 4 つで

す：[MapleStartRootTask](#)、[MapleCreateContinuationTask](#)、[MapleStartChildTask](#)、[MapleTaskReturn](#)

## ▼ RunWorksheet

- 新規コマンド [RunWorksheet](#) が [DocumentTools](#) パッケージに追加されました。このコマンドはワークシートをプロシージャであるかのように実行します。ワークシートはディスク上に保存し、引数は決まった仕組みを使用して渡す必要があります。ワークシートはヘッドレス (headless ; ユーザーインターフェイスなし) かつモダル (modal ; 呼び出されたワークシートの実行が終了するまで呼び出し元のワークシートにコントロールが戻らない) モードで実行されます。また、独自のカーネルで実行されます。
- Maple 14 ではこの機能はまだ実験的であり、将来のリリースではその設計が変わっている可能性があります。

## ▼ 関連項目

[Index of New Maple 14 Features, Language and System Updates in Maple 14](#)