Preliminaries

Created using Maple 14.01
Jake Bobowski

It's a good idea to start your Maple code using *restart*. This statement clears all variables and allows you to start re-evaluating your code from scratch. I always like to include the date at the top of the page. This can be achieved using *FormatTime*( ), but this command requires that you first load the *StringTools* package using the command *with*(*StringTools*). If you don't first include *with*(*StringTools*) the *FormatTime* command will not work.

```
> restart;
  FormatTime;
```

FormatTime **(1)**

```
> with(StringTools);
  FormatTime( );
```

[*Anagrams, AndMap, ApproximateSearch, ApproximateSearchAll, ArithmeticMean, Border, BorderArray, BorderLength, CamelCase, Capitalize, CaseJoin, CaseSplit, Center, Centre, Char, CharacterFrequencies, CharacterMap, Chomp, Chop, CommonPrefix, CommonSuffix, Compare, CompareCI, Compress, CountCharacterOccurrences, Decode, Delete, DeleteSpace, Drop, EditDistance, Encode, Entropy, Escape, Exchange, ExpandCharacterClass, ExpandTabs, Explode, Fence, Fibonacci, Fill, FirstFromLeft, FirstFromRight, FormatMessage, FormatTime, FromByteArray, Generate, GenerateIdentifier, Group, HammingDistance, HammingSearch, HammingSearchAll, Has, HasASCII, HasAlpha, HasAlphaNumeric, HasBinaryDigit, HasControlCharacter, HasDigit, HasGraphic, HasHexDigit, HasIdentifier, HasIdentifier1, HasLower, HasOctalDigit, HasPrintable, HasPunctuation, HasSpace, HasUpper, HasVowel, Hash, Implode, Indent, IndexOfCoincidence, Insert, I, IsASCII, IsAlpha, IsAlphaNumeric, IsAnagram, IsBalanced, IsBinaryDigit, IsConjugate, IsControlCharacter, IsDigit, IsEodermdrome, IsGraphic, IsHexDigit, IsIdentifier, IsIdentifier1, IsLower, IsMonotonic, IsOctalDigit, IsPalindrome, IsPeriod, IsPermutation, IsPrefix, IsPrimitive, IsPrintable, IsPunctuation, IsSorted, IsSpace, IsSubSequence, IsSuffix, IsUpper, IsVowel, Join, Kasiski, LeftFold, LeftRecursivePathOrder, Length, LengthSplit, Levenshtein, LexOrder, LongestCommonSubSequence, LongestCommonSubString, LowerCase, LyndonFactors, Map, MatchFence, MaxChar, MaximalPalindromicSubstring, Metaphone, MinChar, MinimumConjugate, MonotonicFactors, NGrams, NthWord, OrMap, Ord, OtherCase, Overlap, PadLeft, PadRight, ParseTime, PatternCanonicalForm, PatternDictionary, PatternEquivalent, Period, Permute, PrefixDistance, PrimitiveRoot, Random, Randomize, Readability, RegMatch, RegSplit, RegSub, RegSubs, Remove, Repeat, Repeats, RevLexOrder, Reverse, RightFold, RightRecursivePathOrder, Rotate, Search, SearchAll, Select, SelectRemove, Sentences, Shift, ShortLexOrder, ShortRevLexOrder,*

SimilarityCoefficient, Sort, SortPermutation, Soundex, Split, Squeeze, Stem, StringBuffer, SubString, Substitute, SubstituteAll, SuffixDistance, Support, SyllableLength, Tabulate, Take, ThueMorse, ToByteArray, Trim, TrimLeft, TrimRight, Uncompress, Unique, UpperCase, Visible, WildcardMatch, WordContaining, WordCount, WordEnd, WordStart, Words, WrapText]

"2012-07-30" **(2)**

Maple commands are terminated using a semicolon. Notice that the *with*(*StringTools*) command outputs a bunch of useless text. To suppress this output, terminate the *with*(*StringTools*) using a colon instead of a semicolon. The colon can be used to suppress the output of any Maple command.

```
> with(StringTools):
  FormatTime();
```

"2012-07-30" **(3)**

Syntax is important. The most common problems are spelling mistakes or typographical errors. Here's a spelling mistake. The resulting error message is cryptic.

```
> restart;
  with(StingTools):
Error, invalid input: with expects its 1st argument, pname, to be of type {`module`, package}, but received StingTools
```

Here's a typographical error. A space has been inserted between the *with* and the (*StringTools*) where there should not be one. Now there is no error message! Maple is intrepretting this input as the variable *with* multiplied by the variable *StringTools*. When we try to execute *FormatTime*( ) it will fail because *with*(*StringTools*) has yet to be successfully loaded after the last *restart*.

```
> with (StringTools);
```

with StringTools **(4)**

```
> FormatTime();
```

FormatTime( ) **(5)**

Arguments can be used within the brackets of *FormatTime*( ) to modify the format of the output. The %m, %d, %Y, %H, %m represent the month, day, year, hour, and minutes respectively.

```
> with(StringTools):
  FormatTime("%m−%d−%y, %H:%m");
```

"07-30-12, 22:07" **(6)**

Notice that multiple lines of code can be associated with each Maple input >, this is often very convenient as it keeps code compact. To create a new line without executing the previous lines of code, press *shift* + *enter*. Pressing enter on its own will execute all commands associated with the current Maple input. So rahter than doing...

```
> x := 2;
```

x := 2 **(7)**

```
> y := 3;
```

y := 3 **(8)**

```
> z := x + y;
```

z := 5 **(9)**

... a more compact block of code can be used:

```
> x := 2 :
  y := 3 :
  z := x + y;
```

$$z := 5 \quad \textbf{(10)}$$

Finally, notice that in this Maple worksheet text has been used to add comments to the Maple code throughout. This is an extremely good practice to adopt. To insert text into you code select **Insert⇒ Text**. Alternatively, click on the "T" symbol on the menu bar at the top of the screen. The text you enter can be formatted using the usual attributes. To insert another line of Maple code, selected **Insert⇒ Maple Input**. Alternatively, click on the "[>" symbol on the menu bar at the top of the screen.

```
>
```