It is sometimes useful to store the bytes contained in a string into a byte wave so commands that work on waves can be used to analyze the contents. One example is determining the frequency of each letter in a piece of text for a linguistic analysis or analysis of DNA/RNA/protein sequences. Another use is to convert the bytes of a downloaded string into a numeric wave. Igor provides the **StringToUnsignedByteWave** and **WaveDataToString** functions to make facilitate such analyses. These functions were added in Igor Pro 9.00.

Examples can be found at https://www.wavemetrics.com/code-snippet/working-binary-string-data-examples.

Regular Expressions

A regular expression is a pattern that is matched against a subject string from left to right. Regular expressions are used to identify lines of text containing a particular pattern and to extracts substrings matching a particular pattern.

A regular expression can contain regular characters that match the same character in the subject and special characters, called "metacharacters", that match any character, a list of specific characters, or otherwise identify patterns.

The regular expression syntax is based on PCRE — the Perl-Compatible Regular Expression Library.

Igor syntax is similar to regular expressions supported by various UNIX and POSIX `egrep(1)` commands.

See **Regular Expressions References** on page IV-194 for details on PCRE.

Regular Expression Operations and Functions

Here are the Igor operations and functions that work with regular expressions:

Grep

The **Grep** operation identifies lines of text that match a pattern.

The subject is each line of a file or each row of a text wave or each line of the text in the clipboard.

Output is stored in a file or in a text wave or in the clipboard.

Grep Example

```
Function DemoGrep()
   Make/T source={"Monday","Tuesday","Wednesday","Thursday","Friday"}
   Make/T/N=0 dest
   Grep/E="sday" source as dest           // Find rows containing "sday"
   Print dest
End
```

The output from Print is:

```
   dest[0]= {"Tuesday","Wednesday","Thursday"}
```

GrepList

The **GrepList** function identifies items that match a pattern in a string containing a delimited list.

The subject is each item in the input list.

The output is a delimited list returned as the function result.

GrepList Example

```
Function DemoGrepList()
   String source = "Monday;Tuesday;Wednesday;Thursday;Friday"
   String dest = GrepList(source, "sday") // Find items containing "sday"
```