C: Regular expressions (RE)

REs are a **very** powerful **pattern-matching** tools for manipulating text.

Some frequently used **single-character** REs:

RE	What it matches
\s	Any whitespace character: space, tab, newline, return, etc.
\S	Any character that is NOT whitespace
\d	Any digit: 0-9
\D	Any character that is NOT a digit
\w	Any word character: letters, digits, underscore
\W	Any character that is NOT a word character
.	Any character except a newline
^	Just before the first character of the string
$	Just after the last character of the string

Can be combined to match a **sequence** of characters:

RE	What it matches
\d\d	Any two digits
\w\w\w	Any three word characters
\w\S	A word character followed by any non-whitespace character
^\w	A word character at the beginning of the string

Can match **sequences of varying** length:

RE	What it matches
\d+	One or more digits
\w+	One or more word characters

\s+	One or more whitespace characters

Characters **without a backslash** match themselves:

RE	What it matches
a\d	Letter 'a' followed by any digit: a0 … a9

Can be combined with a vertical bar to match **either left OR right** RE:

RE	What it matches
\d|\s	A digit or a space
a|\d	Letter 'a' or a digit

In code, REs usually written in **strings prefixed with r** to avoid excess \'s

```
new = old.str.replace( r'\D', '', regex=True )
```

Would remove any non-digit from the string.

This just scratches the surface:

REs can be used for very complex pattern matching and substitution

Continue with g12 demo.py