Regular expression

Why regular expressions?

A **regular expression** (**regex** or **regexp** for short) is a sequence of characters that describes a common **pattern** for a set of strings. Such patterns can be used to search, edit, and manipulate texts.

They can either check if a whole string or its substring matches the given pattern or replace the substring with another one.

When do we need such patterns?
Say we want to obtain all the files with the same extension (like .pdf), or extract all the entries of a particular name in different forms (for example, Edgar Poe, Edgar Allan Poe, E. A. Poe, etc.),
all email addresses, or even find all numeric structures denoting dates (02/03/2020). With regexps, such tasks can be done in one line.

https://regex101.com/
To play with regex... choose **PCRE** as the flavour. It
means **P**erl **C**ompatible **R**egular **E**xpressions which are the most common standard in practice.

In regular expressions, the case of characters is relevant: *park* is not the same as *PARK*, i.e., they do not match.

In addition, let's consider another sequence of characters *PAKR*. It does not match our pattern since the two characters are in the wrong (reverse) order.

The power of regular expressions
.The real power of regular expressions comes when you start using special metacharacters called **wildcards**.
They allow you to define a pattern, so you can match strings that do not necessarily contain an identical sequence of characters. You can skip some characters in a string or match different characters in the same positions, or even repeat a character several times.

The dot character
The **dot** character . matches any single character including letters, digits, and so on, except for the newline character, unless it is specified.

The question mark

The **question mark** ? is a special character that means “the preceding character or nothing". The question mark ? signals that the character before it

can occur once or zero times in a string to match the pattern. When can we use it?

The word *ARK* does not match the .ARK pattern. But if we add ? right after the dot character .?ARK, the word *ARK* will match the new pattern since the first character is optional now.