**regular expression**[**¶**](http://localhost:8888/notebooks/notes%20in%20ipynb%20format/regular%20expression.ipynb#regular-expression)

A regular expression (regex or regexp) is a sequence of characters that define a search pattern, usually used for string matching within text. Regular expressions are widely used in programming, text processing tasks, and search engines for tasks such as data validation, text manipulation, and pattern matching.

Here are some common symbols and usage patterns in regular expressions:

**Literal Characters:**

* Regular characters such as letters, digits, and special characters match themselves.
* Example: The regular expression cat matches the string "cat".

**Character Classes:**

* A character class matches any one of a set of characters enclosed in square brackets [ ].
* Example: The regular expression [aeiou] matches any vowel.

**Quantifiers:**

* These specify the number of occurrences of a character or group in a pattern.
* \*: Matches the preceding element zero or more times.
* +: Matches the preceding element one or more times.
* ?: Matches the preceding element zero or one time.
* {n}: Matches exactly n occurrences of the preceding element.
* {n,}: Matches n or more occurrences of the preceding element.
* {n,m}: Matches at least n and at most m occurrences of the preceding element.

**Anchors:**

* Anchors are used to specify the position of a match within a string.
* ^: Matches the beginning of the string.
* $: Matches the end of the string.
* \b: Matches a word boundary.

**Alternation:**

* The pipe symbol | is used to specify alternative matches.
* Example: cat|dog matches either "cat" or "dog".

**Grouping and Capturing:**

* Parentheses () are used to group and capture sub-patterns.
* Example: (abc)+ matches "abc", "abcabc", "abcabcabc", etc.
* Escaping Special Characters: Special characters like . (dot) or \* (asterisk) can be escaped with a backslash \ to match them literally.

Regular expressions provide a powerful and flexible way to search, match, and manipulate text based on patterns. However, they can be complex and may require practice to master effectively. There are various implementations of regular expressions across programming languages and tools, but the basic principles remain consistent.

Let's go through some examples of regular expressions along with explanations:

**Matching a Specific Word:**

* Pattern: cat
* Explanation: This matches the word "cat" exactly.
* Example: The regular expression cat matches "cat" in the string "The cat is black."

**Matching Any Vowel:**

* Pattern: [aeiou]
* Explanation: This matches any vowel character.
* Example: The regular expression [aeiou] matches "a" in the string "apple".

**Matching a Range of Characters:**

* Pattern: [a-z]
* Explanation: This matches any lowercase letter.
* Example: The regular expression [a-z]+ matches "hello" in the string "hello world".

**Matching Zero or More Occurrences:**

* Pattern: a\*
* Explanation: This matches zero or more occurrences of the character "a".
* Example: The regular expression ba\* matches "b", "ba", "baa", etc. in the string "baaab".

**Matching One or More Occurrences:**

* Pattern: b+
* Explanation: This matches one or more occurrences of the character "b".
* Example: The regular expression b+ matches "b", "bb", "bbb", etc. in the string "bbab".

**Matching Zero or One Occurrence:**

* Pattern: colou?r
* Explanation: This matches "color" or "colour".
* Example: The regular expression colou?r matches both "color" and "colour".

**Matching Exactly n Occurrences:**

* Pattern: a{3}
* Explanation: This matches exactly three occurrences of the character "a".
* Example: The regular expression ba{3} matches "baaa" in the string "baaaab".

**Matching Word Boundaries:**

* Pattern: \bcat\b
* Explanation: This matches the word "cat" with word boundaries.
* Example: The regular expression \bcat\b matches "cat" in the string "A cat is running."

**Matching Beginning and End of String:**

* Pattern: ^start
* Explanation: This matches "start" only at the beginning of the string.
* Example: The regular expression ^start matches "start" in the string "start here" but not in "please start".

**Using Alternation:**

* Pattern: cat|dog Explanation: This matches either "cat" or "dog".
* Example: The regular expression cat|dog matches "cat" in the string "The cat is black." and "dog" in "The dog is brown."
* These examples illustrate how regular expressions can be used to match patterns within text data, enabling powerful text processing capabilities in various programming languages and tools.

**Examples:**

**'info@example.org' > pattern = r'[\w.-]+@[a-z]+\.org'**

**'john.doe@example.com' >** **pattern = r"[\w\.-]+@[a-z]+\.[a-z]{2,}"**

**'1234-567890' > pattern = r"\d{4}-\d{6}"**

**'555-123-456' > pattern = r"\d{3}-\d{3}-\d{3}"**

**‘1885-1900’ Pattern = r'(\b\d{4}-\d{4}\b)'**

**‘1900’ pattern = r’(\b\d{4})**