**Regular Expression Practice Questions**

Question 1- Write a RegEx pattern in python program to check that a string contains only a certain set of characters (in this case a-z, A-Z and 0-9).

Question 2- Write a RegEx pattern that matches a string that has an a followed by zero or more b's

Question 3- Write a RegEx pattern that matches a string that has an a followed by one or more b's

Question 4- Write a RegEx pattern that matches a string that has an a followed by zero or one 'b'.

Question 5- Write a RegEx pattern in python program that matches a string that has an a followed by three 'b'.

Question 6- Write a RegEx pattern in python program that matches a string that has an a followed by two to three 'b'.

Question 7- Write a Python program that matches a string that has an 'a' followed by anything, ending in 'b'.

Question 8- Write a RegEx pattern in python program that matches a word at the beginning of a string.

Question 9- Write a RegEx pattern in python program that matches a word at the end of a string.

Question 10- Write a RegEx pattern in python program to find all words that are 4 digits long in a string.

Sample text- '01 0132 231875 1458 301 2725.'

Expected output- ['0132', '1458', '2725']

**REGULAR EXPRESSIONS**

**1. `+` (One or More):**

* Usage: `+` matches one or more occurrences of the preceding character or group.
* Example: `a+` matches one or more consecutive "a" characters. For instance, it matches "a," "aa," "aaa," and so on.
* Use case: When you want to find patterns with at least one occurrence of a character or group, such as matching one or more digits in a number (e.g., `\d+` for matching numbers like "123" or "42").

**2. `\*` (Zero or More):**

* Usage: `\*` matches zero or more occurrences of the preceding character or group.
* Example: `ab\*` matches "a" followed by zero or more "b" characters. It matches "a," "ab," "abb," "abbb," and so on.
* Use case: When you want to find patterns with optional occurrences of a character or group, like matching variations of a word that may have extra characters (e.g., `colou?r` matches "color" and "colour").

**3. `$` (End of Line):**

* Usage: `$` represents the end of a line or string.
* Example: `end$` matches "end" only when it appears at the end of a line or string.
* Use case: When you want to anchor a pattern to the end of a line or string, ensuring that it doesn't match if the specified characters are followed by anything else. For example, matching the end of a sentence or a filename extension (e.g., `\.txt$` matches strings ending with ".txt").

These symbols are essential for controlling the number of occurrences and the position of patterns within text when using regular expressions. They provide flexibility in matching specific patterns in a wide range of scenarios.

You can use the following Python program with a regex pattern that matches a word at the beginning of a string:

```python

import re

pattern = r'^\w+'

# Example usage

test\_string = "Hello, world!"

match = re.search(pattern, test\_string)

if match:

print("Match found:", match.group())

else:

print("No match found.")

```

In this pattern:

- `^` represents the start of the string, indicating that the match should occur at the beginning of the string.

- `\w+` matches one or more word characters (letters, digits, or underscores) while ‘.+’ matches one or more characters (including non-word characters) at the start of a string.

The + quantifier allows the pattern to continue matching characters as long as they exist.

So, this pattern will match the word "Hello" in the example string "Hello, world!" because it appears at the beginning of the string.