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).

Answer - Imagine you have a magic code to check if a string has only three types of letters and numbers: small letters (a-z), big letters (A-Z), and numbers (0-9). Here's how you can use it:

In [3]:
```
import re

# Magic code
magic_code = r'^[a-zA-Z0-9]+$'

# Your secret message
message = "Hello123"

# Use the magic code to check
if re.match(magic_code, message):
    print("Your message contains only normal letters and numbers.")
else:
    print("Your message has some other tricky characters.")
```

```
Your message contains only normal letters and numbers.
```

Think of the "magic code" as the rules that say, "Only small letters, big letters, and numbers are allowed." You give it a message, and it tells you if the message follows these rules.

If it says, "Your message contains only normal letters and numbers," that means your message is made up of the right letters and numbers. If it says, "Your message has some other tricky characters," that means there are strange characters in your message.

So, it's like a magic detective for letters and numbers in your messages!

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

Answer - Imagine you have a special code to find a specific pattern in words. The pattern you want to find is an 'a' followed by some 'b's (or maybe no 'b's at all). Here's how you can use your special code:

In [4]:
```
import re

# Special code
special_code = r'ab*'

# Your word to check
word = "abb"

# Use the special code to check
if re.search(special_code, word):
    print("Your word has an 'a' followed by some 'b's or just an 'a'.")
else:
    print("Your word doesn't have that pattern.")
```

```
Your word has an 'a' followed by some 'b's or just an 'a'.
```

Think of the "special code" as a detective tool. You give it a word, and it tells you if that word has an 'a' followed by some 'b's (or maybe just an 'a'). If it says, "Your word has an 'a' followed by some 'b's or just an 'a'," it means the word matches your pattern. If it says, "Your word doesn't have that pattern," it means the word is different.

So, it's like your secret word detective, and it's good at finding words with 'a' and some 'b's or just 'a'!

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

Answer - Imagine you have a special code to find a specific pattern in words. The pattern you want to find is an 'a' followed by some 'b's, but there must be at least one 'b'. Here's how you can use your special code:

In [5]:
```
import re

# Special code
special_code = r'ab+'

# Your word to check
word = "abb"

# Use the special code to check
if re.search(special_code, word):
    print("Your word has an 'a' followed by one or more 'b's.")
else:
    print("Your word doesn't have that pattern.")
```

```
Your word has an 'a' followed by one or more 'b's.
```

Think of the "special code" as a detective tool. You give it a word, and it tells you if that word has an 'a' followed by some 'b's, but there must be at least one 'b'. If it says, "Your word has an 'a' followed by one or more 'b's," it means the word matches your pattern. If it says, "Your word doesn't have that pattern," it means the word is different.

So, it's like your secret word detective, and it's good at finding words with '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'.

Answer - To create a regex pattern that matches a string with an 'a' followed by zero or one 'b', you can use the following pattern:

In [6]:
```
import re

# Define the regex pattern
pattern = r'ab?'

# Input string to check
input_string = "ab"

# Use re.search() to check if the pattern is found in the string
if re.search(pattern, input_string):
    print("The string contains 'a' followed by zero or one 'b'.")
else:
    print("The string doesn't match the pattern.")
```

```
The string contains 'a' followed by zero or one 'b'.
```

Explanatio terms :

The pattern variable is set to r'ab?', which is a regular expression pattern. 'a' matches the character 'a' exactly. 'b?' matches zero or one occurrence of the character 'b'. We provide an input_string that you want to check. We use re.search(pattern, input_string) to look for the pattern in the input string. If the pattern is found, it means the string contains an 'a' followed by either zero 'b's or just one 'b', and we print "The string contains 'a' followed by zero or one 'b'." If not, we print "The string doesn't match the pattern." So, the pattern ab? looks for an 'a' followed by either no 'b' or just one 'b'!

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

Answer - To create a regex pattern that matches a string with an 'a' followed by exactly three 'b's, you can use the following pattern:

In [7]:
```
import re

# Define the regex pattern
pattern = r'ab{3}'

# Input string to check
input_string = "abbbb"

# Use re.search() to check if the pattern is found in the string
if re.search(pattern, input_string):
    print("The string contains 'a' followed by three 'b's.")
else:
    print("The string doesn't match the pattern.")
```

```
The string contains 'a' followed by three 'b's.
```

Explanation terms:

The pattern variable is set to r'ab{3}', which is a regular expression pattern. 'a' matches the character 'a' exactly. 'b{3}' means exactly three occurrences of the character 'b'. We provide an input_string that you want to check. We use re.search(pattern, input_string) to look for the pattern in the input string. If the pattern is found, it means the string contains an 'a' followed by exactly three 'b's, and we print "The string contains 'a' followed by three 'b's." If not, we print "The string doesn't match the pattern." So, the pattern ab{3} looks for an 'a' followed by three 'b's!

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

Answer - Imagine you have a special rule for words. You want to find words that have an 'a' followed by either two or three 'b's. Here's how you can do it:

In [9]:
```
import re

# Special rule
special_rule = r'ab{2,3}'

# Your word to check
word = "abbb"

# Use the special rule to check
if re.search(special_rule, word):
    print("Your word has an 'a' followed by two or three 'b's, like 'abb' or 'abbb'.")
else:
    print("Your word doesn't have that pattern.")
```

```
Your word has an 'a' followed by two or three 'b's, like 'abb' or 'abbb'.
```

Think of the "special rule" as a detective's guideline. You give it a word, and it tells you if that word has an 'a' followed by either two or three 'b's, like "abb" or "abbb." If it says, "Your word has an 'a' followed by two or three 'b's, like 'abb' or 'abbb'," it means the word matches your special rule. If it says, "Your word doesn't have that pattern," it means the word is different.

So, it's like having a detective rule to find words with 'a' followed by either two or three 'b's in them!

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

Answer - Imagine you have a magic rule for words. You want to find words that start with 'a', have anything in the middle (it could be any letters or even nothing), and end with 'b'. Here's how you can do it:

In [10]:
```
import re

# Magic rule
magic_rule = r'a.*b$'

# Your word to check
word = "axxxxxb"

# Use the magic rule to check
if re.search(magic_rule, word):
    print("Your word starts with 'a', has anything in between, and ends with 'b'.")
else:
    print("Your word doesn't have that pattern.")
```

```
Your word starts with 'a', has anything in between, and ends with 'b'.
```

Think of the "magic rule" as a detective's instruction. You give it a word, and it tells you if that word starts with 'a', has anything in the middle (it can be anything), and ends with 'b'. If it says, "Your word starts with 'a', has anything in between, and ends with 'b'," it means the word matches your magic rule. If it says, "Your word doesn't have that pattern," it means the word is different.

So, it's like having a magic detective to find words that begin with 'a', have something in the middle, and finish with 'b'!

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

Answer - Imagine you have a special rule for finding the first word in a sentence. Here's how it works:

In [1]:
```
import re

# Special rule
special_rule = r'^\w+'

# Your sentence to check
sentence = "Hello World"

# Use the special rule to find the first word
match = re.search(special_rule, sentence)

if match:
    matched_word = match.group()
    print("The first word in the sentence is:", matched_word)
else:
    print("There is no word at the beginning of the sentence.")
```

```
The first word in the sentence is: Hello
```

Think of the "special rule" as a detective's instruction. You give it a sentence, and it tells you the first word in that sentence. If it finds a word at the beginning, it tells you what it is. If there's no word at the start, it tells you that too.

So, it's like having a detective who finds the very first word in a sentence for you!

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

Answer - Imagine you want to find the last word in a sentence. Here's how you can do it with a special rule:

In [2]:
```
import re

# Special rule
special_rule = r'\w+$'

# Your sentence to check
sentence = "Hello World"

# Use the special rule to find the last word
match = re.search(special_rule, sentence)

if match:
    matched_word = match.group()
    print("The last word in the sentence is:", matched_word)
else:
    print("There is no word at the end of the sentence.")
```

```
The last word in the sentence is: World
```

Think of the "special rule" as a detective's instruction. You give it a sentence, and it tells you the last word in that sentence. If it finds a word at the end, it tells you what it is. If there's no word at the end, it tells you that there's nothing there.

So, it's like having a detective who finds the very last word in a sentence for you!

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']
Answer - To find all words that are 4 digits long in a string using a Python program and regular expressions, you can use the following code:

In [3]:
```
import re

# Define the regex pattern to match four digits in a row
pattern = r'\b\d{4}\b'

# Input string to search for words
input_string = '01 0132 231875 1458 301 2725.'

# Use re.findall() to find all matching words in the input string
matched_words = re.findall(pattern, input_string)

# Print the matched words
print(matched_words)
```

```
['0132', '1458', '2725']
```

Explanation in simple terms:

The pattern variable is set to r'\b\d{4}\b', which is a regular expression pattern.

\b represents a word boundary, ensuring that we match complete words. \d matches any digit. {4} means exactly four occurrences of the previous pattern (four digits). \b again represents a word boundary. We provide an input_string that contains words with varying numbers of digits.

We use re.findall(pattern, input_string) to find all words that match the pattern in the input string.

The result, stored in matched_words, contains all the words that are exactly 4 digits long.

So, when you run this code, it will find and print the words with 4 digits in a list, giving you the expected output: ['0132', '1458', '2725'].

In [ ]:
```

```