

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:** # Define the regular expression pattern

```
pattern = "[a-zA-Z0-9]+$"
```

# Function to check if a string matches the pattern

```
def contains_only_allowed_characters(input_string):
```

```
    return re.match(pattern, input_string) is not None
```

# Test the function

```
input_string = "Hello123"
```

```
if contains_only_allowed_characters(input_string):
```

```
    print(f'{input_string}' contains only allowed characters.")
```

```
else:
```

```
    print(f'{input_string}' contains characters other than 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

**Answer:** pattern = r'a\*bb\*'

So, this pattern will match strings like "a", "ab", "abb", "abbb", and so on, where "a" is 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

**Answer:** pattern = r'ab+'

So, this pattern will match strings like "a", "ab", "abb", "abbb", and so on, where "a" is followed by zero 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:** pattern = r'ab?'

So, this pattern will match strings like "a" and "ab," where "a" is 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'.

**Answer:** pattern = r'abbb'

So, this pattern will match strings like "abbb," where "a" is followed by exactly 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:** pattern = r'ab{2,3}'

So, this pattern will match strings like "abb" and "abbb," where "a" is followed by two or three "b"s.

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

**Answer:** import re

```
# Define the regular expression pattern
pattern = r'a.*b$'

# Function to check if a string matches the pattern
def matches_pattern(input_string):
    return re.match(pattern, input_string) is not None

# Test the function
input_string = "axxxxxb"

if matches_pattern(input_string):
    print(f'{input_string} matches the pattern.')
else:
    print(f'{input_string} does not match the pattern.')
```

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

**Answer:** import re

```
# Define the regular expression pattern
pattern = r'^\w+'

# Function to find and return the word at the beginning of a string
def find_word_at_beginning(input_string):
    match = re.match(pattern, input_string)

    if match:
        return match.group()
    else:
```

```

        return None

# Test the function

input_string = "Hello, world!"

result = find_word_at_beginning(input_string)

if result:

    print(f"The word at the beginning is: {result}")

else:

    print("No word found at the beginning.")

```

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

**Answer:** import re

```

# Define the regular expression pattern

pattern = r'\w+$'

# Function to find and return the word at the end of a string

def find_word_at_end(input_string):

    match = re.search(pattern, input_string)

    if match:

        return match.group()

    else:

        return None

# Test the function

input_string = "Hello, world!"

result = find_word_at_end(input_string)

if result:

    print(f"The word at the end is: {result}")

else:

    print("No word found at the end.")

```

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:**

```
import re

# Define the regular expression pattern
pattern = r'\b\d{4}\b'

# Function to find all 4-digit words in a string
def find_four_digit_words(input_string):
    return re.findall(pattern, input_string)

# Test the function
input_string = '01 0132 231875 1458 301 2725.'
result = find_four_digit_words(input_string)

print(result) # Output: ['0132', '1458', '2725']
```