Double-click (or enter) to edit

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

```
import re

def contains_only_allowed_characters(input_string):
    pattern = r'^[a-zA-Z0-9]+$'
    if re.match(pattern, input_string):
        return True
    else:
        return False

input_string = input("Enter a string: ")
if contains_only_allowed_characters(input_string):
    print("The string contains only allowed characters.")
else:
    print("The string contains characters other than a-z, A-Z, and 0-9.")
```

```
 Enter a string: ython program to check that a string contains only a certain set of character
 The string contains characters other than a-z, A-Z, and 0-9.
```

2. Write a Python program that matches a string that has an a followed by zero or more b's.

```
import re

def match_pattern(input_string):
    pattern = r'ab*'

    if re.search(pattern, input_string):
        return True
    else:
        return False

# Test the function
input_string = input("Enter a string: ")
if match_pattern(input_string):
    print("The string matches the pattern 'a' followed by zero or more 'b's.")
else:
    print("The string does not match the pattern.")
```

```
 Enter a string: Replace only the first occurrence of 5 with five for the given string
 The string matches the pattern 'a' followed by zero or more 'b's.
```

3. Replace only the first occurrence of 5 with five for the given string

```
input_string = "This is a sample string with 5 apples, and 5 oranges."

# Replace the first occurrence of '5' with 'five'
output_string = input_string.replace('5', 'five', 1)

print(output_string)
```

```
 This is a sample string with five apples, and 5 oranges.
```

4. Write a Python program that matches a string that has an a followed by three 'b'.

```
import re

def match_pattern(input_string):
    pattern = r'ab{3}'

    if re.search(pattern, input_string):
        return True
```

```
    else:
        return False

# Test the function
input_string = input("Enter a string: ")
if match_pattern(input_string):
    print("The string matches the pattern 'a' followed by three 'b's.")
else:
    print("The string does not match the pattern.")
```

```
Enter a string: Write a Python program that matches a string that has an a followed abb
The string does not match the pattern.
```

5. Write a Python program that matches a string that has an 'a' followed by anything ending in 'b'.

```
import re

def match_pattern(input_string):
    pattern = r'a.*b$'

    if re.search(pattern, input_string):
        return True
    else:
        return False

# Test the function
input_string = input("Enter a string: ")
if match_pattern(input_string):
    print("The string matches the pattern 'a' followed by anything ending in 'b'.")
else:
    print("The string does not match the pattern.")
```

```
Enter a string: Write a Python program that matches a string that has an 'a' followed by anything ending in 'b'  anu ansnshsb
The string matches the pattern 'a' followed by anything ending in 'b'.
```

6. Write a Python program to search for numbers (0-9) of length between 1 and 3 in a given string.

```
import re

def find_numbers(input_string):
    pattern = r'\b\d{1,3}\b'

    numbers = re.findall(pattern, input_string)

    return numbers

# Test the function
input_string = input("Enter a string: ")
found_numbers = find_numbers(input_string)

if found_numbers:
    print("Numbers found in the string:")
    for number in found_numbers:
        print(number)
else:
    print("No numbers of length 1 to 3 found in the string.")
```

```
Enter a string: Write a Python program to search for numbers (0-9) of length between 1 and 3 in a given strin
Numbers found in the string:
0
9
1
3
```

7. Write a Python program to search for literal strings within a string.
   Sample text : 'The quick brown fox jumps over the lazy dog.'
   Searched words : 'fox', 'dog', 'horse'

```
import re

input_str = str(input("enter a srt : "))
pattern = r"\b\wo."

result = re.findall(pattern, input_str)
print(result)
```

```
enter a srt : The quick brown fox jumps over the lazy dog
['fox', 'dog']
```

8. Write a Python program to search for a literal string in a string and also find the location within the original string where the pattern occurs.

Sample text : 'The quick brown fox jumps over the lazy dog.'
Searched words : 'fox'

```
sample_text = 'The quick brown fox jumps over the lazy dog.'
searched_word = 'fox'

index = sample_text.find(searched_word)

if index != -1:
    print(f"The word '{searched_word}' was found at position {index}.")
else:
    print(f"The word '{searched_word}' was not found in the text.")
```

```
The word 'fox' was found at position 16.
```

9. Write a Python program to extract year, month and date from an URL

```
import re

url1 = "https://www.washingtonpost.com/news/football-insider/wp/2016/09/02/odell-beckhams-fame-rests-on-one-stupid-little-ball-josh-norman-te

# Define a regular expression pattern to match the year, month, and date in the URL
pattern = r'(\d{4})/(\d{2})/(\d{2})/'

# Use re.search() to find the matched groups in the URL
match = re.search(pattern, url1)

if match:
    year = match.group(1)
    month = match.group(2)
    date = match.group(3)
    print("Year:", year)
    print("Month:", month)
    print("Date:", date)
else:
    print("No date information found in the URL.")
```

```
Year: 2016
Month: 09
Date: 02
```

10. Write a Python program to find URLs in a string.

```
import re

text = '<p>Contents :</p><a href="https://w3resource.com">Python Examples</a><a href="http://github.com">Even More Examples</a>'

# Define a simplified regular expression pattern to match URLs
pattern = r'https?://\S+'

# Use re.findall() to find all URLs in the text
urls = re.findall(pattern, text)

if urls:
    print("Found URLs:")
```

```
    for url in urls:
        print(url)
else:
    print("No URLs found in the text.")
```

```
Found URLs:
https://w3resource.com">Python
http://github.com">Even
```

11. Write a Python program to remove the parenthesis area in a string.\

Sample data : ["example (.com)", "w3resource", "github (.com)", "stackoverflow (.com)"]
Expected Output:
example
w3resource
github
stackoverflow.

```
import re

data = ["example (.com)", "w3resource", "github (.com)", "stackoverflow (.com)"]

# Define a regular expression pattern to match text inside parentheses and the parentheses themselves
pattern = r'\s*\([^)]*\)'

# Iterate over the data and remove the parenthesis area using re.sub()
result = [re.sub(pattern, '', item) for item in data]

# Print the result
for item in result:
    print(item)
```

```
example
w3resource
github
stackoverflow
```

12. Write a Python program to concatenate the consecutive numbers in a given string.

Original string:
Enter at 1 20 Kearny Street. The security desk can direct you to floor 1 6. Please have your identification ready.

After concatenating the consecutive numbers in the said string:
Enter at 120 Kearny Street. The security desk can direct you to floor 16. Please have your identification ready.

```
import re

original_string = "Enter at 1 20 Kearny Street. The security desk can direct you to floor 1 6. Please have your identification ready."

# Define a regular expression pattern to match consecutive numbers
pattern = r'(\d+)\s+(\d+)'

# Use re.sub() to replace consecutive numbers with their concatenation
result_string = re.sub(pattern, lambda x: x.group(1) + x.group(2), original_string)

print("Original string:")
print(original_string)

print("\nAfter concatenating the consecutive numbers:")
print(result_string)
```

```
Original string:
Enter at 1 20 Kearny Street. The security desk can direct you to floor 1 6. Please have your identification ready.

After concatenating the consecutive numbers:
Enter at 120 Kearny Street. The security desk can direct you to floor 16. Please have your identification ready.
```