

Assignment 2

July 16, 2023

```
[1]: import re

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

string1 = "Abc123"
string2 = "Hello_World"
string3 = "12345"
string4 = "@#$$%^"
string5 = "abcDEF123"

print(check_string(string1))
print(check_string(string2))
print(check_string(string3))
print(check_string(string4))
print(check_string(string5))
```

True
False
True
False
True

```
[2]: import re

def match_string(input_string):
    pattern = r'ab*'
    if re.match(pattern, input_string):
        return True
    else:
        return False

string1 = "a"
string2 = "ab"
```

```
string3 = "abb"
string4 = "abc"

print(match_string(string1))
print(match_string(string2))
print(match_string(string3))
print(match_string(string4))
```

True
True
True
True

```
[3]: import re

def match_string(input_string):
    pattern = r'ab+'
    if re.match(pattern, input_string):
        return True
    else:
        return False

string1 = "a"
string2 = "ab"
string3 = "abb"
string4 = "abc"

print(match_string(string1))
print(match_string(string2))
print(match_string(string3))
print(match_string(string4))
```

False
True
True
True

```
[4]: import re

def match_string(input_string):
    pattern = r'ab?'
    if re.match(pattern, input_string):
        return True
    else:
        return False

string1 = "a"
```

```

string2 = "ab"
string3 = "abb"
string4 = "abc"

print(match_string(string1))
print(match_string(string2))
print(match_string(string3))
print(match_string(string4))

```

True
True
True
True

```

[6]: import re

def match_string(input_string):
    pattern = r'ab{3}'
    if re.search(pattern, input_string):
        return True
    else:
        return False

```

```

string1 = "abbb"
string2 = "abb"
string3 = "aabbbcc"
string4 = "abbbbbb"
string5 = "abc"

print(match_string(string1))
print(match_string(string2))
print(match_string(string3))
print(match_string(string4))
print(match_string(string5))

```

True
False
True
True
False

```

[11]: import re

text = "ImportanceOfRegularExpressionsInPython"
result = re.findall(r'[A-Z][^A-Z]*', text)
print(result)

```

['Importance', 'Of', 'Regular', 'Expressions', 'In', 'Python']

```
[8]: import re

def match_string(input_string):
    pattern = r'ab{2,3}'
    if re.search(pattern, input_string):
        return True
    else:
        return False

string1 = "abb"
string2 = "abbb"
string3 = "aabbbcc"
string4 = "abbbbb"
string5 = "abc"

print(match_string(string1))
print(match_string(string2))
print(match_string(string3))
print(match_string(string4))
print(match_string(string5))
```

```
True
True
True
True
False
```

```
[13]: import re

def find_sequences(input_string):
    pattern = r'[a-z]+_[a-z]+'
    sequences = re.findall(pattern, input_string)
    return sequences

string = "My_Name is_Krishna_kant_sinha!"

sequences = find_sequences(string)
print(sequences)
```

```
['rishna_kant']
```

```
[14]: import re

def match_string(input_string):
    pattern = r'a.*b$'
    if re.match(pattern, input_string):
        return True
```

```

        else:
            return False

string1 = "ab"
string2 = "acdb"
string3 = "axbycz"
string4 = "abc"
string5 = "abcd"

print(match_string(string1))
print(match_string(string2))
print(match_string(string3))
print(match_string(string4))
print(match_string(string5))

```

True
True
False
False
False

```

[16]: import re

def match_word(input_string):
    pattern = r'^\w+'
    match = re.match(pattern, input_string)
    if match:
        return match.group()
    else:
        return None

string1 = "Hello,I!"
string2 = "Am "
string3 = "Krishna"
string4 = "from"
string5 = "Ranchi"
print(match_word(string1))
print(match_word(string2))
print(match_word(string3))
print(match_word(string4))
print(match_word(string5))

```

Hello
Am
Krishna
from
Ranchi

```
[18]: import re

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

string1 = "Hello_World"
string2 = "Python-Assignment"
string3 = "Two!"
string4 = "_deadline_today"
string5 = "12"

print(match_string(string1))
print(match_string(string2))
print(match_string(string3))
print(match_string(string4))
print(match_string(string5))
```

```
True
False
False
True
True
```

```
[19]: def check_startswith_number(input_string, number):
        return input_string.startswith(str(number))

string1 = "123abc"
string2 = "456xyz"
string3 = "789pqr"
number1 = 123
number2 = 456

print(check_startswith_number(string1, number1))
print(check_startswith_number(string1, number2))
print(check_startswith_number(string2, number2))
print(check_startswith_number(string3, number1))
```

```
True
False
True
False
```

```
[21]: def remove_leading_zeros(ip_address):
    parts = ip_address.split('.')
    new_parts = [str(int(part)) for part in parts]
    new_ip_address = '.'.join(new_parts)
    return new_ip_address

ip_address1 = "456.578.090.001"
ip_address2 = "078.045.012.010"
ip_address3 = "000.001.002.003"

print(remove_leading_zeros(ip_address1))
print(remove_leading_zeros(ip_address2))
print(remove_leading_zeros(ip_address3))
```

```
456.578.90.1
78.45.12.10
0.1.2.3
```

```
[50]: import re

def match_date_string(date_string):
    pattern = _
    ↪r"\b(January|February|March|April|May|June|July|August|September|October|November|December)\s"
    matches = re.findall(pattern, date_string)
    return matches

text = "Today's date is July 9, 2023. Tomorrow will be July 10, 2023."

dates = match_date_string(text)
print(dates)
```

```
['July', 'July']
```

```
[ ]: import re

def match_date_string(text):
    pattern = _
    ↪r"\b(January|February|March|April|May|June|July|August|September|October|November|December)\s"
    ↪:st|nd|rd|th)?\s\d{4}\b"
    matches = re.matches(pattern, text)
    return matches

text = "On August 15th 1947 that India was declared independent from British_
    ↪colonialism, and the reins of control were handed over to the leaders of the_
    ↪Country."

dates = match_date_string(text)
```

```
print(dates)
```

```
[52]: def search_literals(input_string):
    search_strings = ['fox', 'dog', 'horse']
    found_strings = []

    for search_string in search_strings:
        if search_string in input_string:
            found_strings.append(search_string)

    return found_strings

text = 'The quick brown fox jumps over the lazy dog.'
found_strings = search_literals(text)
print(found_strings)
```

```
['fox', 'dog']
```

```
[58]: def search_literals_location(input_string, search_string):
    locations = []
    start = 0
    def search_literals(input_string):
        search_strings = ['fox',]
        found_strings = []

        for search_string in search_strings:
            if search_string in input_string:
                found_strings.append(search_string)

        return found_strings
    while True:
        found_index = input_string.find(search_string, start)
        if found_index == -1:
            break
        locations.append(found_index)
        start = found_index + 1

    return locations

text = 'The quick brown fox jumps over the lazy dog.'
search_string = 'o'
found_locations = search_literals_location(text, search_string)
print(found_locations)
print(found_strings)
```

None


```
['fox', 'dog']
```

```
[59]: import re

def find_substrings(input_string, pattern):
    substrings = re.findall(pattern, input_string)
    return substrings

text = 'Python exercises, PHP exercises, C# exercises'
pattern = 'exercises'

found_substrings = find_substrings(text, pattern)
print(found_substrings)
```

```
['exercises', 'exercises', 'exercises']
```

```
[61]: import re

def find_substring_occurrences(input_string, substring):
    occurrences = []
    pattern = re.compile(substring)

    for match in re.finditer(pattern, input_string):
        start = match.start()
        end = match.end()
        occurrence = {
            'substring': substring,
            'start': start,
            'end': end
        }
        occurrences.append(occurrence)

    return occurrences

text = 'Data Science, Data Scientist, Data Analyst'
substring = 'Data'

found_occurrences = find_substring_occurrences(text, substring)
for occurrence in found_occurrences:
    print(f"Occurrence: {occurrence['substring']}, Start: {occurrence['start']}, End: {occurrence['end']}")
```

```
Occurrence: Data, Start: 0, End: 4
Occurrence: Data, Start: 14, End: 18
Occurrence: Data, Start: 30, End: 34
```

```
[62]: from datetime import datetime

def convert_date(date_str):
    date_obj = datetime.strptime(date_str, "%Y-%m-%d")

    formatted_date = date_obj.strftime("%d-%m-%Y")

    return formatted_date

date_string = "2022-05-10"
converted_date = convert_date(date_string)
print(converted_date)
```

10-05-2022

```
[65]: import re

def find_words_starting_with_a_or_e(input_string):
    pattern = r'\b[aAeE]\w+\b'
    words = re.findall(pattern, input_string)
    return words

text = 'The apple is on the tree. I love my country endlessly.'
found_words = find_words_starting_with_a_or_e(text)
print(found_words)
```

['apple', 'endlessly']

```
[67]: import re

def separate_numbers(input_string):
    pattern = r'\d+'
    matches = re.finditer(pattern, input_string)

    for match in matches:
        number = match.group()
        position = match.start()
        print(f"Number: {number}, Position: {position}")

text = 'The price is $15, and the quantity is 7.'
separate_numbers(text)
```

Number: 15, Position: 14

Number: 7, Position: 38

```
[69]: import re

def extract_maximum_numeric_value(input_string):
```

```

pattern = r'\d+'
numbers = re.findall(pattern, input_string)

if numbers:
    max_number = max(numbers, key=int)
    return max_number
else:
    return None

text = 'The maximum value is 921. The minimum value is 18.'
max_number = extract_maximum_numeric_value(text)
print(max_number)

```

921

```

[71]: import re

def add_spaces_to_capital_words(input_string):
    pattern = r'(?!\s)(?=[A-Z])'
    modified_string = re.sub(pattern, ' ', input_string)
    return modified_string

text = 'HeyNeighbourHowYouAllAreDoingToday'
modified_text = add_spaces_to_capital_words(text)
print(modified_text)

```

Hey Neighbour How You All Are Doing Today

```

[73]: import re

def find_uppercase_lowercase_sequences(input_string):
    pattern = r'[A-Z][a-z]+'
    sequences = re.findall(pattern, input_string)
    return sequences

text = 'HeyNeighbourHowYouAllAreDoingToday'
found_sequences = find_uppercase_lowercase_sequences(text)
print(found_sequences)

```

['Hey', 'Neighbour', 'How', 'You', 'All', 'Are', 'Doing', 'Today']

```

[75]: import re

def remove_duplicate_words(sentence):
    pattern = r'\b(\w+)\b(?:.*\b\1\b)'
    modified_sentence = re.sub(pattern, '', sentence)
    return modified_sentence

```

```

text = 'My My name name is is Krishna Krishna.'
modified_text = remove_duplicate_words(text)
print(modified_text)

```

My name is Krishna.

```

[76]: import re

def validate_ending_alphanumeric(string):
    pattern = r'^[a-zA-Z0-9]+[a-zA-Z0-9]$$'
    match = re.match(pattern, string)
    return match is not None

text = 'Hello123'
valid = validate_ending_alphanumeric(text)
print(valid)

```

True

```

[77]: import re

def extract_hashtags(text):
    pattern = r'#\w+'
    hashtags = re.findall(pattern, text)
    return hashtags

text = """RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by
↳#Demonetization as the same has rendered USELESS
↳<ed><U+00A0><U+00BD><ed><U+00B1><U+0089> "acquired funds" No wo"""
hashtags = extract_hashtags(text)
print(hashtags)

```

['#Doltiwal', '#xyzabc', '#Demonetization']

```

[78]: import re

def remove_U_plus_symbols(text):
    pattern = r'<U\[A-F0-9\]>+'
    modified_text = re.sub(pattern, '', text)
    return modified_text

text = "@Jags123456 Bharat band on 28??
↳<ed><U+00A0><U+00BD><ed><U+00B8><U+0082>Those who are protesting
↳#demonetization are all different party leaders"
modified_text = remove_U_plus_symbols(text)
print(modified_text)

```

@Jags123456 Bharat band on 28??<ed><ed>Those who are protesting #demonetization are all different party leaders

```
[89]: import re

def extract_dates_from_file(file_path):
    pattern = r'\d{2}-\d{2}-\d{4}'
    dates = []

    with open(file_path, 'r') as file:
        text = file.read()
        dates = re.findall(pattern, text)

    return dates

file_path = 'Ron was born on 12-09-1992 and he was admitted to school 15-12-1999'
dates = extract_dates_from_file(file_path)
print(dates)
```

```
-----
FileNotFoundError                                Traceback (most recent call last)
/var/folders/2c/79v1sbds3wx6rc16_b4djbd40000gn/T/ipykernel_61091/310932759.py in
↳<module>
    13 # Test the function
    14 file_path = 'Ron was born on 12-09-1992 and he was admitted to school
↳15-12-1999'
--> 15 dates = extract_dates_from_file(file_path)
    16 print(dates)

/var/folders/2c/79v1sbds3wx6rc16_b4djbd40000gn/T/ipykernel_61091/310932759.py in
↳extract_dates_from_file(file_path)
     5     dates = []
     6
--> 7     with open(file_path, 'r') as file:
     8         text = file.read()
     9         dates = re.findall(pattern, text)

FileNotFoundError: [Errno 2] No such file or directory: 'Ron was born on
↳12-09-1992 and he was admitted to school 15-12-1999'
```

```
[91]: def replace_characters_with_colon(text):
        replacements = [' ', ',', '.', ':']
        for char in replacements:
            text = text.replace(char, ':')
        return text
```

```
text = 'Python Exercises, PHP exercises.'  
modified_text = replace_characters_with_colon(text)  
print(modified_text)
```

Python:Exercises::PHP:exercises:

[]: