```
+*In[1]:*+
[source, ipython3]
----
import regex as re
----


+*In[2]:*+
[source, ipython3]
----
import re
def is_allowed_specific_char(string):
    charRe = re.compile(r'[^a-zA-Z0-9]')
    string = charRe.search(string)
    return not bool(string)

print(is_allowed_specific_char("ABCDEFabcdef123450"))
print(is_allowed_specific_char("*&%@#!}{"))

----


+*Out[2]:*+
----
True
False
----


+*In[2]:*+
[source, ipython3]
----
import re
def text_match(text):
        patterns = '^a(b*)$'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')
print(text_match("ac"))
print(text_match("abc"))
print(text_match("a"))
print(text_match("ab"))
print(text_match("abb"))
----


+*Out[2]:*+
----
Not matched!
Not matched!
Found a match!
Found a match!
Found a match!
----


+*In[3]:*+
[source, ipython3]
----
import re
def text_match(text):
        patterns = 'ab?'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("ab"))
print(text_match("abc"))
print(text_match("abbc"))
print(text_match("aabbc"))
----


+*Out[3]:*+
----
Found a match!
Found a match!
Found a match!
Found a match!
----


+*In[4]:*+
[source, ipython3]
----
import re
def text_match(text):
        patterns = 'ab{3}?'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("abbb"))
print(text_match("aabbbbbc"))
----


+*Out[4]:*+
----
Found a match!
Found a match!
----


+*In[7]:*+
[source, ipython3]
----
import re


ini_str = 'ImportanceOfRegularExpressionsInPython'


print ("Initial String", ini_str)


res_list = []
res_list = re.findall('[A-Z][^A-Z]*', ini_str)


print("Resultant prefix", str(res_list))
----


+*Out[7]:*+
----
Initial String ImportanceOfRegularExpressionsInPython
Resultant prefix ['Importance', 'Of', 'Regular', 'Expressions', 'In', 'Python']
----


+*In[8]:*+
[source, ipython3]
----
import re
```

```
def text_match(text):
        patterns = 'ab{2,3}'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')
print(text_match("ab"))
print(text_match("aabbbbbc"))
----


+*Out[8]:*+
----
Not matched!
Found a match!
----


+*In[9]:*+
[source, ipython3]
----
import re
def text_match(text):
        patterns = '^[a-z]+_[a-z]+$'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("aab_cbbbc"))
print(text_match("aab_Abbbc"))
print(text_match("Aaab_abbbc"))
----


+*Out[9]:*+
----
Found a match!
Not matched!
Not matched!
----


+*In[10]:*+
[source, ipython3]
----
import re
def text_match(text):
        patterns = 'a.*?b$'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("aabbbbd"))
print(text_match("aabAbbbc"))
print(text_match("accddbbjjjb"))
----


+*Out[10]:*+
----
Not matched!
Not matched!
Found a match!
----


+*In[11]:*+
```



```
[source, ipython3]
----
import re
def text_match(text):
        patterns = '^\w+'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("The quick brown fox jumps over the lazy dog."))
print(text_match(" The quick brown fox jumps over the lazy dog."))
----


+*Out[11]:*+
----
Found a match!
Not matched!
----


+*In[12]:*+
[source, ipython3]
----
import re
def text_match(text):
        patterns = '^[a-zA-Z0-9_]*$'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("The quick brown fox jumps over the lazy dog."))
print(text_match("Python_Exercises_1"))
----


+*Out[12]:*+
----
Not matched!
Found a match!
----


+*In[13]:*+
[source, ipython3]
----
import re
def match_num(string):
    text = re.compile(r"^5")
    if text.match(string):
        return True
    else:
        return False
print(match_num('5-2345861'))
print(match_num('6-2345861'))

----


+*Out[13]:*+
----
True
False
----


+*In[14]:*+
```

```
[source, ipython3]
----
import re
ip = "216.08.094.196"
string = re.sub('\.[0]*', '.', ip)
print(string)

----


+*Out[14]:*+
----
216.8.94.196
----


+*In[23]:*+
[source, ipython3]
----
import re
from datetime import datetime

Schedule.txt
'On August 15th 1947 that India was declared independent from British colonialis
m, and the reins of control were handed over to the leaders of the Country'


file = open("schedule.txt",'r')
text = file.read()

match = re.search(r'\d+-\d+-\d{4}', text)

date = datetime.strptime(match.group(), '%d-%m-%Y').date()
print(f"'August 15th 1947 {date}.")
file.close()
----


+*Out[23]:*+
----

    ---------------------------------------------------------------------------

    NameError                                 Traceback (most recent call last)

    Cell In[23], line 3
          1 import re
          2 from datetime import datetime
    ----> 3 Schedule.txt
          4 'On August 15th 1947 that India was declared independent from Britis
h colonialism, and the reins of control were handed over to the leaders of the C
ountry'
          6 # open the data file


    NameError: name 'Schedule' is not defined

----


+*In[24]:*+
[source, ipython3]
----
import re
pattern = 'fox'
text = 'The quick brown fox jumps over the lazy dog.'
match = re.search(pattern, text)
s = match.start()
e = match.end()
```



```
print('Found "%s" in "%s" from %d to %d ' % \
    (match.re.pattern, match.string, s, e))
----


+*Out[24]:*+
----
Found "fox" in "The quick brown fox jumps over the lazy dog." from 16 to 19
----


+*In[25]:*+
[source, ipython3]
----
import re
pattern = 'dog'
text = 'The quick brown fox jumps over the lazy dog.'
match = re.search(pattern, text)
s = match.start()
e = match.end()
print('Found "%s" in "%s" from %d to %d ' % \
    (match.re.pattern, match.string, s, e))
----


+*Out[25]:*+
----
Found "dog" in "The quick brown fox jumps over the lazy dog." from 40 to 43
----


+*In[26]:*+
[source, ipython3]
----
import re
pattern = 'horse'
text = 'The quick brown fox jumps over the lazy dog.'
match = re.search(pattern, text)
s = match.start()
e = match.end()
print('Found "%s" in "%s" from %d to %d ' % \
    (match.re.pattern, match.string, s, e)
----


+*Out[26]:*+
----

      Cell In[26], line 8
        (match.re.pattern, match.string, s, e)

    ^
    SyntaxError: unexpected EOF while parsing


----


+*In[27]:*+
[source, ipython3]
----
import re
text = 'Python exercises, PHP exercises, C# exercises'
pattern = 'exercises'
for match in re.findall(pattern, text):
    print('Found "%s"' % match)
----
```

```
+*Out[27]:*+
----
Found "exercises"
Found "exercises"
Found "exercises"
----


+*In[28]:*+
[source, ipython3]
----
import re
text = 'Python exercises, PHP exercises, C# exercises'
pattern = 'exercises'
for match in re.finditer(pattern, text):
    s = match.start()
    e = match.end()
    print('Found "%s" at %d:%d' % (text[s:e], s, e))
----


+*Out[28]:*+
----
Found "exercises" at 7:16
Found "exercises" at 22:31
Found "exercises" at 36:45
----


+*In[30]:*+
[source, ipython3]
----
import re
def change_date_format(dt):
        return re.sub(r'(\d{4})-(\d{1,2})-(\d{1,2})', '\\3-\\2-\\1', dt)
dt1 = "2023-07-10"
print("Original date in YYY-MM-DD Format: ",dt1)
print("New date in DD-MM-YYYY Format: ",change_date_format(dt1))
----


+*Out[30]:*+
----
Original date in YYY-MM-DD Format:  2023-07-10
New date in DD-MM-YYYY Format:  10-07-2023
----


+*In[31]:*+
[source, ipython3]
----
import re

text = "The following example creates an ArrayList with a capacity of 50 element
s. Four elements are then added to the ArrayList and the ArrayList is trimmed ac
cordingly."

list = re.findall("[ae]\w+", text)

print(list)
----


+*Out[31]:*+
----
['example', 'eates', 'an', 'ayList', 'apacity', 'elements', 'elements', 'are', '
en', 'added', 'ayList', 'and', 'ayList', 'ed', 'accordingly']
----
```



```
+*In[32]:*+
[source, ipython3]
----
import re

text = "The following example creates an ArrayList with a capacity of 50 element
s. Four elements are then added to the ArrayList and the ArrayList is trimmed ac
cordingly."

for m in re.finditer("\d+", text):
    print(m.group(0))
    print("Index position:", m.start())

----


+*Out[32]:*+
----
50
Index position: 62
----


+*In[33]:*+
[source, ipython3]
----
import re

string='ab12cd123ef23'

number = re.findall('\d+', string)

number = map(int, number)
print("Max_value:",max(number))
----


+*Out[33]:*+
----
Max_value: 123
----


+*In[34]:*+
[source, ipython3]
----
import re
def capital_words_spaces(str1):
  return re.sub(r"(\w)([A-Z])", r"\1 \2", str1)

print(capital_words_spaces("Python"))
print(capital_words_spaces("PythonExercises"))
print(capital_words_spaces("PythonExercisesPracticeSolution"))
----


+*Out[34]:*+
----
Python
Python Exercises
Python Exercises Practice Solution
----


+*In[35]:*+
[source, ipython3]
----
import re
```

```
def text_match(text):
        patterns = '[A-Z]+[a-z]+$'
        if re.search(patterns, text):
                return 'Found a match!'
        else:
                return('Not matched!')
print(text_match("AaBbGg"))
print(text_match("Python"))
print(text_match("python"))
print(text_match("PYTHON"))
print(text_match("aA"))
print(text_match("Aa"))

----


+*Out[35]:*+
----
Found a match!
Found a match!
Not matched!
Not matched!
Not matched!
Found a match!
----


+*In[36]:*+
[source, ipython3]
----
def unique_list(text_str):
    l = text_str.split()
    temp = []
    for x in l:
        if x not in temp:
            temp.append(x)
    return ' '.join(temp)

text_str = "Python Exercises Practice Solution Exercises"
print("Original String:")
print(text_str)
print("\nAfter removing duplicate words from the said string:")
print(unique_list(text_str))
----


+*Out[36]:*+
----
Original String:
Python Exercises Practice Solution Exercises

After removing duplicate words from the said string:
Python Exercises Practice Solution
----


+*In[37]:*+
[source, ipython3]
----

import re

regex = '[a-zA-z0-9]$'

def check(string):


    if(re.search(regex, string)):
        print("Accept")
```



```
    else:
        print("Discard")


# Driver Code
if __name__ == '__main__' :


    string = "ankirai@"


    check(string)

    string = "ankitrai326"
    check(string)

    string = "ankit."
    check(string)

    string = "geeksforgeeks"
    check(string)
----


+*Out[37]:*+
----
Discard
Accept
Discard
Accept
----


+*In[45]:*+
[source, ipython3]
----
from dateutil import parser
str1 = "Ron was born on 12-09-1992 and he was admitted to school 15-12-1999"

print("The given string is")
print(str1)

date = parser.parse(str1, fuzzy=True)
print("The date present in the string is")
print(str(date)[:10])
----


+*Out[45]:*+
----
The given string is
Ron was born on 12-09-1992 and he was admitted to school 15-12-1999
The date present in the string is
1992-12-09
----


+*In[46]:*+
[source, ipython3]
----
import re
text = 'Python Exercises, PHP exercises.'
print(re.sub("[ ,.]", ":", text))
----


+*Out[46]:*+
----
```

```
Python:Exercises::PHP:exercises:
----


+*In[50]:*+
[source, ipython3]
----
text1 = """RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetizat
ion as the same has rendered USELESS <ed><U+00A0><U+00BD><ed><U+00B1><U+0089> "a
cquired funds" No wo"""
textList = text1.split()
for i in textList:
    if(i.startswith("#")):
        x = i.replace("#", '')
        print(x)
----


+*Out[50]:*+
----
Doltiwal
xyzabc
Demonetization
----


+*In[56]:*+
[source, ipython3]
----
import re


original = "@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U+00
82>Those who  are protesting #demonetization  are all different party leaders"


print ("Original String: " + original)


unwanted = "[<U+00A0>]"


original = re.sub(unwanted, '', original)
print("Final Output:"+ original)
----


+*Out[56]:*+
----
Original String: @Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8>
<U+0082>Those who  are protesting #demonetization  are all different party leade
rs
Final Output:@Jags123456 Bharat band on 28??edBDedB882Those who  are protesting
#demonetization  are all different party leaders
----


+*In[ ]:*+
[source, ipython3]
----

----
```