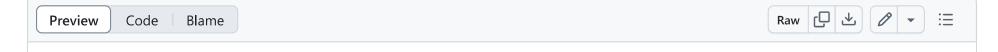


Module-3 / 3A STRING - FIND AND REPLACE.md ☐



39 lines (29 loc) · 1.07 KB



Exp.No:3a

STRING - FIND AND REPLACE

AIM

To write a Python function to accept a string, identify a word to be replaced, and replace it with a new word provided by the user.

ALGORITHM

1. Begin the program.

- 2. Input the original string str1 and the word to be replaced replace_str.
- 3. Ask the user to input the new replacement word str2.
- 4. Use the replace() method in Python to replace all occurrences of replace_str in str1 with str2.
- 5. Store the modified string in str3.
- 6. Display the original string (str1) and the modified string (str3).
- 7. Terminate the program.

```
Reg no-212223020028

Name-Tharani devi.G

with your code

def replacestr(n,p):
    print("The old string is {}".format(n))
    print("the new string is {}".format(n.replace(p,x)))

x=str(input())
```

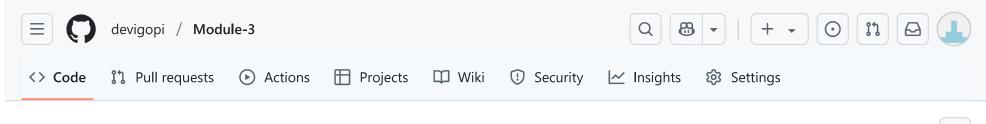
OUTPUT

	Test	Input	Expected	Got	
~	replacestr("cold coffee","cold")	Hot	The old string is cold coffee the new string is Hot coffee	The old string is cold coffee the new string is Hot coffee	~
~	replacestr("learning python","learning")	Mastering	The old string is learning python the new string is Mastering python	The old string is learning python the new string is Mastering python	~
~	replacestr("Hot chocolate","chocolate")	burger	The old string is Hot chocolate the new string is Hot burger	The old string is Hot chocolate the new string is Hot burger	~

Passed all tests! 🗸

RESULT

This program for string, identify a word to be replaced, and replace it with a new word provided by the user is successfully executed.



Module-3 / 3B REGEX - PATTERN MATCHING USING REGEX.md



41 lines (32 loc) · 1.01 KB



Exp.No:3b

REGEX - PATTERN MATCHING USING REGEX

AIM

To write a Python program that matches a string containing an 'a' followed by **two to three** 'b' characters using regular expressions.

ALGORITHM

1. Begin the program.

- 2. Accept a string str1 from the user.
- 3. Define the regular expression pattern as $r"[a]+b\{2,3\}"$.
- 4. Use the re.match() function to check if the string str1 matches the pattern.
- 5. If a match is found, print "Found a match!".
- 6. If no match is found, print "Not matched!".
- 7. Terminate the program.

```
Reg no:212223020028

Name:Tharani devi.G

write your code

import re

a=input()

x=re.findall(r".{2,3}b",a)

if x:
    print("Found a match!")

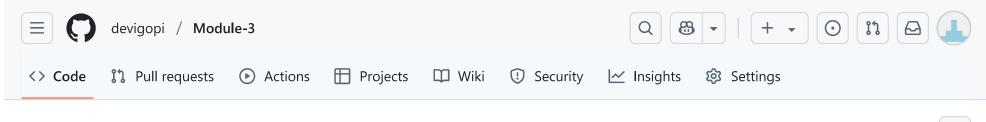
else:
    print("Not matched!")
```

OUTPUT



RESULT

This program for string containing an 'a' followed by **two to three** 'b' characters using regular expression is successfully executed.



Module-3 / 3C LIST - EVEN NUMBERS LIST.md



42 lines (31 loc) · 972 Bytes



Exp.No:3c

LIST - EVEN NUMBERS LIST

AIM

To write a Python function that accepts a number N and creates a list containing all even numbers up to N.

ALGORITHM

- 1. Begin the program.
- 2. Accept an integer a from the user.

35c0142 · 37 minutes ago

- 3. Create an empty list 1.
- 4. Use a for loop to iterate through numbers from 1 to a 1:
 - For each number i, check if it is even using i % 2 == 0.
 - o If it is even, append i to the list 1.
- 5. Print the final list 1 containing all the even numbers.
- 6. Terminate the program.

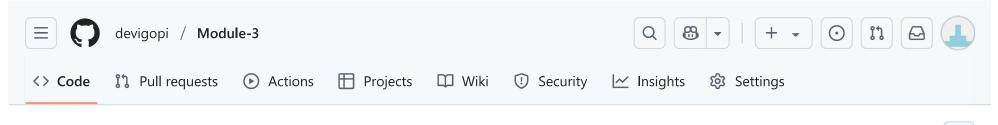
OUTPUT

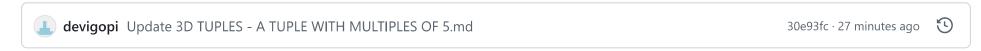
	Test	Expected	Got	
~	createlist(10)	[2, 4, 6, 8]	[2, 4, 6, 8]	~
~	createlist(15)	[2, 4, 6, 8, 10, 12, 14]	[2, 4, 6, 8, 10, 12, 14]	~

Passed all tests! 🗸

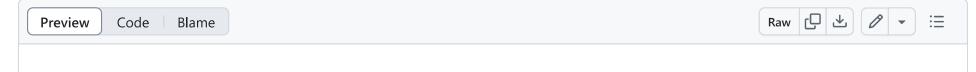
RESULT

This program number ${\bf N}$ and creates a list containing all even numbers up to ${\bf N}$ is successfully executed.





36 lines (26 loc) · 863 Bytes



Exp.No:3d

AIM

To write a Python program to create a tuple containing all multiples of 5 up to a given number N.

ALGORITHM

- 1. Begin the program.
- 2. Accept an integer N from the user.

- 3. Use a generator expression inside the tuple() function to create a tuple multiples_of_5 with values starting from 5 up to N 1, stepping by 5.
- 4. Return the tuple multiples_of_5.
- 5. Print the resulting tuple.
- 6. Terminate the program.

```
Reg no-212223020028
Name-Tharani devi.G
n= int(input())

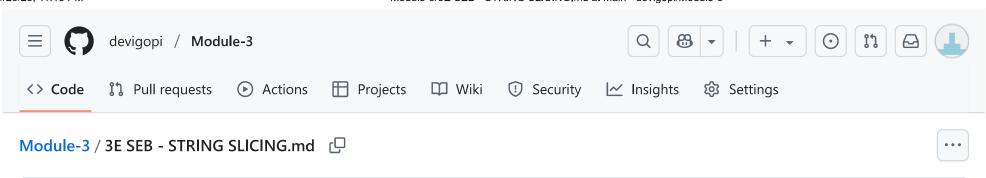
L = list(range(5,n,5))
print(tuple(L))
```

OUTPUT

	Input	Expected	Got	
~	25	(5, 10, 15, 20)	(5, 10, 15, 20)	~
~	50	(5, 10, 15, 20, 25, 30, 35, 40, 45)	(5, 10, 15, 20, 25, 30, 35, 40, 45)	~

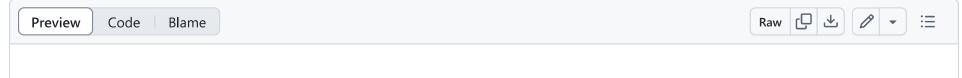
RESULT

This program for tuple containing all multiples of 5 up to a given number N is successufully executed.



44 lines (32 loc) · 997 Bytes

devigopi Update 3E SEB - STRING SLICING.md



Exp.No:3e

SEB - Resultant list

AIM

Write a non fruitful and non parameterized function to get two lists and extend the second list with the first and print the resultant list.

ALGORITHM

6ca9064 · 1 minute ago

- 1. Begin the program.
 - 2.Define a function with no parameters. 3.Inside the function: Create two lists (list1 and list2) with some values. Use list2.extend(list1) to add all elements of list1 to list2. Print list2. 4.Call the function.
- 2. Terminate the program.

```
Reg no-212223020028

Name-Tharani devi.G

write your code

def extendlist():

    L1=eval(input())

    L2=eval(input())

    print(f"List1= {L1}")

    print(f"List2= {L2+L1}")

    print(f"Resultant List={L2+L1}")
```

OUTPUT

	Test	Input	Expected	Got	
*	extendlist()		List1= ['python', 'language'] List2= ['rocks', '!!!', 'python', 'language'] Resultant List=['rocks', '!!!', 'python', 'language']	List1= ['python', 'language'] List2= ['rocks', '!!!', 'python', 'language'] Resultant List=['rocks', '!!!', 'python', 'language']	*
~	extendlist()	[1,2,3,4] [5,6,7,8,9]	List1= [1, 2, 3, 4] List2= [5, 6, 7, 8, 9, 1, 2, 3, 4] Resultant List=[5, 6, 7, 8, 9, 1, 2, 3, 4]	List1= [1, 2, 3, 4] List2= [5, 6, 7, 8, 9, 1, 2, 3, 4] Resultant List=[5, 6, 7, 8, 9, 1, 2, 3, 4]	~

RESULT

This program for non parameterized function to get two lists and extend the second list with the first and print the resultant list is successfully executed.

Started on Saturday, 6 April 2024, 1:52 PM

State Finished

Completed on Saturday, 6 April 2024, 2:07 PM

Time taken 14 mins 54 secs **Grade 80.00** out of 100.00

Question 1

Correct

Mark 20.00 out of 20.00

Write a python program to read and print the complex number

For example:

Input	Result
5	(5+7j)
7	

Answer: (penalty regime: 0 %)

- 1 a=int(input())
- b=int(input())
- 3 d=complex(a,b)
- 4 print(d)

	Input	Expected	Got	
~	5 7	(5+7j)	(5+7j)	~
~	8	(8+6j)	(8+6j)	~

Passed all tests! 🗸

Correct

Marks for this submission: 20.00/20.00.

Question $\mathbf{2}$

Correct

Mark 20.00 out of 20.00

Write a non fruitful and non parameterized function to get two lists and extend the second <u>list</u> with the first and print the resultant list.

Hint:use eval()

For example:

Test	Input	Result
extendlist()	["python","language"] ["rocks","!!!"]	List1= ['python', 'language'] List2= ['rocks', '!!!', 'python', 'language'] Resultant List=['rocks', '!!!', 'python', 'language']

Answer: (penalty regime: 0 %)

	Test	Input	Expected	Got	
~	extendlist()	["python","language"] ["rocks","!!!"]	List1= ['python', 'language'] List2= ['rocks', '!!!', 'python', 'language'] Resultant List=['rocks', '!!!', 'python', 'language']	List1= ['python', 'language'] List2= ['rocks', '!!!', 'python', 'language'] Resultant List=['rocks', '!!!', 'python', 'language']	*
~	extendlist()	[1,2,3,4] [5,6,7,8,9]	List1= [1, 2, 3, 4] List2= [5, 6, 7, 8, 9, 1, 2, 3, 4] Resultant List=[5, 6, 7, 8, 9, 1, 2, 3, 4]	List1= [1, 2, 3, 4] List2= [5, 6, 7, 8, 9, 1, 2, 3, 4] Resultant List=[5, 6, 7, 8, 9, 1, 2, 3, 4]	~

Passed all tests! 🗸

Marks for this submission: 20.00/20.00.

```
Question \bf 3
Correct
Mark 20.00 out of 20.00
 Copy element 44 and 55 from the following <u>tuple</u> into a new <u>tuple</u>
 tuple1 = (11, 22, 33, 44, 55, 66)
 Expected output:
 tuple2 = (44, 55)
 For example:
  Result
  (44, 55)
 Answer: (penalty regime: 0 %)
    1 | tuple1 = (11,22,33,44,55,66)
       tuple2 = (44,55)
    3 print(tuple2)
         Expected Got
         (44, 55)
                    (44, 55) 🗸
  Passed all tests! ✓
 Marks for this submission: 20.00/20.00.
```

Question 4
Correct
Mark 20.00 out of 20.00

Write a Python program to find sequences of Lower case letters joined with a '@'.

For example:

Input	Result
saveetha@engineering	Found a match!
saveetha engineering	Not matched!

Answer: (penalty regime: 0 %)

```
import re
a=input()
pat='[a-z]+@+[a-z]'
if re.search(pat,a):
    print("Found a match!")
else:
    print("Not matched!")
```

	Input	Expected	Got	
~	saveetha@engineering	Found a match!	Found a match!	~
~	saveetha engineering	Not matched!	Not matched!	~

Passed all tests! 🗸



Marks for this submission: 20.00/20.00.

Question ${\bf 5}$ Not answered Mark 0.00 out of 20.00 Write a python function Vowcount() that accepts a string and should count and display the occurrence of words starting with a vowel in the given string. For example: Test Result Vowcount("Education is one thing no one can take away from you") Education is one one away No. of Words starts with vowels are 5 Answer: (penalty regime: 0 %) 1