

Regular Expressions Questions and Answers

Q) Write a Python Program to check whether the given number is valid mobile number or not?

```
import re
n = input("Enter number:")
m = re.fullmatch("[6-9]\d{9}" , n)
if m != None:
    print("Valid Mobile Number")
else:
    print("Invalid Mobile Number")
```

Output:

Enter number: 9948853025
Valid Mobile Number

Enter number:3898823
Invalid Mobile Number

Q) Write a Python Program to check whether the given mail id is valid gmail id or not ?

```
import re
s = input("Enter Mail id : ")
m = re.fullmatch("\w[a-zA-Z0-9_]*@gmail[.].com" , s)
if m != None:
    print("Valid Mail Id");
else:
    print("Invalid Mail id")
```

Output:

Enter Mail id : srinivas@gmail.com
Valid Mail Id

Enter Mail id : satya@mail
Invalid Mail id

Q) Write a python program to check whether given car registration number is valid Telangana State Registration number or not ?

```
import re
```

```
s = input("Enter Vehicle Registration Number : ")
m = re.fullmatch("TS[012][0-9][A-Z]{2}\\d{4}" , s)
if m != None:
    print("Valid Vehicle Registration Number");
else:
    print("Invalid Vehicle Registration Number")
```

Output:

Enter Vehicle Registration Number:TS12AB9010
Valid Vehicle Registration Number

Enter Vehicle Registration Number:TS12XY90
Invalid Vehicle Registration Number

Q) Write a Python Program to check whether the given mobile number is valid OR not (10 digit OR 11 digit OR 12 digit) ?

```
import re
s=input("Enter Mobile Number:")
m=re.fullmatch("(0|91)?[7-9][0-9]{9}" , s)
if m != None:
    print("Valid Mobile Number");
else:
    print("Invalid Mobile Number")
```

Output:

Enter Mobile Number:9010607010
Valid Mobile Number

Enter Mobile Number:9010203040506
Invalid Mobile Number

Q) Validate user input using regular expressions

```
import re
user_name = input("Please enter your name: ")
if not re.match("^[A-Za-z]*$", user_name):
    print ("Error! Make sure you only use letters in your name")
else:
    print("Hello "+ user_name)
```

How to Find Valid Phone Number

Task : Write a function that accepts a string and searches it for a valid phone number . Return the phone number if found.

A valid phone number may be one of the following:

- (xxx)-xxx-xxxx
- xxx-xxx-xxxx

Example:

```
import re
def find_valid_phone(text):
    PHONE_REGEX = r'(\d{3}|\(\d{3}\))-\d{3}-\d{4}'
    phone = re.search(PHONE_REGEX, text)

    if phone != None:
        return phone.group()

if __name__ == '__main__':
    print(find_valid_phone('Call me at 415-555-1011 tomorrow.'))
    print(find_valid_phone('My number is (212)-543-2212 tomorrow.'))
```

Output:

```
415-555-1011
(212)-543-2212
```

Strong Password Detector

Write a function that employs regular expressions to ensure the password given to the function is strong.

A strong password is defined as follows:

- at least eight characters long
- contains one uppercase character
- contains one lowercase character
- has at least one digit
- has at least one special character

You may need to test the string against multiple regex patterns to validate its strength.

To solve this problem, we'll be using the regex “**search**” operator again to ensure that each pattern (password criteria) is fulfilled.

Example:

```
import re
```

```
def validate_password(password):
```

```
    # REGEX PATTERN THAT CHECKS PASSWORD HAS AT LEAST 8  
    CHARACTERS
```

```
    at_least_8 = r".{8,}"
```

```
    # REGEX PATTERN THAT CHECKS PASSWORD HAS 1 LOWERCASE  
    one_lowercase = r"[a-z]"
```

```
    # REGEX PATTERN THAT CHECKS PASSWORD HAS 1 UPPERCASE  
    one_uppercase = r"[A-Z]"
```

```
    # REGEX PATTERN THAT CHECKS PASSWORD HAS 1 DIGIT  
    one_digit = r"\d"
```

```
    # REGEX PATTERN THAT CHECKS PASSWORD HAS 1 SPECIAL  
    CHARACTER
```

```
    special_characters = r"[!\"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~]"
```

```
    if re.search(at_least_8, password) == None:  
        print('Error: Password must have at least 8 characters')  
        return False
```

```
    elif re.search(one_lowercase, password) == None:  
        print('Error: Password must have at least 1 lowercase character')  
        return False
```

```
    elif re.search(one_uppercase, password) == None:  
        print('Error: Password must have at least 1 uppercase character')  
        return False
```

```
    elif re.search(one_digit, password) == None:  
        print('Error: Password must have at least 1 digit')  
        return False
```

```
elif re.search(special_characters, password) == None:  
    print('Error: Password must have at least 1 special character')  
    return False
```

```
return True
```

```
password = input('Enter a password :')  
print(validate_password(password))
```

Output 1:

```
Enter a password :Password123  
Error: Password must have at least 1 special character  
False
```

Output 2:

```
Enter a password :Welcome*123  
True
```

Python Program to check special characters

We will learn to check if a string contains any **special character** using Python. [Strings in Python](#) are a sequence of characters wrapped inside single, double, or triple quotes. The special character is a character that is not an alphabet or number. Symbols, accent marks, and punctuation marks are considered special characters.

[@ _ ! # \$ % ^ & * () < > ? / \ | { } ~ :] are some **special characters**

We have to write a program that will check for such special characters in the given string and will accept only those strings which does not have any special character

Look at the examples to understand the input and output format.

Input: "Hello!!"

Output: string is not accepted

Input: "hello123"

Output: string is accepted

To execute this task we will make a regular expression using **compile()** that will have all the special characters which we don't want in our string. Then using the `search()` method we will search if any special character is present

in the string or not. If no character is found the **search()** method will return None and then we can print that the string is accepted.

Algorithm

Step 1- Import re module

Step 2- Define a function to check for special characters

Step 3- Create a regular expression of all the special characters

Step 4- Check if this expression is in the string

Step 5- If not found return that the string is accepted

Step 6- Else return that the string is accepted

Example:

```
import re
def find(string):
    special_char=re.compile('[!@#$%^&*~+~]')

    if special_char.search(string):
        return "string not accpeted"
    else:
        return "string is accepted"

input_str = input('Enter a string value :')
print(find(input_str))
```

Output 1:

```
Enter a string value :Welcome
string is accepted
```

Output 2:

```
Enter a string value :Welcome*123
string not accpeted
```

Replace domain names of all email IDs in a list.

```
import re
emails=['aa@xyz.com', 'bb@abc.com', 'cc@mnop.com']
gmails=[re.sub(r'@\w+(\w+)', '@gmail.com', x) for x in emails]
print(gmails)
```

Output:

```
['aa@gmail.com', 'bb@gmail.com', 'cc@gmail.com']
```

```
|
```