REGULAR EXPRESSION

```
import re
```

Search function

```
txt="C++ is a programming langugage"
x= re.search("\AC++.*langugage$", txt)
print(x)
```

⤷ `<re.Match object; span=(0, 30), match='C++ is a programming langugage'>`

```
p = re.search("\s", txt)
print("The first white-space character is located in position:", p.start())
```

⤷ The first white-space character is located in position: 3

findall function

```
y = re.findall("ar", txt)
y
```

⤷ `[]`

```
z=re.findall("ing",txt)
z
```

⤷ `['ing']`

```
z1=re.findall(" ",txt)
z1
```

⤷ `[' ', ' ', ' ', ' ']`

Split function

```
a = re.split("\s", txt)
a
```

⤷ `['C++', 'is', 'a', 'programming', 'langugage']`

```
b = re.split("\s", txt, 1)
b
```

⤷ `['C++', 'is a programming langugage']`

```
b = re.split("\s", txt, 5)
b
```

⤷ `['C++', 'is', 'a', 'programming', 'langugage']`

Matching individual word characters [Using \w]

```
ans = re.findall(r'\w', txt)
print("Matches for \\w:", ans)
```

⤷ Matches for \w: ['C', 'i', 's', 'a', 'p', 'r', 'o', 'g', 'r', 'a', 'm', 'm', 'i', 'n', 'g', 'l', 'a', 'n', 'g', 'u', 'g', 'a', 'g',

◀ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ▶

Matches non-word characters

```
ans1 = re.findall(r'\W', txt)
print("Matches for \\W:", ans1)
```

⤷ Matches for \W: ['+', '+', ' ', ' ', ' ', ' ']

Splitting on Non-Word Characters (\W)

```
ans2 = re.split(r'\W+', txt)
print("Split txt:", ans2)
```

```
Split txt: ['C', 'is', 'a', 'programming', 'langugage']
```

Matching Words (\w+) and Non-Words (\W+)

```
ans3 = re.findall(r'\w+', txt)
print("Words:", ans3)
```

```
Words: ['C', 'is', 'a', 'programming', 'langugage']
```

Matches sequences of non-word characters

```
ans4 = re.findall(r'\W+', txt)
print("Non-Words:", ans4)
```

```
Non-Words: ['++ ', ' ', ' ', ' ']
```

Start coding or generate with AI.

Sub function [replaces the matches with the text of your choice]

```
c = re.sub("\s", "9", txt)
c
```

```
c1 = re.sub("\s", "9", txt, 2)
c1
```

⌄ LAB 2

```
import re
string = "MuditGarg79"
print(bool(re.fullmatch(r"[a-zA-Z0-9]+", string)))
```

```
True
```

```
string = "abbb"
print(bool(re.fullmatch(r"a*b", string)))  # '*' means zero or more occurrences
```

```
False
```

```
string = "abbb"
print(bool(re.fullmatch(r"a+b", string)))  # '+' means one or more occurrences
```

```
False
```

```
string = "ab"
print(bool(re.fullmatch(r"a?b", string)))  # '?' means zero or one occurrence
```

```
True
```

```
string = "abbb"
print(bool(re.fullmatch(r"a{3}b", string)))  # '{3}' means exactly 3 occurrences
```

```
False
```

```
string = "abb"
print(bool(re.fullmatch(r"a{2,3}b", string)))  # '{2,3}' means between 2 to 3 occurrences
```

```
False
```

```python
string = "hiii mudit this side"
print(re.findall(r"[a-z]+_[a-z]+", string))  # Matches lowercase letters joined by '_'
```

⇥ []

```python
string = "axyzb"
print(bool(re.fullmatch(r"a.*b", string)))  # '.*' means anything in between
```

⇥ True

```python
string = "Hello Manav Rachna!"
print(bool(re.match(r"^\w+", string)))  # '^' means start of line
```

⇥ True

```python
string = "This is Faridabad."
print(bool(re.search(r"\w+[.!]?$", string)))  # '\w+' matches word, '[.!]?$' checks optional punctuation
```

⇥ True

```python
string = "Currently in 3rd year"
print(re.findall(r"\b\w*z\w*\b", string))  # '\b' ensures whole word match
```

⇥ []

```python
string = "amazing crowd of college"
print(re.findall(r"\b\w+z\w+\b", string))  # Ensures 'z' is in the middle
```

⇥ ['amazing']

```python
string = "Namaste_123"
print(bool(re.fullmatch(r"\w+", string)))  # '\w' matches letters, numbers, and underscores
```

⇥ True

```python
string = "099999"
print(bool(re.fullmatch(r"5\d*", string)))  # '5' is fixed, '\d*' means any number of digits
```

⇥ False

```python
ip = "192.168.100.001"
print(re.sub(r'\b0+(\d)', r'\1', ip))  # '\b0+(\d)' removes leading zeros
```

⇥ 192.168.100.1

```python
string = "hyeboy"
print(bool(re.search(r"\d+$", string)))  # '\d+$' ensures number is at the end
```

⇥ False

```python
string = "Natural language processes 1, 45, 34, and 18 are important"
print(re.findall(r"\b\d{1,3}\b", string))  # '\d{1,3}' matches numbers of length 1 to 3
```

⇥ ['1', '45', '34', '18']

```python
text = "Good Morning everyone present here myself Mudit."
searched_words = ['everyone', 'Mudit', 'Morning']
print([word for word in searched_words if re.search(rf"\b{word}\b", text)])  # Matches exact words
```

⇥ ['everyone', 'Mudit', 'Morning']

```python
text = "Hello Everyone Mudit this side"
word = "Mudit"
print([(m.start(), m.end()) for m in re.finditer(rf"\b{word}\b", text)])  # Finds start and end index
```

```
[(15, 20)]
```