REGULAR EXPRESSIONS IN PYTHON

Regular expressions

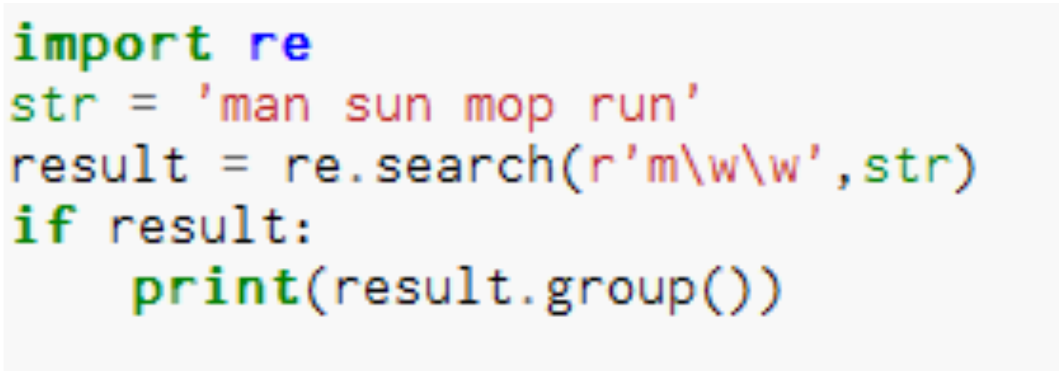
- A regular expression is a string that contains special symbols and characters to find and extract the information needed by us from the given data.
- A regular expression helps us to search information, match, find and split information as per our requirements.
- A regular expression is also called as ***regex.***
- Python provides ***re module*** that stands for regular expression.
- This module contains methods like compile(), search(), match(), findall(), split() etc.. which are used in finding the information in the available data.

Regular expressions

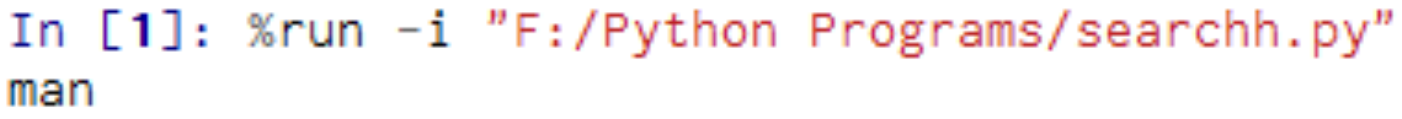
- To import re module write:
  - **import re**
- Regular expressions are nothing but strings containing characters and special symbols.
- A simple regular expression may look like this:
  - **reg = r'm\w\w'**
- The string is prefixed with 'r' to represent that it is raw string.
- Generally, write regular expressions as raw strings.
- Example if you write a normal string:
  - **str='This is normal\nstring'**

Regular expressions

- Python program to create a regular expression to search for strings starting with m and having total 3 characters using the **search()** method.



```python
import re
str = 'man sun mop run'
result = re.search(r'm\w\w',str)
if result:
    print(result.group())
```



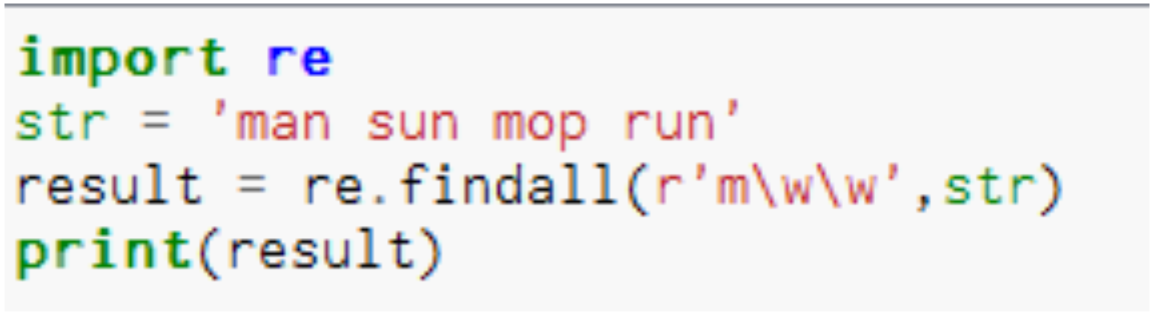
```
In [1]: %run -i "F:/Python Programs/searchh.py"
man
```

Regular expressions

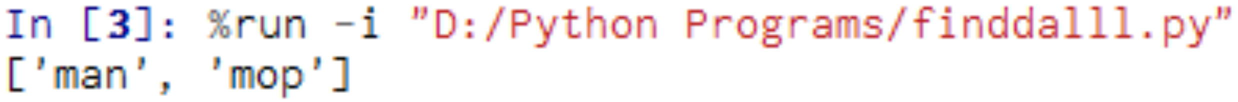
- The search() method searches for the strings according to the regular expression and returns only the first string.
- Even though there are two strings ‘man’ and ‘mop’ it extracts only the first one.
- This string can be extracted using group() method.
- Suppose we want to get all the strings that match the pattern use findall() method instead of search() method.

Regular expressions

- Python program to create a regular expression to search for strings starting with m and having total 3 characters using findall() method.



```
import re
str = 'man sun mop run'
result = re.findall(r'm\w\w',str)
print(result)
```

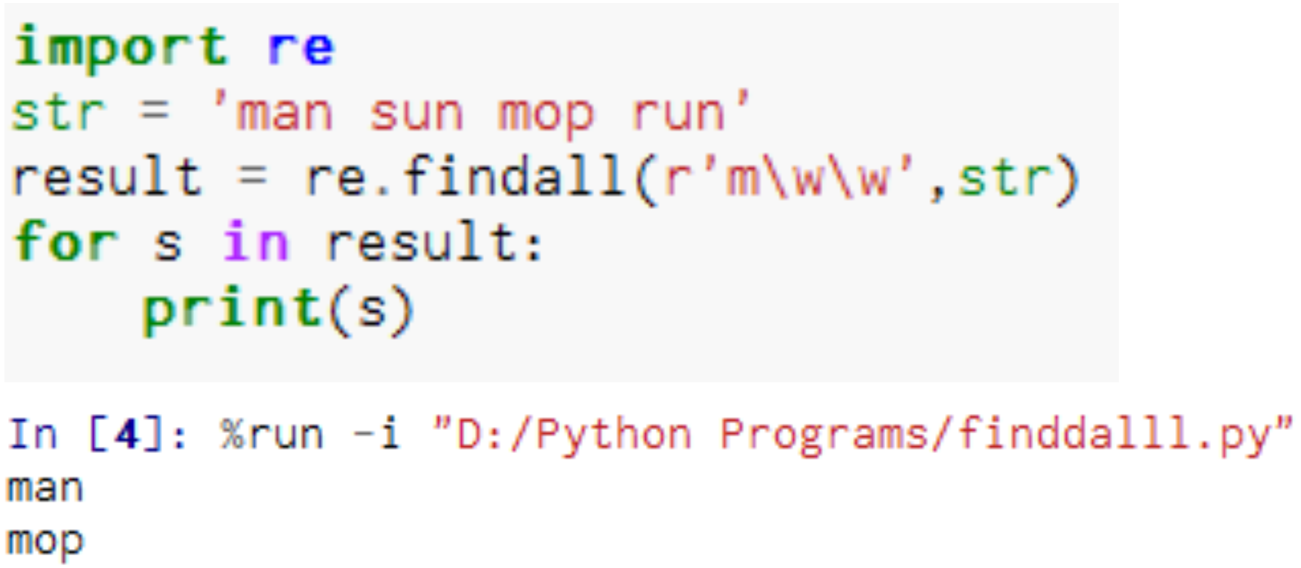


```
In [3]: %run -i "D:/Python Programs/finddalll.py"
['man', 'mop']
```

- The findall() method returns the result as a list.
- The elements of the list can be displayed using a for loop as:

Regular expressions

- **finall()** method to print elements of list using for loop:

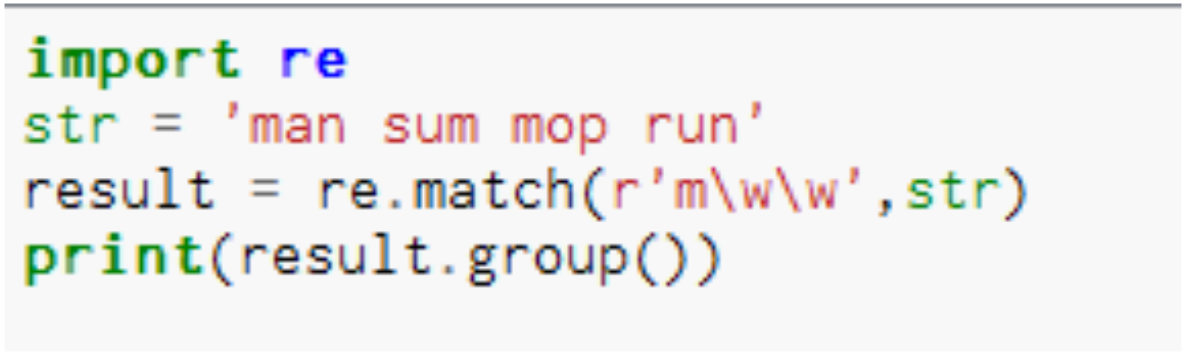


```
import re
str = 'man sun mop run'
result = re.findall(r'm\w\w',str)
for s in result:
    print(s)
```

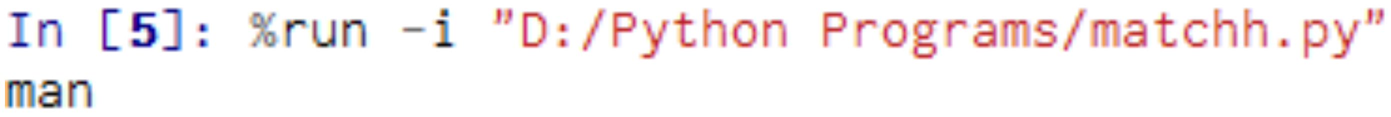
```
In [4]: %run -i "D:/Python Programs/finddalll.py"
man
mop
```

REGULAR EXPRESSIONS

- **match()** method that returns the resultant string only if it is found in the beginning of the string.
- The match() method will give None if the string is not in the beginning.



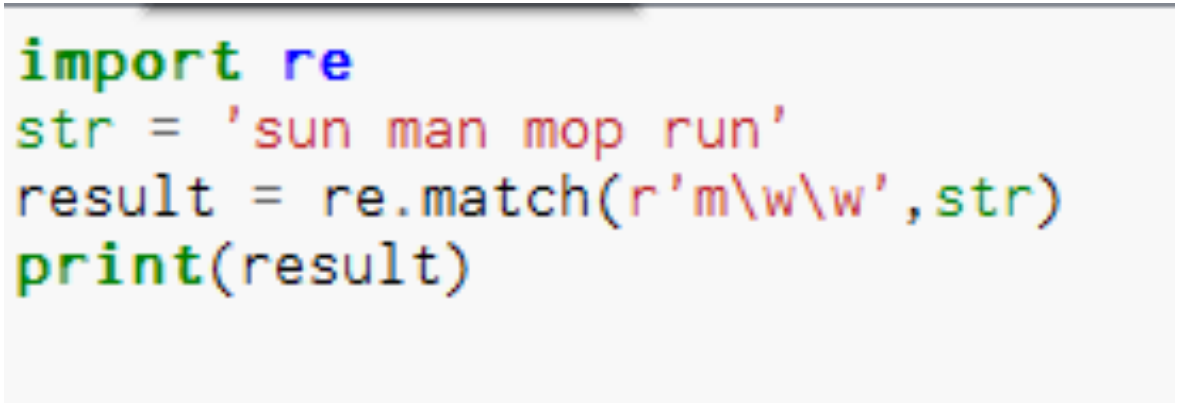
```python
import re
str = 'man sum mop run'
result = re.match(r'm\w\w',str)
print(result.group())
```



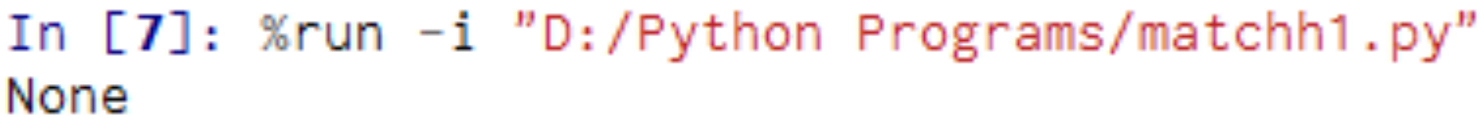
```
In [5]: %run -i "D:/Python Programs/matchh.py"
man
```

Regular expressions

- Python program to create a regular expression using match() method to search for strings starting with m and having total 3 characters.



```
import re
str = 'sun man mop run'
result = re.match(r'm\w\w',str)
print(result)
```



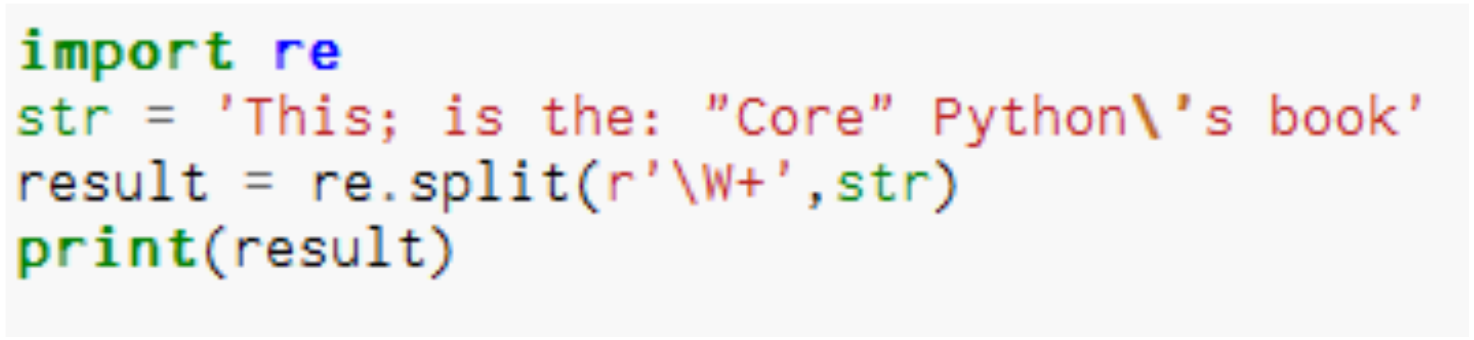
```
In [7]: %run -i "D:/Python Programs/matchh1.py"
None
```

Regular expressions

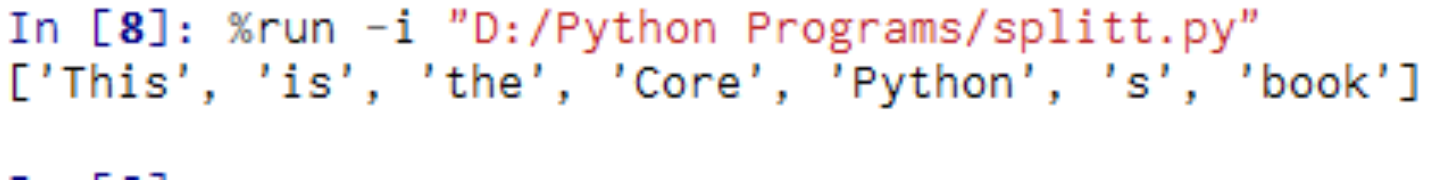
- split() method that splits the given string into pieces according to regular expression and returns the pieces as elements of a list.
  - re.split(r'\W+',str)
- 'W' represents any character that is not alpha numeric.
- So, this regular expression split the string where there is no alpha numeric character.
- The '+' after W represents to match 1 or more occurrences indicated by W.
- The result is that the string will be split into pieces where 1 or more non alpha numeric characters are found.

Regular expressions

- Python program to create a regular expression to split a string into pieces where one or more non alpha numeric characters are found.



```
import re
str = 'This; is the: "Core" Python\'s book'
result = re.split(r'\W+',str)
print(result)
```



```
In [8]: %run -i "D:/Python Programs/splitt.py"
['This', 'is', 'the', 'Core', 'Python', 's', 'book']
```

Regular expressions

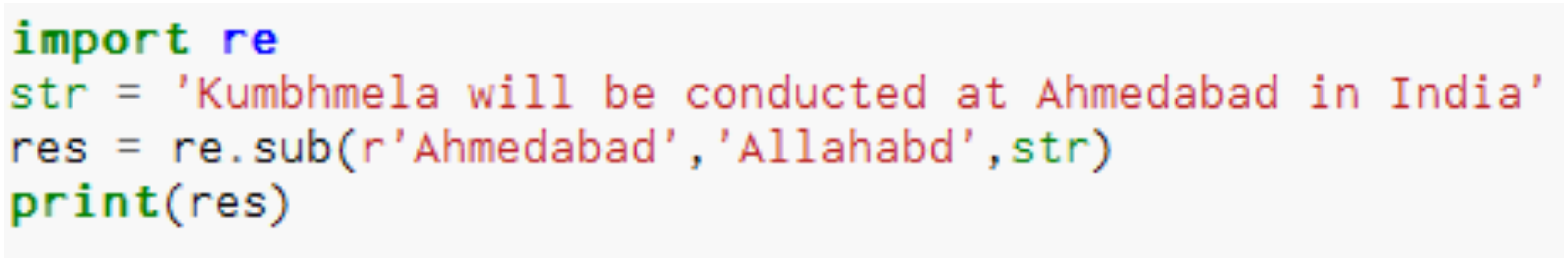
- Regular expression can also be used to find a string and then replace it with a new string.

- Use sub() method of ‘re’ module.
  - **sub(regular expression, new string, string)**

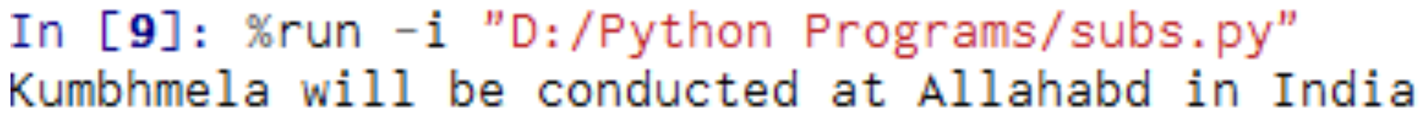
- Example, sub(‘Ahmedabad’, ‘Allahabad’ str) will replace ‘Ahmedabad’ with ‘Allahabad’ in the string ‘str’.

Regular expressions

- Python program, to create a regular expression to replace a string with a new string.



```python
import re
str = 'Kumbhmela will be conducted at Ahmedabad in India'
res = re.sub(r'Ahmedabad','Allahabd',str)
print(res)
```



```
In [9]: %run -i "D:/Python Programs/subs.py"
Kumbhmela will be conducted at Allahabd in India
```

Regular expressions

- Regular expressions are used to perform the following operations:
  - Matching strings
  - Searching for strings
  - Finding all Strings
  - Splitting a string into pieces
  - Replacing Strings.

REGULAR EXPRESSION

- The following methods belong to the ‘re’ module that are used in regular expressions:
  - **match():** Searches in the beginning of the string and if the matching string is found, it returns an object that contains the resultant string, otherwise it returns None. Access the string from the returned object using group() method.

  - **search():** Searches the string from beginning till the end and returns the first occurrence of the matching string, otherwise it returns None. Use group(0 method to retrieve the string from the object returned by this method.

REGULAR EXPRESSION

- **findall():** Searches the string from beginning till the end and returns all occurrences of the matching string in the form of a list object. If the matching strings are not found, then it returns an empty list. Retreive the resultant strings from the list using a for loop.
- **split():** Splits the string according to the regular expression and the resultant pieces are returned as a list. If there are no string pieces, then it returns an empty list. Retrieve the resultant string pieces from the list using for loop.
- **sub():** Substitutes or replaces new strings in the place of existing strings. After substitution, the main string is returned by this method.

SEQUENCE CHARACTERS IN REGULAR EXPRESSION

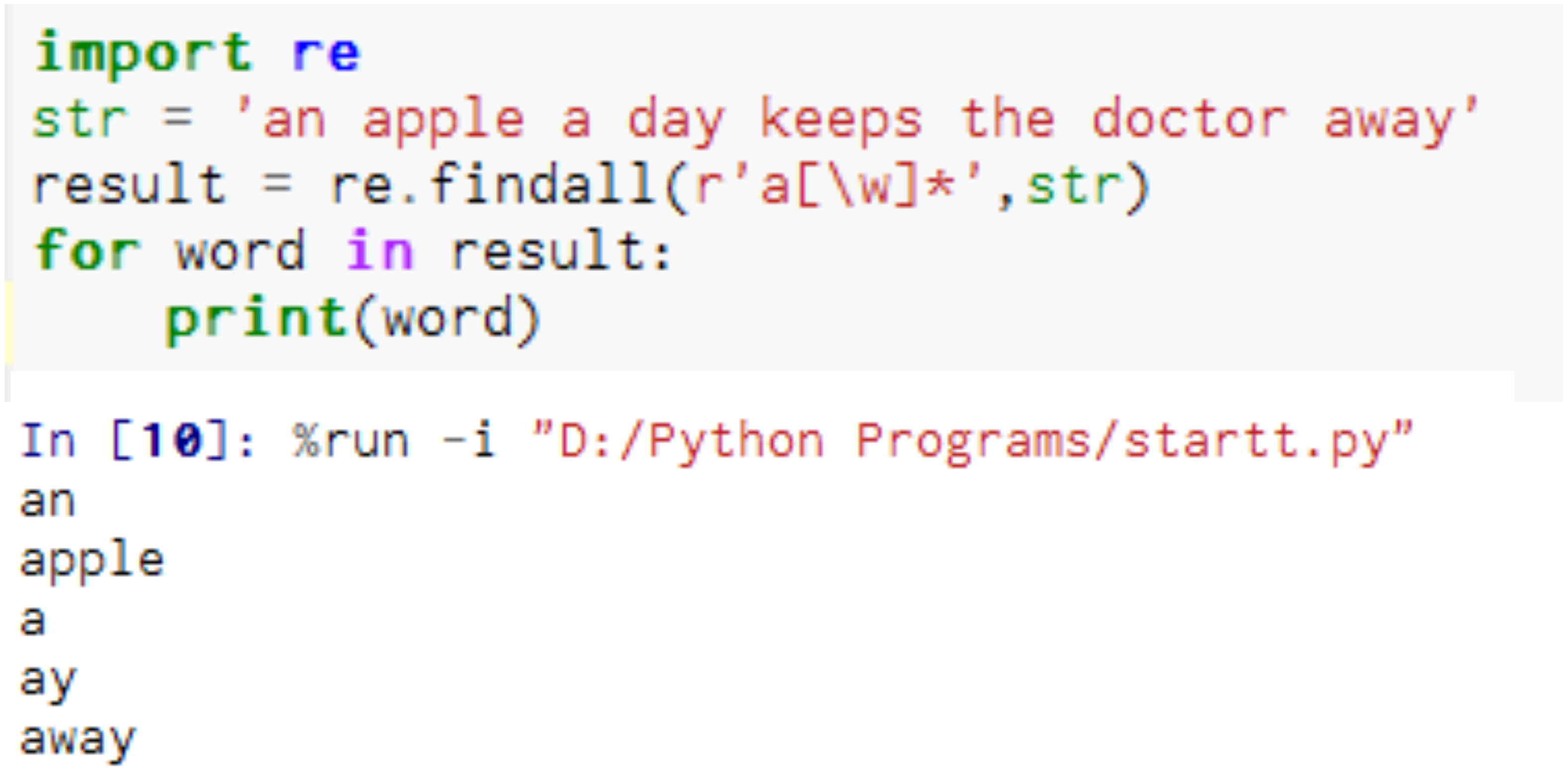
Character	Description
\d	Represents any digit (0 to 9)
\D	Represents any non digit
\s	Represents white space.Ex:\t \n \r \f\v
\S	Represents non-white space character.
\w	Represents any alphanumeric (A to Z, a to z, 0 to 9)
\W	Represents non-alphanumeric.
\b	Represents a space around words.
\A	Matches only at the start of the string.
\Z	Matches only at the end of the string.

SEQUENCE CHARACTERS IN REGULAR EXPRESSION

- '\w' indicates any one alphanumeric character.
- Suppose we write it as [\w]*.
- '*' represents 0 or more repetitions.
- [\w]* represents 0 or more alphanumeric characters.
- Regular expression to retrieve all words starting with 'a'.
  - **r'a[\w]*'**

Sequence characters in regular expression

- Python program to create a regular expression to retrieve all words starting with a in a given string.



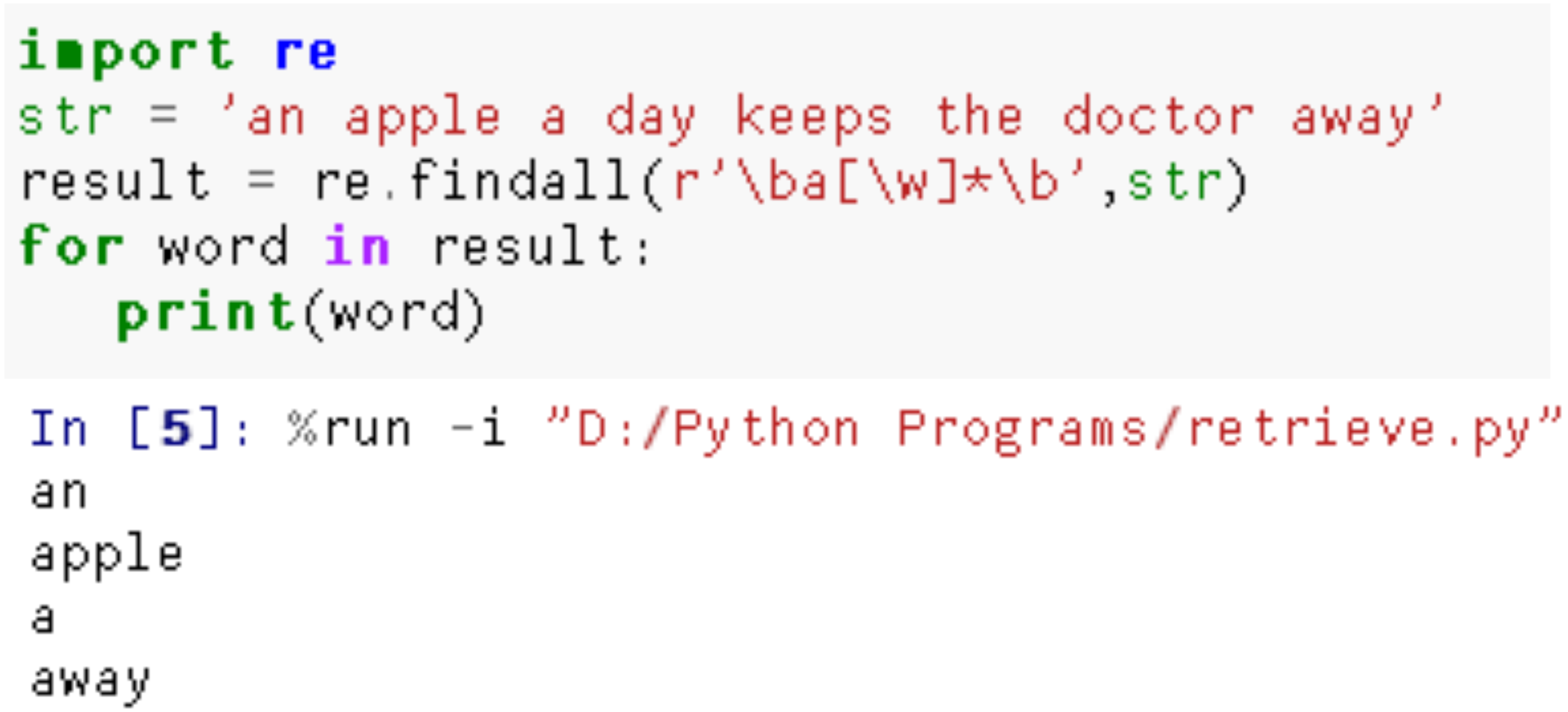
```python
import re
str = 'an apple a day keeps the doctor away'
result = re.findall(r'a[\w]*',str)
for word in result:
    print(word)
```

```
In [10]: %run -i "D:/Python Programs/startt.py"
an
apple
a
ay
away
```

SEQUENCE CHARACTERS IN REGULAR EXPRESSION

- The output contains ‘ay’ which is not a word.
- This ‘ay’ is part of word ‘away’.
- So, it is displaying both ‘ay’ and ‘away’ as they are starting with ‘a’.
- Since a word will have space in the beginning or ending, we can use ‘\b’ before and after the words in the regular expression.
  - **result = re.findall(r’\ba[\w]*\b’,str)**

Sequence characters in regular expression



```
import re
str = ’an apple a day keeps the doctor away’
result = re.findall(r’\ba[\w]*\b’,str)
for word in result:
   print(word)
```

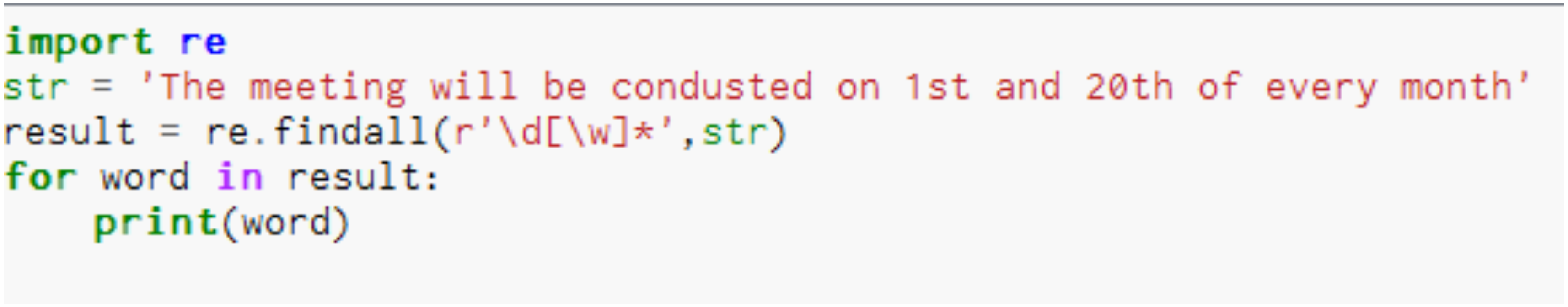
```
In [5]: %run -i ”D:/Python Programs/retrieve.py”
an
apple
a
away
```

Sequence characters in regular expression

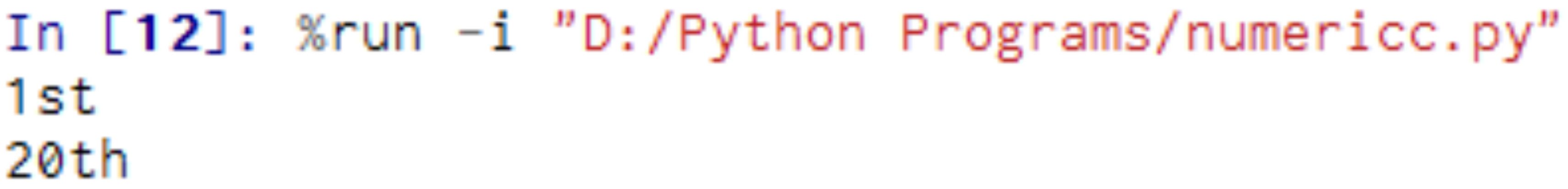
- To retrieve all the words starting with a numeric digit like 0, 1,2.
- The numeric digit is represented by '\d' and the expression will be r'\d[\w]*'

Sequence characters in regular expression

- Python program to create a regular expression to retrieve all words starting with a numeric digit.



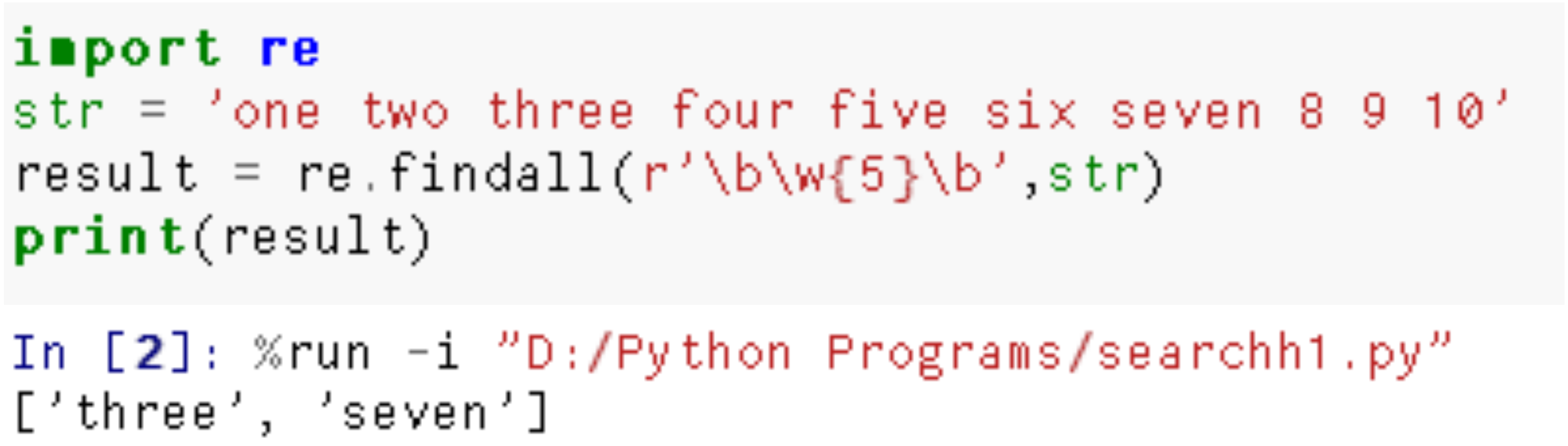
```python
import re
str = 'The meeting will be condusted on 1st and 20th of every month'
result = re.findall(r'\d[\w]*',str)
for word in result:
    print(word)
```



```
In [12]: %run -i "D:/Python Programs/numericc.py"
1st
20th
```

Sequence characters in regular expression

- Python program to create a regular expression to retrieve all words having 5 characters length.

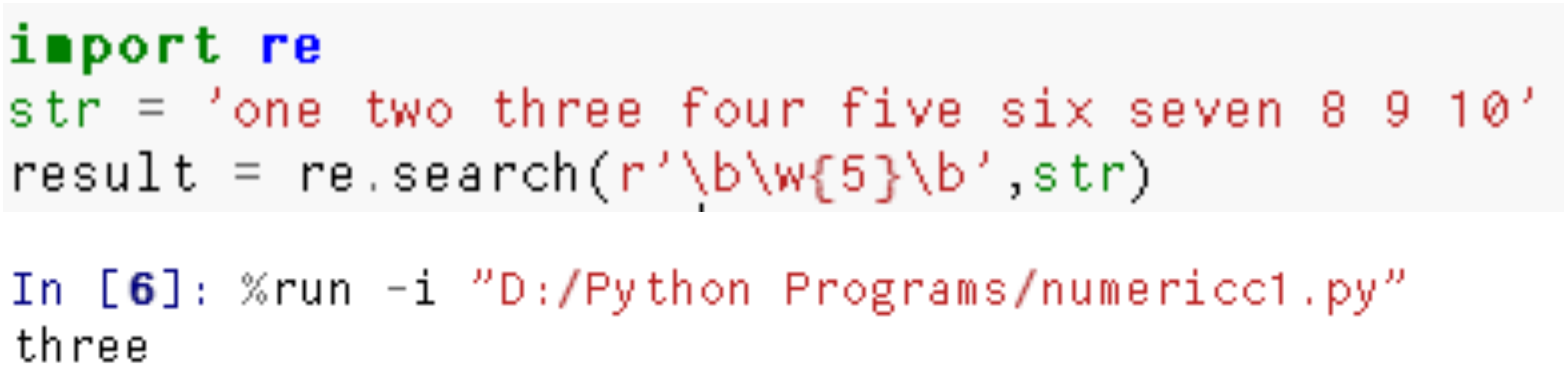


```
import re
str = ’one two three four five six seven 8 9 10’
result = re.findall(r’\b\w{5}\b’,str)
print(result)
```

```
In [2]: %run -i ”D:/Python Programs/searchh1.py”
[’three’, ’seven’]
```

Sequence characters in regular expression

- Python program to create a regular expression to retrieve all words having 5 characters length using search().

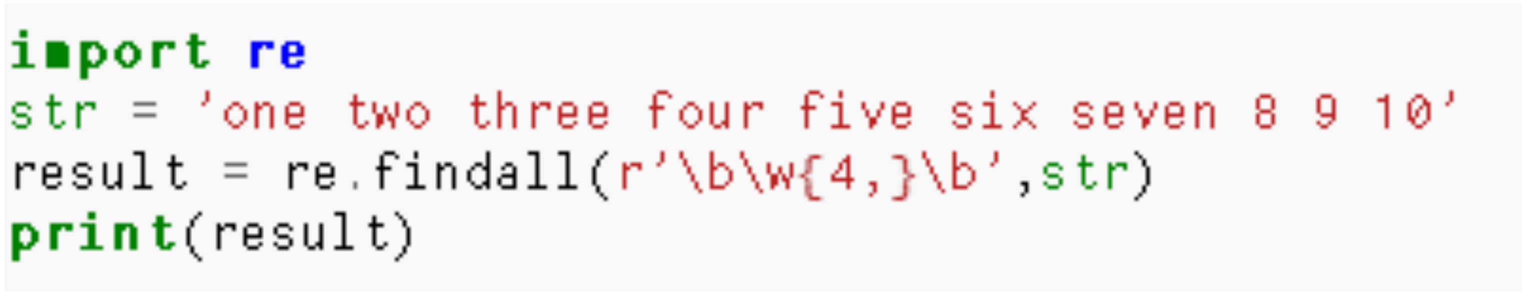


```
import re
str = 'one two three four five six seven 8 9 10'
result = re.search(r'\b\w{5}\b',str)
```

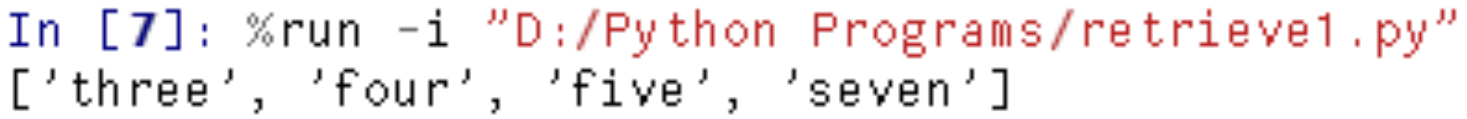
```
In [6]: %run -i "D:/Python Programs/numericc1.py"
three
```

SEQUENCE CHARACTERS IN REGULAR EXPRESSION

- Python program to create a regular expression to retrieve all the words that are having the length of at least 4 characters.



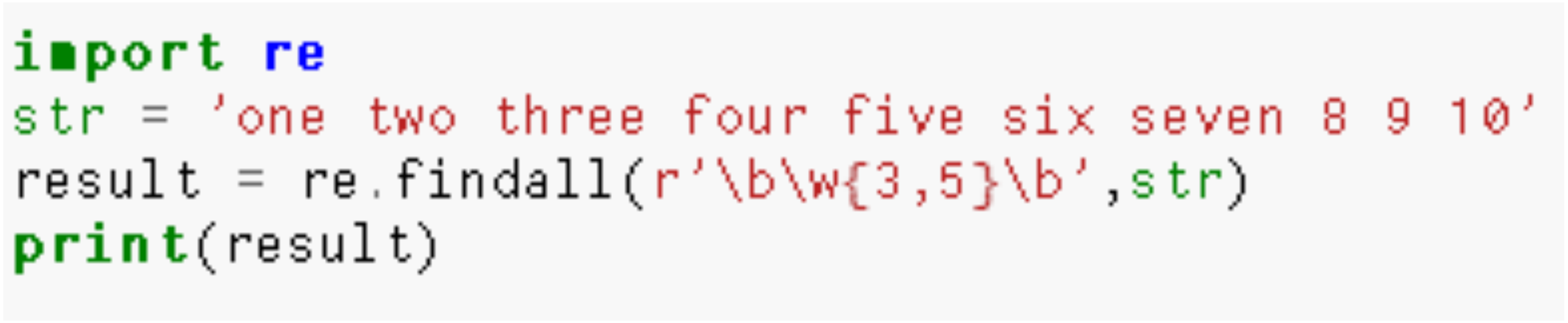
```
import re
str = 'one two three four five six seven 8 9 10'
result = re.findall(r'\b\w{4,}\b',str)
print(result)
```



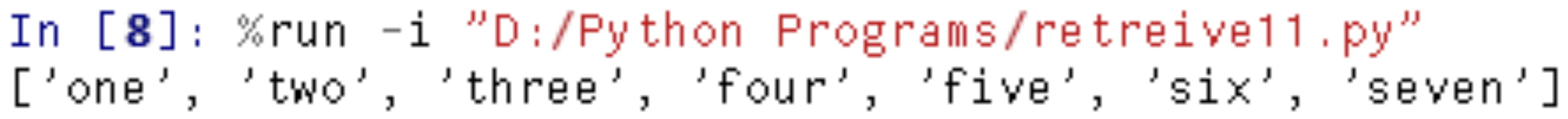
```
In [7]: %run -i "D:/Python Programs/retrieve1.py"
['three', 'four', 'five', 'seven']
```

Sequence characters in regular expression

- Python program to create a regular expression to retrieve all words with 3 or 4 or 5 characters length.



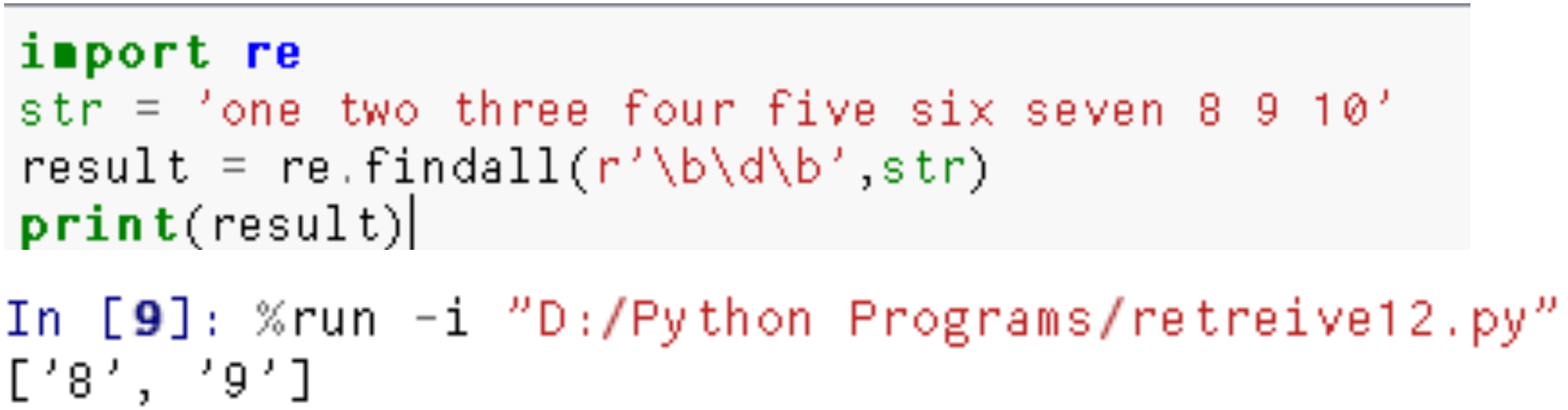
```
import re
str = ’one two three four five six seven 8 9 10’
result = re.findall(r’\b\w{3,5}\b’,str)
print(result)
```



```
In [8]: %run -i ”D:/Python Programs/retreive11.py”
[’one’, ’two’, ’three’, ’four’, ’five’, ’six’, ’seven’]
```

SEQUENCE CHARACTERS IN REGULAR EXPRESSION

- Python program to create a regular expression to retrieve only single digit from a string.

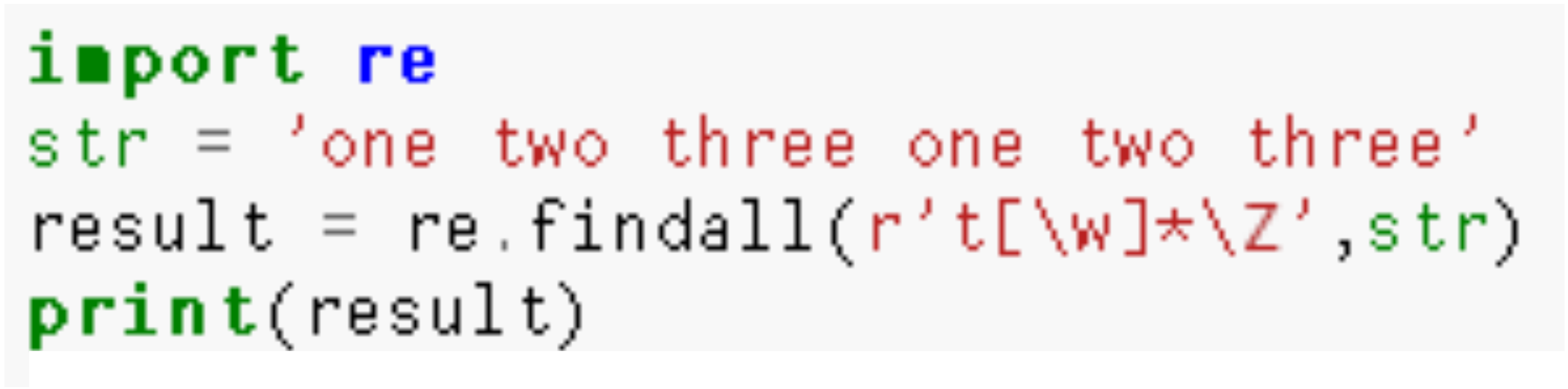


```
import re
str = 'one two three four five six seven 8 9 10'
result = re.findall(r'\b\d\b',str)
print(result)
```

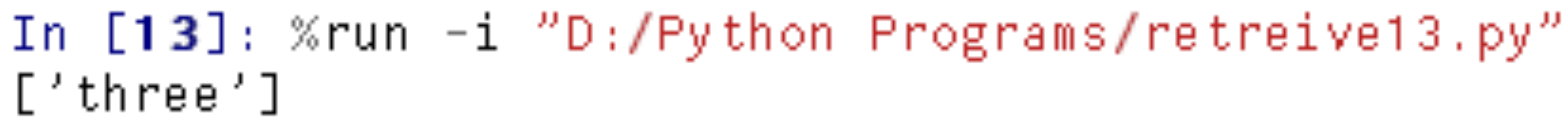
```
In [9]: %run -i "D:/Python Programs/retreive12.py"
['8', '9']
```

Sequence characters in regular expression

- Python program to create a regular expression to retrieve the last word of a string, if it starts with t.



```
import re
str = ’one two three one two three’
result = re.findall(r’t[\w]*\Z’,str)
print(result)
```



```
In [13]: %run -i ”D:/Python Programs/retreive13.py”
[’three’]
```

QUANTIFIERS IN REGULAR EXPRESSION

- In regular expressions, some characters represent more than one character to be matched in the string.

- Such characters are called ‘quantifiers’

- If we write ‘+’ it represents 1 or more repetitions of the preceding character.

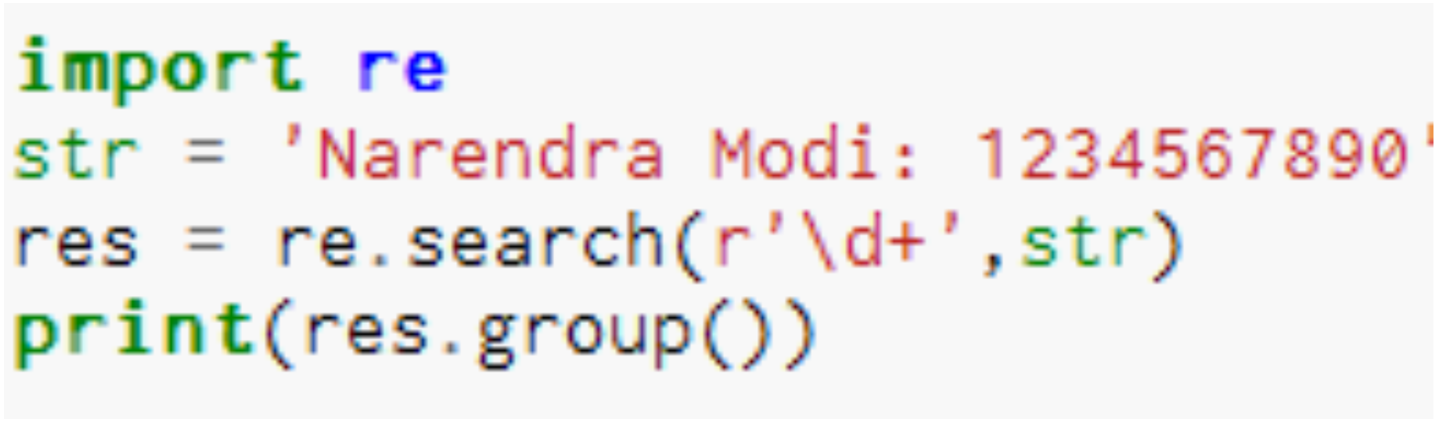
- If we write an expression as: r’\d+’ indicates that all numeric digits which occur for 1 or more times should be extracted.

Quantifiers in regular expression

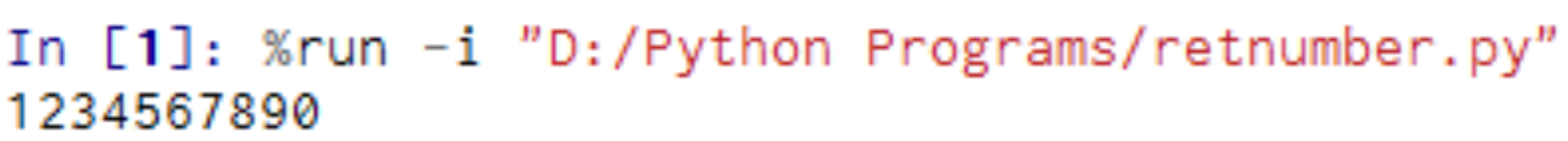
Character	Description
+	1 or more repetitions of the preceding regexp
*	0 or more repetitions of the preceding regexp
?	0 or 1 repetitions of the preceding regexp
{m}	Exactly m occurences
(m,n)	From m to n, m defaults to 0, n to infinity

Quantifiers in regular expression

- Python program to create a regular expression to retrieve the phone number of a person.



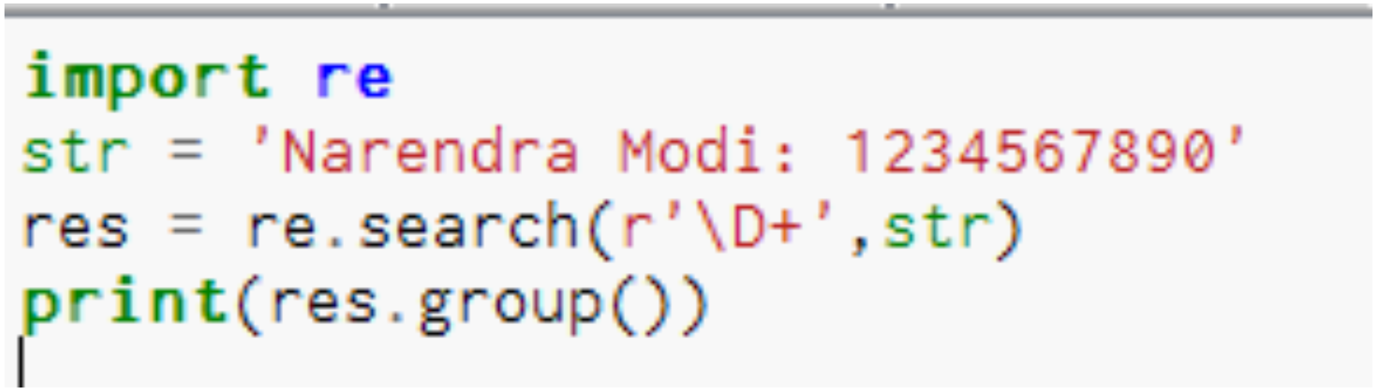
```python
import re
str = 'Narendra Modi: 1234567890'
res = re.search(r'\d+',str)
print(res.group())
```



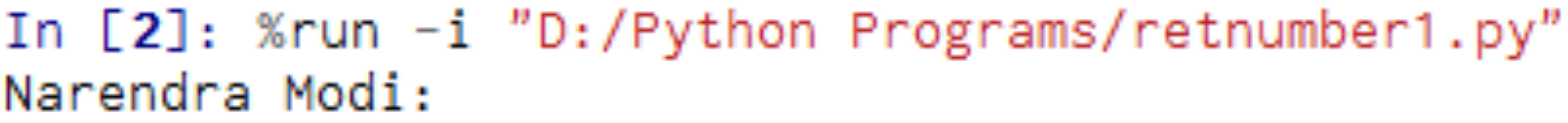
```
In [1]: %run -i "D:/Python Programs/retnumber.py"
1234567890
```

Quantifiers in regular expression

- Python program to create a regular expression to extract only name but not number from a string.



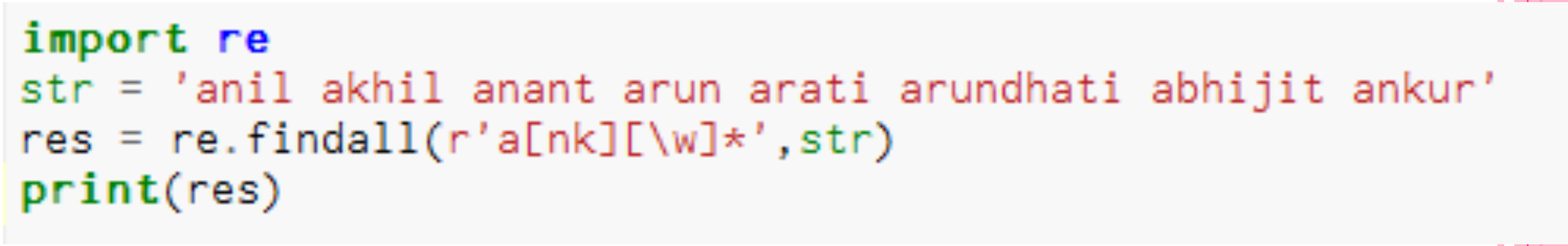
```python
import re
str = 'Narendra Modi: 1234567890'
res = re.search(r'\D+',str)
print(res.group())
```



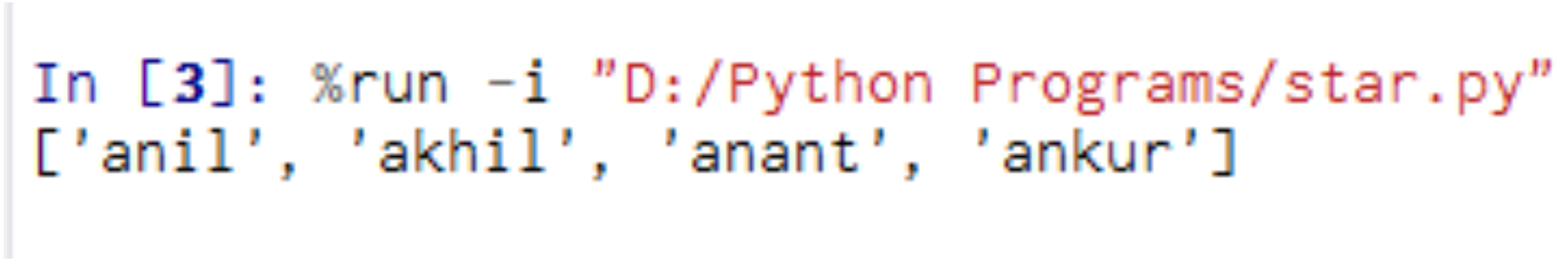
```
In [2]: %run -i "D:/Python Programs/retnumber1.py"
Narendra Modi:
```

Quantifiers in regular expression

- Python program to create a regular expression to find all words starting with ‘an’ or ‘ak’



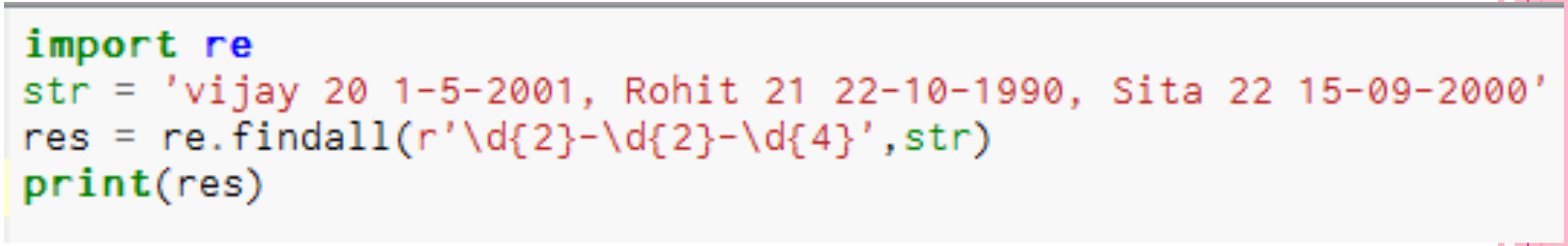
```
import re
str = 'anil akhil anant arun arati arundhati abhijit ankur'
res = re.findall(r'a[nk][\w]*',str)
print(res)
```



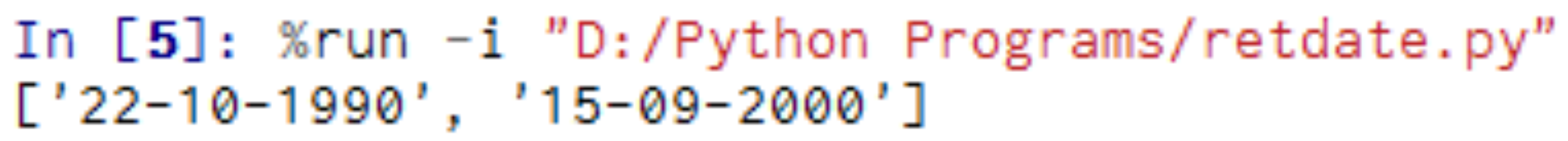
```
In [3]: %run -i "D:/Python Programs/star.py"
['anil', 'akhil', 'anant', 'ankur']
```

QUANTIFIERS IN REGULAR EXPRESSION

- Python program to create a regular expression to retrieve date of birth from a string.



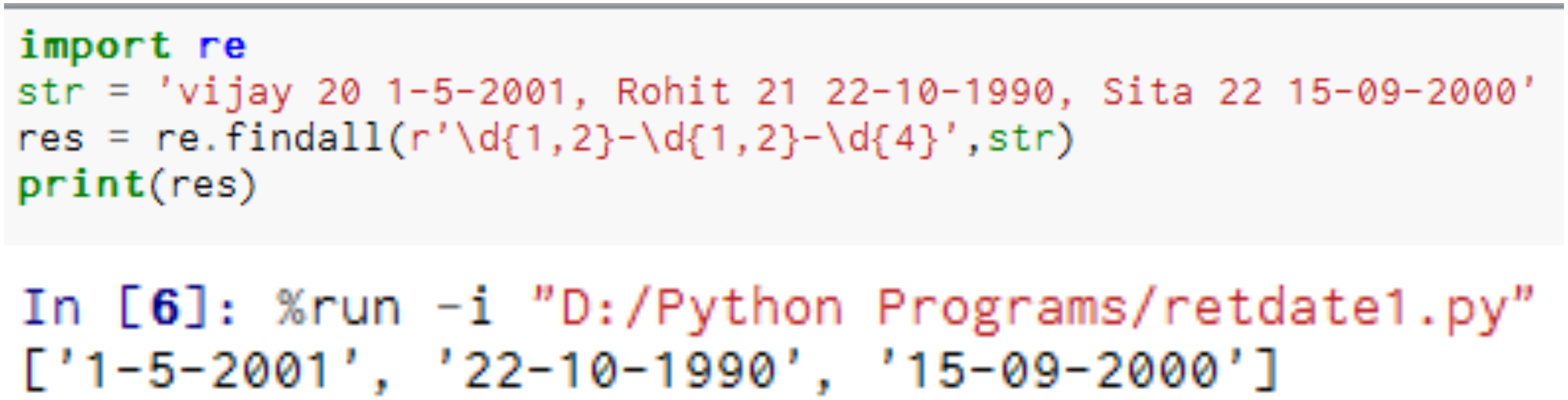
```
import re
str = 'vijay 20 1-5-2001, Rohit 21 22-10-1990, Sita 22 15-09-2000'
res = re.findall(r'\d{2}-\d{2}-\d{4}',str)
print(res)
```



```
In [5]: %run -i "D:/Python Programs/retdate.py"
['22-10-1990', '15-09-2000']
```

Quantifiers in regular expression

- Regular expression that retrieves either 1 or 2 digits



```
import re
str = 'vijay 20 1-5-2001, Rohit 21 22-10-1990, Sita 22 15-09-2000'
res = re.findall(r'\d{1,2}-\d{1,2}-\d{4}',str)
print(res)
```

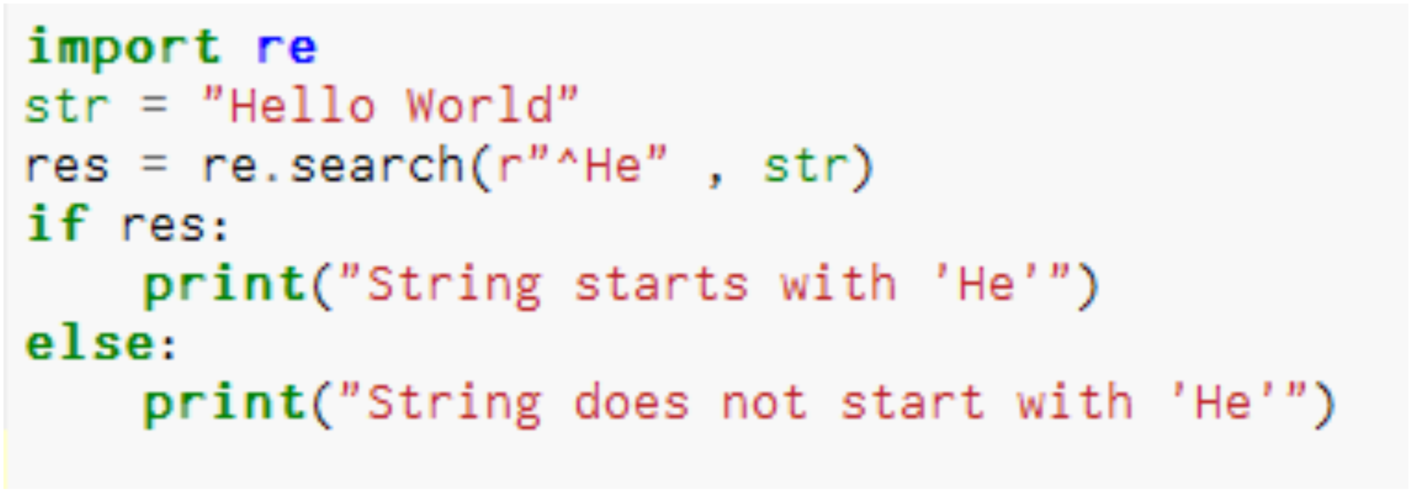
```
In [6]: %run -i "D:/Python Programs/retdate1.py"
['1-5-2001', '22-10-1990', '15-09-2000']
```

Special characters in regular expressions

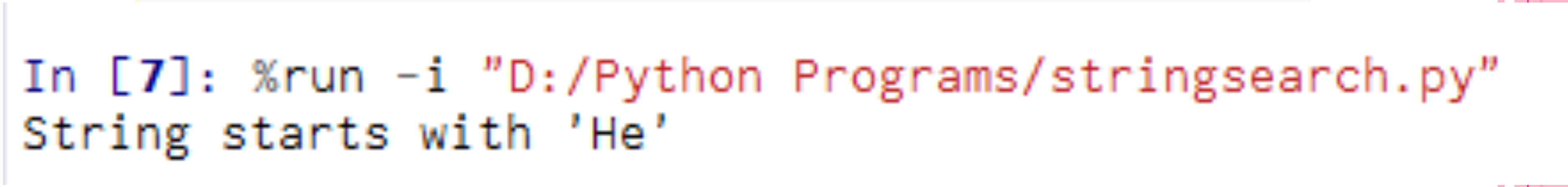
Character	Description
\	Escape special character nature.
.	Matches any character except new line.
^	Matches beginning of a string.
$	Matches ending of a string.
[…]	Denotes a set of possible characters . Eg: [6b-d] matches any character ‘6’,’b’,’c’ or ‘d’
[^…]	Matches every characters except the ones inside brackets. [^a-c6] matches any character except ‘a’,’b’,’c’ or ‘6’
(…)	Matches the regular expression inside the parentheses and the result can be captured.
R | S	Matches either regex R or regex S

SPECIAL CHARACTERS IN REGULAR EXPRESSIONS

- Python program to create a regular expression to search whether a given string is starting with 'He' or not.



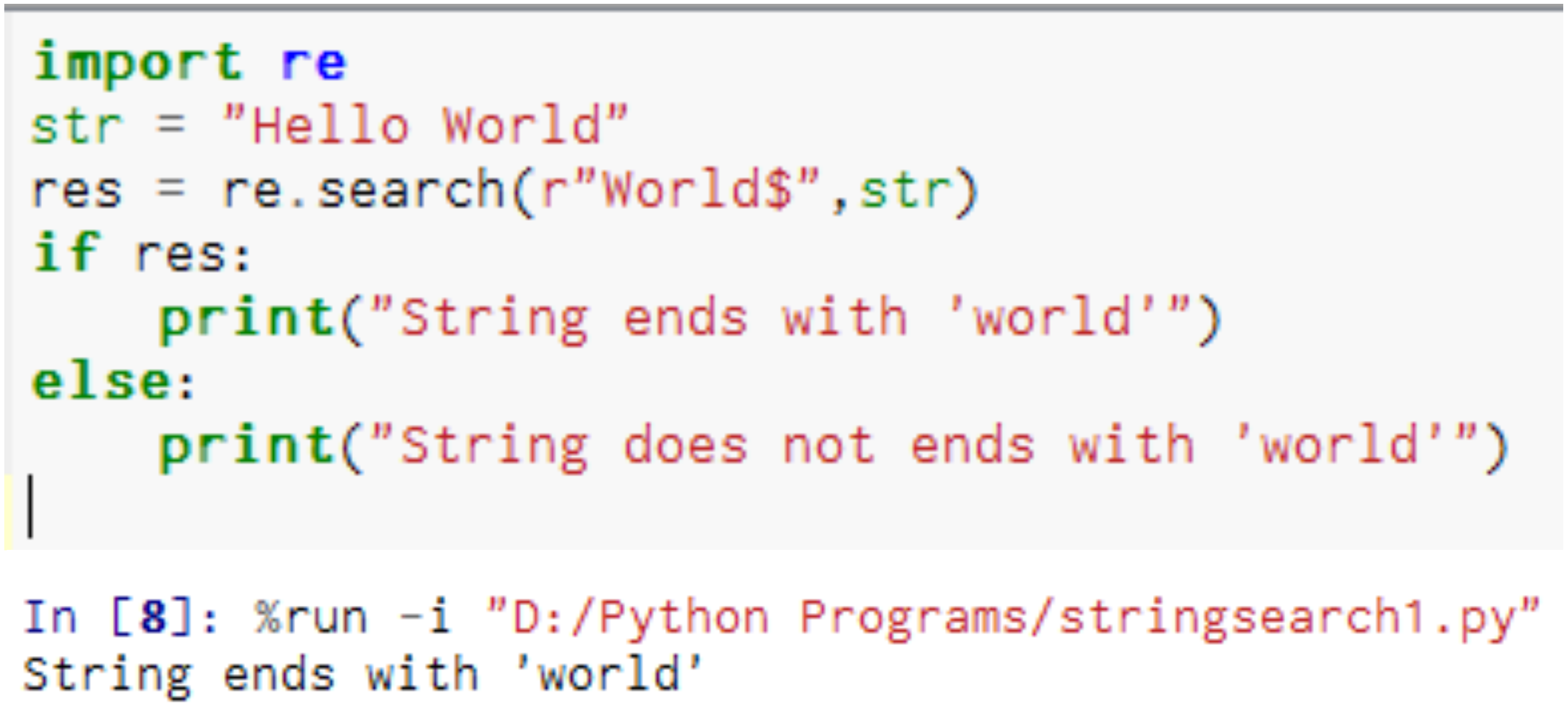
```
import re
str = "Hello World"
res = re.search(r"^He" , str)
if res:
    print("String starts with 'He'")
else:
    print("String does not start with 'He'")
```



```
In [7]: %run -i "D:/Python Programs/stringsearch.py"
String starts with 'He'
```

Special characters in regular expressions

- Python program to create a regular expression to search for a word at the ending of a string.

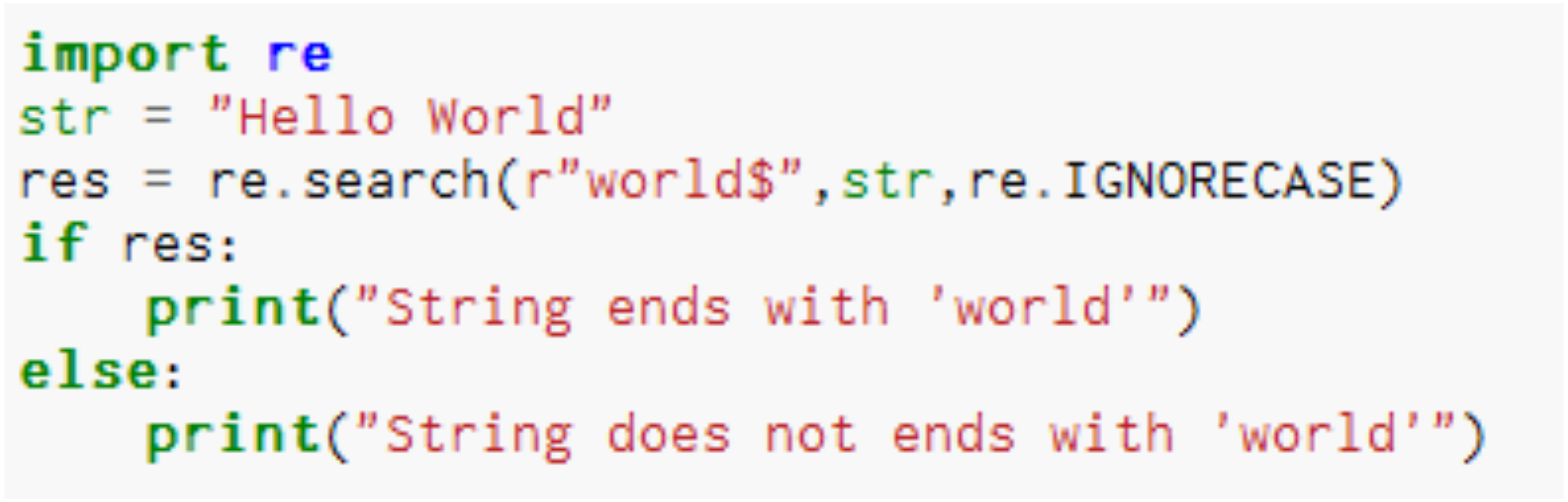


```python
import re
str = "Hello World"
res = re.search(r"World$",str)
if res:
    print("String ends with 'world'")
else:
    print("String does not ends with 'world'")
```

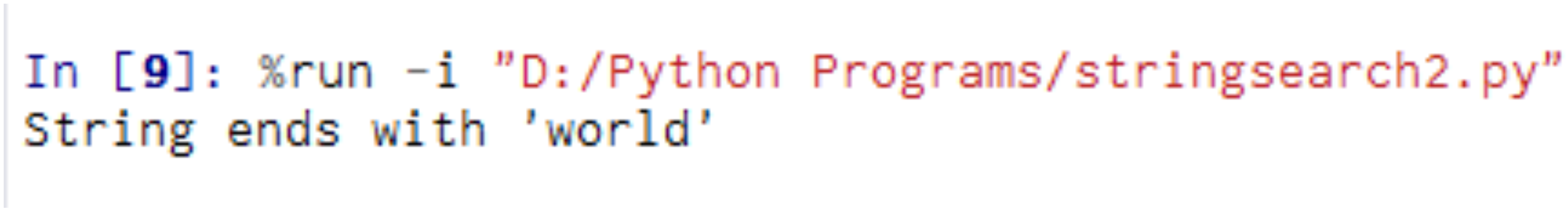
```
In [8]: %run -i "D:/Python Programs/stringsearch1.py"
String ends with 'world'
```

Special characters in regular expressions

- Python program to create a regular expression to search at the ending of a string by ignoring the case.



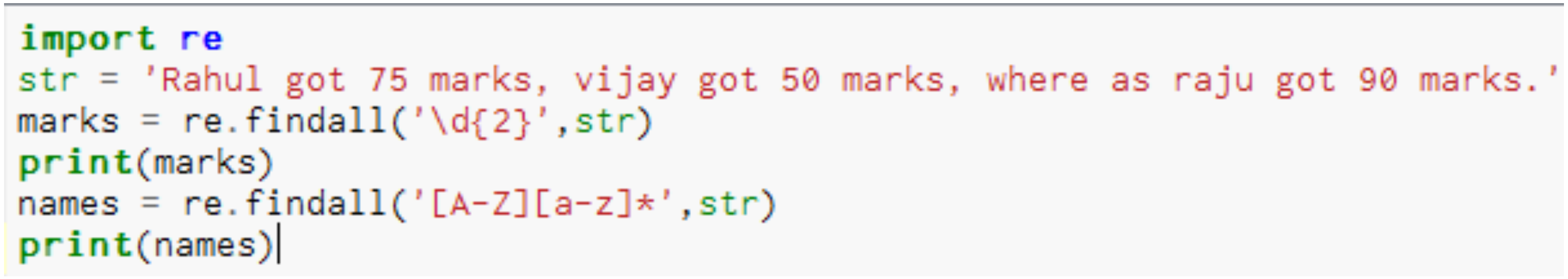
```
import re
str = "Hello World"
res = re.search(r"world$",str,re.IGNORECASE)
if res:
    print("String ends with 'world'")
else:
    print("String does not ends with 'world'")
```



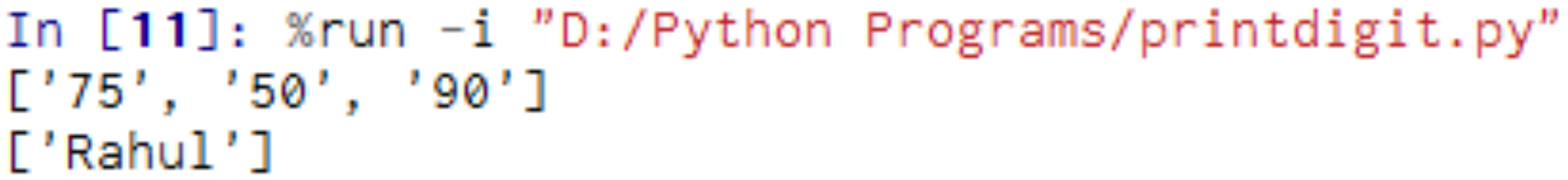
```
In [9]: %run -i "D:/Python Programs/stringsearch2.py"
String ends with 'world'
```

Special characters in regular expressions

- Python program to retrieve names and marks from a given string.



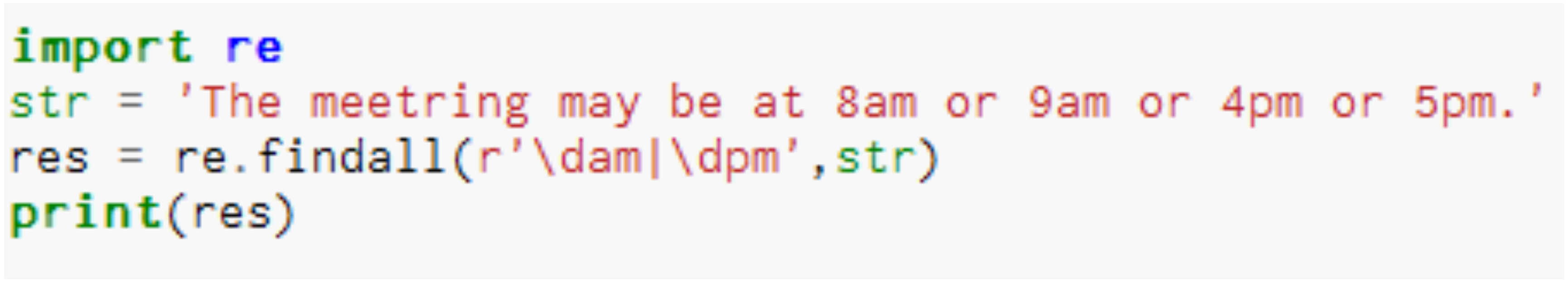
```
import re
str = 'Rahul got 75 marks, vijay got 50 marks, where as raju got 90 marks.'
marks = re.findall('\d{2}',str)
print(marks)
names = re.findall('[A-Z][a-z]*',str)
print(names)
```



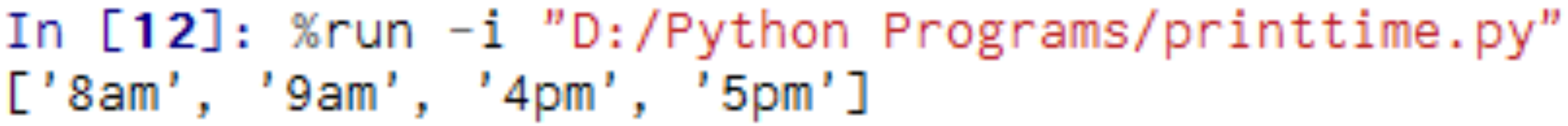
```
In [11]: %run -i "D:/Python Programs/printdigit.py"
['75', '50', '90']
['Rahul']
```

SPECIAL CHARACTERS IN REGULAR EXPRESSIONS

- Python program to create a regular expression to retrieve the timings either ‘am’ or ‘pm’.



```
import re
str = 'The meetring may be at 8am or 9am or 4pm or 5pm.'
res = re.findall(r'\dam|\dpm',str)
print(res)
```



```
In [12]: %run -i "D:/Python Programs/printtime.py"
['8am', '9am', '4pm', '5pm']
```