Chapter - 6 : Functions

String Pattern Matching

Regular Expression

- It is a special text string used for a search pattern. It is useful for extracting information from text like code, files, log, spreadsheets or even documents.
- Python has a built-in package called **re Module**, which helps to work with Regular Expressions.
- It is required to import the re Module before using it. e.g. **import re**

Regular Expression functions

- **match( )** checks for a match only at the beginning of the string.
- **search( )** checks for a match anywhere in the string.
- **findall( )** checks for all the match in the string and returns the list.
- **split( )** gives a list where the string has been split at each match.
- **sub( )** replaces one or many matches with a string.

Metacharacters

Metacharacters are characters with specific meaning.

Character	Description	Example
[ ]	A set of characters	"[a-m]"
\	Signals a special sequence (can also be used to escape special characters)	"\d"
.	Any character (except newline character)	"he..o"
^	Starts with	"^hello"
$	Ends with	"world$"
*	Zero or more occurrences	"aix*"
+	One or more occurrences	"aix+"
{ }	Exactly the specified number of occurrences	"al{2}"
|	Either or	"falls|stays"
( )	Capture and group	

Example

```
import re

#Check if the string starts with "The" and ends with "Spain":

txt = "The dog chase the cat"
x = re.search("^The.*cat$", txt)

if (x):
    print("String found!")
else:
   print("String not found")
```

Special Sequences

A special sequence is a \ followed by one of the characters.

Character	Description	Example
\A	Check if the specified characters are at the beginning of the string	"\AThe"
\b	Check the specified characters are at the beginning or at the end of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\bain" r"ain\b"
\B	Check the specified characters are present, but NOT at the beginning (or at the end) of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\Bain" r"ain\B"
\d	Checks the string contains digits (numbers from 0-9)	"\d"
\D	Checks the string DOES NOT contain digits	"\D"
\s	Checks the string contains a white space character	"\s"
\S	Checks the string DOES NOT contain a white space character	"\S"
\w	Checks the string contains any word characters (characters from a to z, digits from 0-9, and the underscore _ character)	"\w"
\W	Checks the string DOES NOT contain any word characters	"\W"
\Z	Checks the specified characters are at the end of the string	"Spain\Z"

Example

```
import re

txt = "The rain in Train"
x = re.search("ai", txt)
print(x)    #this will print an object
```

Output

```
<re.Match object; span=(5, 7), match='ai'>
```

Example

```
import re

#searches all the words starting with r
txt = "rain in train"
x = re.findall("r\w+", txt)
print(x)

#searches the occurrence of words starting with r, in the beginning of string.
txt = "rain in train"
x = re.match("r\w+", txt)
print(x.group())

#searches the occurrence of words starting with r, in the beginning of string.
txt = "pain in train"
x = re.match("r\w+", txt)
print(x)

#searches the occurrence of words starting with r, anywhere of string.
txt = "pain in train"
x = re.search("r\w+", txt)
print(x)

#split the string, wherever the occurrence of word found.
txt = "The rain in Spain"
x = re.split("ai", txt)
print(x)
```

Output

```
['rain', 'rain']
<re.Match object; span=(0, 4), match='rain'>
None
<re.Match object; span=(9, 13), match='rain'>
['The r', 'n in Sp', 'n']
```

Example

```
import re
#splits the string wherever the whitespace found

txt = "The rain in train"
x = re.split("\s", txt, 1)
print(x)

x = re.split("\s", txt, 2)
print(x)

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

Output

```
['The', 'rain in train']
['The', 'rain', 'in train']
['The', 'rain', 'in', 'train']
```