Regular Expression

```
import re
string = "The quick brown fox jumps over the lazy dog."
pattern="brown"
match=re.search(pattern, string) # returns True/False, used for validation
if match:
    print("Match found:", match.group())
else:
    print("Match not found")
```

```
Match found: brown
```

```
import re
string = "The quick brown fox jumps over the lazy dog."
pattern="The"
match=re.match(pattern, string)
if match:
    print("Match found:", match.group())
else:
    print("Match not found")
```

```
Match found: The
```

```
import re
string = "The quick brown fox jumps over the lazy dog."
pattern="[aeiou]"
matches=re.findall(pattern, string)
if matches:
    print("Match found:", matches)
else:
    print("Match not found")
```

```
Match found: ['e', 'u', 'i', 'o', 'o', 'u', 'o', 'e', 'e', 'a', 'o']
```

```
import re
string = "The quick brown fox jumps over the lazy dog."
pattern="brown"
new_str=re.sub(pattern, "red", string)
print(new_str)
print(string) # actual string
```

```
The quick red fox jumps over the lazy dog.
The quick brown fox jumps over the lazy dog.
```

```
import re
string = "The quick brown fox jumps over the lazy dog."
pattern="\s+"
split_str=re.split(pattern, string)
print(split_str)
```

```
['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog.']
```

Match any character: . (dot) Example: a.b matches "a b", "a1b", "a!b", etc.

Match the start of a string: ^ Example: ^hello matches "hello world", but not "world hello"

Match the end of a string: $Example:world$ matches "hello world", but not "world hello"

Match any single digit: \d Example: \d matches any digit from 0 to 9

Match any non-digit character: \D Example: \D matches any character that is not a digit

Match any whitespace character: \s Example: \s matches spaces, tabs, and newlines

Match any non-whitespace character: \S Example: \S matches any character that is not whitespace

Match any word character (letters, digits, or underscore): \w Example: \w+ matches one or more word characters (letters, digits, or underscore)

Match any non-word character: \W Example: \W matches any character that is not a word character

Match zero or one occurrence of the previous character: ? Example: colou?r matches "color" and "colour"

Match zero or more occurrences of the previous character: * Example: ab*c matches "ac", "abc", "abbc", etc.

Match one or more occurrences of the previous character: + Example: ab+c matches "abc", "abbc", "abbbc", etc.

Match a specific number of occurrences of the previous character: {m} Example: a{3} matches "aaa"

Match a range of occurrences of the previous character: {m,n} Example: a{2,4} matches "aa", "aaa", and "aaaa"

Match any one of a set of characters: [] Example: [abc] matches "a", "b", or "c"

Match any character except those in a set: [^] Example: [^abc] matches any character except "a", "b", or "c"

Match the same characters as a previously captured group: \1, \2, etc. Example: (\w) is \1 matches "a is a", "b is b", etc.

These are just some of the most common regular expressions used in Python. There are many more advanced regular expressions available as well.

validation through Regular Expression (RE)

1. re.search(pattern, string) ---- to check if pattern is present or net
2. re.findall(pattern, string) ---- to find all occurences of pattern in string
3. re.sub(pattern, newpattern, inp) to replace string
4.

```
import re
p='[a-z]'
string=input("Enter data to check: ")
if re.search(p, string):
    print("yes")
else:
    print("No")

p2='\d'
string=input("Enter data to check: ")
```

```
if re.search(p2, string):
    print("yes")
else:
    print("No")
```

```
Enter data to check: amar123
yes
Enter data to check: amar123panchal
yes
```

```
import re
p='[a-zA-Z0-9]'
string=input("Enter data to check: ")
if re.search(p, string):
    print("yes")
else:
    print("No")
```

```
Enter data to check: Amar123panchal
yes
```

```
import re
p='(.com|.co.in|.in|man)'
i=input("Enter data to check: ")
if re.search(p, i):
    print("yes")
else:
    print("No")
```

```
Enter data to check: google.com
yes
```

```
import re
p='[a-zA-Z0-9]+\.' # seaches for dot at end
string=input("Enter data to check: ")
if re.search(p, string):
    print("yes")
else:
    print("No")
```

```
Enter data to check: asds121cs.
yes
```

```
import re
p='[0-9]+\.+[0-9]+\.+[0-9]+\.+[0-9]' # IP validation
string=input("Enter data to check: ")
if re.search(p, string):
    print("yes")
else:
    print("No")
```

```
Enter data to check: 123amar.hehe.gone
No
```

```
import re
p='[a-zA-Z0-9]+\@+[a-zA-Z]+\.+[a-zA-Z]' # email validation
string=input("Enter data to check: ")
if re.search(p, string):
    print("yes")
else:
    print("No")
```

```
Enter data to check: sachin@iyla.com
yes
```

```
import re
p='\d{10}' # mobile validation
i=input("Enter data to check: ")
if re.search(p, i):
    print("yes")
else:
    print("No")
```

```
Enter data to check: 8787878787
yes
```

```
import re
p='^\d{2}' # starts with 2
i=input("Enter data to check: ")
if re.search(p, i):
    print("yes")
else:
    print("No")
```

```
Enter data to check: 2656
yes
```

```
import re
p='\d{2}$' # ends with 2
i=input("Enter data to check: ")
if re.search(p, i):
    print("yes")
else:
    print("No")
```

```
Enter data to check: 5662
yes
```

```
import re
p='\d{2}$' #
i=input("Enter data to check: ")
if re.search(p, i):
    print("yes")
else:
    print("No")
```

Extraction

```
pattern="[\W]+" #word separated
inp=input("Enter data to check: ")
data=re.findall(pattern, inp)
print("Final result: ", data)
```

```
Enter data to check: ajkda 789 jnk** k*(
Final result:  [' ', ' ', '** ', '*(']
```

```
pattern="\+(\d{2})-(\d{4})-(\d{6})" # pattern for telephone number
newpattern=r"Country:\1 Area:\2 Number\3" # r"" is raw string, will replace the matching pattern
inp=input("Enter data to check: ")
newstr=re.sub(pattern, newpattern, inp)
print("Final result: ", newstr)
```

```
Enter data to check: +91-1923-220007
Final result:  Country:91 Area:1923 Number220007
```

Double-click (or enter) to edit

```
import re
log='''192.168.198.92 - - [22/Dec/2002:23:08:37 -0400] "GET
   / HTTP/1.1" 200 6394 www.yahoo.com
   "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1...)" "-"
192.168.198.92 - - [22/Dec/2002:23:08:38 -0400] "GET
   /images/logo.gif HTTP/1.1" 200 807 www.yahoo.com
   "http://www.some.com/" "Mozilla/4.0 (compatible; MSIE 6...)" "-"
192.168.72.177 - - [22/Dec/2002:23:32:14 -0400] "GET
   /news/sports.html HTTP/1.1" 200 3500 www.yahoo.com
   "http://www.some.com/" "Mozilla/4.0 (compatible; MSIE ...)" "-"
192.168.72.177 - - [22/Dec/2002:23:32:14 -0400] "GET
   /favicon.ico HTTP/1.1" 404 1997 www.yahoo.com
   "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; rv:1.7.3)..." "-"
192.168.72.177 - - [22/Dec/2002:23:32:15 -0400] "GET
   /style.css HTTP/1.1" 200 4138 www.yahoo.com
   "http://www.yahoo.com/index.html" "Mozilla/5.0 (Windows..." "-"
192.168.72.177 - - [22/Dec/2002:23:32:16 -0400] "GET
   /js/ads.js HTTP/1.1" 200 10229 www.yahoo.com
   "http://www.search.com/index.html" "Mozilla/5.0 (Windows..." "-"
192.168.72.177 - - [22/Dec/2002:23:32:19 -0400] "GET
   /search.php HTTP/1.1" 400 1997 www.yahoo.com
   "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; ...)" "-"'''

```

```
pattern="[0-9]+\.[0-9]+\.[0-9]+\.[0-9]"
iplist=re.findall(pattern, log)
print(iplist)
```

```
['192.168.198.9', '192.168.198.9', '192.168.72.1', '192.168.72.1', '192.168.72.1', '192.168.72.1', '192.168.72.1']
```

```
pattern="[0-9]{2}/[a-zA-Z]{3}/[0-9]{4}"
datelist=re.findall(pattern, log)
```

```
print(datelist)
```

```
['22/Dec/2002', '22/Dec/2002', '22/Dec/2002', '22/Dec/2002', '22/Dec/2002', '22/Dec/2002', '22/Dec/2002']
```

```
pattern="\d+/\w{3}/\d+"
datelist=re.findall(pattern, log)
print(datelist)
```

```
['22/Dec/2002', '22/Dec/2002', '22/Dec/2002', '22/Dec/2002', '22/Dec/2002', '22/Dec/2002', '22/Dec/2002']
```

Web-Scraping

```
from bs4 import BeautifulSoup
import requests
f=requests.get("https://www.timesjobs.com/candidate/job-search.html?searchType=personalizedSearch&from=submit&txtKeywords=python&txtLocation=mumbai&cboWorkExp1=0").text

soup=BeautifulSoup(f, 'lxml')
print(soup)
```

```
<!DOCTYPE html>
<html><head>
<link href="https://fonts.googleapis.com/css?family=Poppins:400,500,600,700" rel="stylesheet"/>
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet"/>
<link href="https://static.timesjobs.com/newtj_css//css/tj_css/usability/materialize.css?v=7.1.7" media="all" rel="stylesheet" type="text/css"/>
<link href="https://static.timesjobs.com/newtj_css//css/tj_css/usability/global-usability.css?v=7.1.7" media="all" rel="stylesheet" type="text/css"/>
<link href="https://static.timesjobs.com/newtj_css//css/tj_css/usability/srp-usability.css?v=7.1.7" media="all" rel="stylesheet" type="text/css"/>
<script src="https://static.timesjobs.com/newtj_js/scripts/tj_scripts/usability/jquery-3.3.1.min.js" type="text/javascript"></script>
<script src="https://static.timesjobs.com/newtj_js/scripts/jquery.tokeninput.js" type="text/javascript"></script>
<!-- <script type="text/javascript" src="https://code.angularjs.org/1.7.0/angular.min.js"></script> -->
<script src="https://static.timesjobs.com/newtj_js/scripts/tj_scripts/usability/angular1.7.0.min.js" type="text/javascript"></script>
<script src="https://static.timesjobs.com/newtj_js/scripts/tj_scripts/usability/materialize.min.js" type="text/javascript"></script>
<script src="https://static.timesjobs.com/newtj_js/scripts/groupJs/dwrGroup.js?v=7.1.7" type="text/javascript"></script>
<script src="https://static.timesjobs.com/newtj_js/scripts/tj_scripts/mustache.js" type="text/javascript"></script>
<link href="https://www.timesjobs.com/candidate/job-search.html" rel="canonical"/>
<link href="https://www.timesjobs.com/candidate/job-search.html?searchType=personalizedSearch&amp;from=submit&amp;txtKeywords=python&amp;txtLocation=mumbai&amp;cboWorkExp1=0/&amp;sequence=2&amp
<title>python Jobs in mumbai  with Entry Level experience - TimesJobs</title>
<meta content="Jobs in python in mumbai  with Entry Level experience. Search and apply jobs online at Timesjobs.com" name="description"/>
<!-- Google Tag Manager -->
<script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':
new Date().getTime(),event:'gtm.js'});var f=d.getElementsByTagName(s)[0],
j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';j.async=true;j.src=
'https://www.googletagmanager.com/gtm.js?id='+i+dl;f.parentNode.insertBefore(j,f);
})(window,document,'script','dataLayer','GTM-59SJDG4');</script>
<!-- End Google Tag Manager -->
<script src="https://www.gstatic.com/charts/loader.js" type="text/javascript"></script>
<script>
          //google.charts.load('current', {'packages':['bar']});
          google.charts.load('current', {packages: ['corechart', 'bar']});
        </script>
<script>!function(e){var n="https://s.go-mpulse.net/boomerang/";if("False"=="True")e.BOOMR_config=e.BOOMR_config||{},e.BOOMR_config.PageParams=e.BOOMR_config.PageParams||{},e.BOOMR_config.PageP
<body>
<div id="site">
<!-- Header Starts -->
<!-- Google Tag Manager (noscript) -->
<noscript>
<iframe height="0" src="https://www.googletagmanager.com/ns.html?id=GTM-59SJDG4" style="display:none;visibility:hidden" width="0"></iframe>
```

```
</noscript>
<!-- End Google Tag Manager (noscript) -->
<meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
<script src="https://wchat.freshchat.com/js/widget.js" type="text/javascript"></script>
<script>
window.fcWidget.init({
token: "d7130618-a917-400b-804c-bdf44d8d178b",
host: "https://wchat.freshchat.com",
config: {
headerProperty: {
hideChatButton: true
}
        }
});

</script>
<!--[if IE]>
<script src="scripts/html5shiv.js"></script>
<![endif]-->
```

```
from bs4 import BeautifulSoup
import requests
f=requests.get("https://www.timesjobs.com/candidate/job-search.html?searchType=personalizedSearch&from=submit&txtKeywords=python&txtLocation=mumbai&cboWorkExp1=0").text

soup=BeautifulSoup(f, 'lxml')
soup.prettify()
#print(soup)
```

```
'<!DOCTYPE html>\n<html>\n <head>\n  <link href="https://fonts.googleapis.com/css?family=
Poppins:400,500,600,700" rel="stylesheet"/>\n  <link href="https://fonts.googleapis.com/i
con?family=Material+Icons" rel="stylesheet"/>\n  <link href="https://static.timesjobs.co
m/newtj_css//css/tj_css/usability/materialize.css?v=7.1.7" media="all" rel="stylesheet" t
ype="text/css"/>\n  <link href="https://static.timesjobs.com/newtj_css//css/tj_css/usabil
ity/global-usability.css?v=7.1.7" media="all" rel="stylesheet" type="text/css"/>\n  <link
```

```
from bs4 import BeautifulSoup
import requests
f=requests.get("https://www.timesjobs.com/candidate/job-search.html?searchType=personalizedSearch&from=submit&txtKeywords=python&txtLocation=mumbai&cboWorkExp1=0").text

soup=BeautifulSoup(f, 'lxml')
soup.prettify().split()

#print(soup)
```

```
['<!DOCTYPE',
 'html>',
 '<html>',
 '<head>',
 '<link',
 'href="https://fonts.googleapis.com/css?family=Poppins:400,500,600,700"',
 'rel="stylesheet"/>',
 '<link',
 'href="https://fonts.googleapis.com/icon?family=Material+Icons"',
 'rel="stylesheet"/>',
 '<link',
 'href="https://static.timesjobs.com/newtj_css//css/tj_css/usability/materialize.css?v=7.1.7"',
 'media="all"',
```

```
 'rel="stylesheet"',
 'type="text/css"/>',
 '<link',
 'href="https://static.timesjobs.com/newtj_css//css/tj_css/usability/global-usability.css?v=7.1.7"',
 'media="all"',
 'rel="stylesheet"',
 'type="text/css"/>',
 '<link',
 'href="https://static.timesjobs.com/newtj_css//css/tj_css/usability/srp-usability.css?v=7.1.7"',
 'media="all"',
 'rel="stylesheet"',
 'type="text/css"/>',
 '<script',
 'src="https://static.timesjobs.com/newtj_js/scripts/tj_scripts/usability/jquery-3.3.1.min.js"',
 'type="text/javascript">',
 '</script>',
 '<script',
 'src="https://static.timesjobs.com/newtj_js/scripts/jquery.tokeninput.js"',
 'type="text/javascript">',
 '</script>',
 '<!--',
 '<script',
 'type="text/javascript"',
 'src="https://code.angularjs.org/1.7.0/angular.min.js"></script>',
 '-->',
 '<script',
 'src="https://static.timesjobs.com/newtj_js/scripts/tj_scripts/usability/angular1.7.0.min.js"',
 'type="text/javascript">',
 '</script>',
 '<script',
 'src="https://static.timesjobs.com/newtj_js/scripts/tj_scripts/usability/materialize.min.js"',
 'type="text/javascript">',
 '</script>',
 '<script',
 'src="https://static.timesjobs.com/newtj_js/scripts/groupJs/dwrGroup.js?v=7.1.7"',
 'type="text/javascript">',
 '</script>',
 '<script',
 'src="https://static.timesjobs.com/newtj_js/scripts/tj_scripts/mustache.js"',
 'type="text/javascript">',
 '</script>',
 '<link',
 'href="https://www.timesjobs.com/candidate/job-search.html"',
```

```
from bs4 import BeautifulSoup

html_doc = """
<html>
<head>
<title>My Website</title>
</head>
<body>
<div id="content">
<h1>Welcome to my website!</h1>
<p>This is a paragraph of text.</p>
<a href="http://www.google.com">Google</a>
</div>
</body>
```

```
</html>
"""
soup=BeautifulSoup(html_doc, 'html.parser')
div_tag=soup.find('div', {'id':'content'})
print(div_tag)
```

```
<div id="content">
<h1>Welcome to my website!</h1>
<p>This is a paragraph of text.</p>
<a href="http://www.google.com">Google</a>
</div>
```

```
from bs4 import BeautifulSoup
html_doc = """
<html>
<head>
<title>My Website</title>
</head>
<body>
<div class="article">
<h2>Article Title</h2>
<p>This is the first paragraph.</p>
<p>This is the second paragraph.</p>
</div>
<div class="article">
<h2>Another Article Title</h2>
<p>This is another paragraph.</p>
</div>
</body>
</html>
"""
soup=BeautifulSoup(html_doc, 'html.parser')
h2tags=soup.find_all('h2')
for h2tag in h2tags:
    print(h2tag)
```

```
<h2>Article Title</h2>
<h2>Another Article Title</h2>
```

```

```