```
import re
txt = "The hud hud cyclone in coastal areas"
x = re.search("^The.*areas$",txt)
if x :
  print("we have a match")
else :
  print("we dont have match")
```

```
we have a match
```

```
import re
txt = "THE HUD HUD cyclone in coastal areas"
x = re.findall("[a-z]", txt)
print(x)
```

```
['c', 'y', 'c', 'l', 'o', 'n', 'e', 'i', 'n', 'c', 'o', 'a', 's', 't', 'a', 'l', 'a', 'r
```

```
##Write a Python program to check that a string contains only a certain set of characters (in
##case a-z, A-Z and 0-9).
import re
txt = "YourString123"
x = re.compile("[A-Za-z0-9]+")
if x.fullmatch(string) is not None:
        print("Found match: " + string)
else:
    # if not found match
    print("No match")
```

```
Found match: YourString123
```

```
## Write a Python program that matches a string that has an a followed by zero or more b's
import re
def text_match(text):
        k = 'ab*?'
        if re.search(k,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("ac"))
print(text_match("abc"))
print(text_match("abbc"))
```

```
Found a match!
Found a match!
Found a match!
```

```
## Write a Python program that matches a string that has an a followed by one or more b's
import re
def text_match(text):
        k = 'ab+?'
        if re.search(k, text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("ab"))
print(text_match("abc"))
```

```
Found a match!
Found a match!
```

```
## Write a Python program that matches a string that has an a followed by zero or one 'b'

import re
def text_match(text):
        k = 'ab?'
        if re.search(k,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("ab"))
print(text_match("abc"))
print(text_match("abbc"))
print(text_match("aabbc"))
```

```
Found a match!
Found a match!
Found a match!
Found a match!
```

```
## Write a Python program that matches a string that has an a followed by three 'b'

import re
def text_match(text):
        k = 'ab{3}?'
        if re.search(k,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("abbb"))
print(text_match("aabbbbbc"))
```

```
Found a match!
Found a match!
```

```
## Write a Python program to find sequences of lowercase letters joined with an underscore
import re
def text_match(text):
        k = '^[a-z]+_[a-z]+$'
        if re.search(k,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("aab_cbbbc"))
print(text_match("aab_Abbbc"))
print(text_match("Aaab_abbbc"))
```

```
Found a match!
Not matched!
Not matched!
```

```
## Write a Python program to download and display the content of robot.txt for en.wikipedia.o
import requests
response = requests.get("https://en.wikipedia.org/robots.txt")
test = response.text
print("robots.txt for http://www.wikipedia.org/")
print("=====================================================")
print(test)
```

```
robots.txt for http://www.wikipedia.org/
=====================================================
# robots.txt for http://www.wikipedia.org/ and friends
#
# Please note: There are a lot of pages on this site, and there are
# some misbehaved spiders out there that go _way_ too fast. If you're
# irresponsible, your access to the site may be blocked.
#

# Observed spamming large amounts of https://en.wikipedia.org/?curid=NNNNNN
# and ignoring 429 ratelimit responses, claims to respect robots:
# http://mj12bot.com/
User-agent: MJ12bot
Disallow: /

# advertising-related bots:
User-agent: Mediapartners-Google*
Disallow: /

# Wikipedia work bots:
User-agent: IsraBot
Disallow:

User-agent: Orthogaffe
Disallow:

# Crawlers that are kind enough to obey, but which we'd rather not have
# unless they're feeding search engines.
User-agent: UbiCrawler
Disallow: /

User-agent: DOC
Disallow: /

User-agent: Zao
Disallow: /

# Some bots are known to be trouble, particularly those designed to copy
# entire sites. Please obey robots.txt.
User-agent: sitecheck.internetseer.com
Disallow: /

User-agent: Zealbot
Disallow: /

User-agent: MSIECrawler
Disallow: /

User-agent: SiteSnagger
Disallow: /

User-agent: WebStripper
Disallow: /

User-agent: WebCopier
Disallow: /
```

```
## Write a Python program to get the number of datasets currently listed on data.gov
from lxml import html
from lxml import cssselect
import requests
response = requests. get('http://www.data.gov/')
doc_gov = html. fromstring(response. text)
link_gov = doc_gov. cssselect('small a')[0]
print("Number of datasets currently listed on data.gov:"
print(link_gov.text)
```

```
  File "<ipython-input-39-13cc00419f82>", line 8
    print(link_gov)
        ^
SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

0s completed at 10:16 PM