```
In [ ]:  .          - Any Character Except New Line
         \d         - Digit (0-9)
         \D         - Not a Digit (0-9)
         \w         - Word Character (a-z, A-Z, 0-9, _)
         \W         - Not a Word Character
         \s         - Whitespace (space, tab, newline)
         \S         - Not Whitespace (space, tab, newline)

         \b         - Word Boundary
         \B         - Not a Word Boundary
         ^          - Beginning of a String
         $          - End of a String

         []         - Matches Characters in brackets
         [^ ]       - Matches Characters NOT in brackets
         |          - Either Or
         ( )        - Group

         Quantifiers:
         *          - 0 or More
         +          - 1 or More
         ?          - 0 or One
         {3}        - Exact Number
         {3,4}      - Range of Numbers (Minimum, Maximum)


         #### Sample Regexs ####

         [a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+
```

In [24]:
```python
import re

text_to_search = '''
abcdefghijklmnopqurtuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
1234567890

Ha HaHa

MetaCharacters (Need to be escaped):
. ^ $ * + ? { } [ ] \ | ( )

coreyms.com

321-555-4321
123.555.1234
350*555*1234
800.555.1234
900.555.1234

Mr. Schafer
Mr Smith
Ms Davis
Mrs. Robinson
Mr. T

cat
bat
pat
mat

'''
sentence = 'Start a sentence and then bring it to an end'
```

In [3]:
```python
print('\tTab')
```

```
        Tab
```

In [4]:
```python
print(r'\tTab') #raw strings
```

```
\tTab
```

In [5]:
```python
pattern = re.compile(r'abc')

matches = pattern.finditer(text_to_search)

for match in matches:
    print(match)
```

```
<re.Match object; span=(1, 4), match='abc'>
```

In [6]:
```python
print(text_to_search[1:4])
```

```
abc
```

```python
In [7]: pattern = re.compile(r'coreyms\.com') #\. if there is . in sentence

        matches = pattern.finditer(text_to_search)

        for match in matches:
            print(match)
```

```
<re.Match object; span=(142, 153), match='coreyms.com'>
```

```python
In [9]: pattern = re.compile(r'\BHa') #ha without space

        matches = pattern.finditer(text_to_search)

        for match in matches:
            print(match)
```

```
<re.Match object; span=(72, 74), match='Ha'>
```

```python
In [10]: pattern = re.compile(r'^Start') #ha without space

         matches = pattern.finditer(sentence)

         for match in matches:
             print(match)
```

```
<re.Match object; span=(0, 5), match='Start'>
```

```python
In [11]: pattern = re.compile(r'end$') #ha without space

         matches = pattern.finditer(sentence)

         for match in matches:
             print(match)
```

```
<re.Match object; span=(41, 44), match='end'>
```

```python
In [12]: pattern = re.compile(r'\d\d\d.\d\d\d.\d\d\d') #search phone number

         matches = pattern.finditer(text_to_search)

         for match in matches:
             print(match)
```

```
<re.Match object; span=(155, 166), match='321-555-432'>
<re.Match object; span=(168, 179), match='123.555.123'>
```

New

```python
In [15]: with open('data.txt','r',encoding = 'utf-8') as f:
             contents = f.read()
```

In [16]:
```python
print(contents)
```

Dave Martin
615-555-7164
173 Main St., Springfield RI 55924
davemartin@bogusemail.com

Charles Harris
800-555-5669
969 High St., Atlantis VA 34075
charlesharris@bogusemail.com

Eric Williams
560-555-5153
806 1st St., Faketown AK 86847
laurawilliams@bogusemail.com

Corey Jefferson
900-555-9340
826 Elm St., Epicburg NE 10671
coreyjefferson@bogusemail.com

Jennifer Martin-White
714-555-7405
212 Cedar St., Sunnydale CT 74983
jenniferwhite@bogusemail.com

Erick Davis
800-555-6771
519 Washington St., Olympus TN 32425
tomdavis@bogusemail.com

Neil Patterson
783-555-4799
625 Oak St., Dawnstar IL 61914
neilpatterson@bogusemail.com

Laura Jefferson
516-555-4615
890 Main St., Pythonville LA 29947
laurajefferson@bogusemail.com

Maria Johnson
127-555-1867
884 High St., Braavos ME 43597
mariajohnson@bogusemail.com

Michael Arnold
608-555-4938
249 Elm St., Quahog OR 90938
michaelarnold@bogusemail.com

Michael Smith
568-555-6051
619 Park St., Winterfell VA 99000
michaelsmith@bogusemail.com

Erik Stuart
292-555-1875

220 Cedar St., Lakeview NY 87282
robertstuart@bogusemail.com

Laura Martin
900-555-3205
391 High St., Smalltown WY 28362
lauramartin@bogusemail.com

Barbara Martin
614-555-1166
121 Hill St., Braavos UT 92474
barbaramartin@bogusemail.com

Linda Jackson
530-555-2676
433 Elm St., Westworld TX 61967
lindajackson@bogusemail.com

Eric Miller
470-555-2750
838 Main St., Balmora MT 56526
stevemiller@bogusemail.com

Dave Arnold
800-555-6089
732 High St., Valyria KY 97152
davearnold@bogusemail.com

Jennifer Jacobs
880-555-8319
217 High St., Old-town IA 82767
jenniferjacobs@bogusemail.com

Neil Wilson
777-555-8378
191 Main St., Mordor IL 72160
neilwilson@bogusemail.com

Kurt Jackson
998-555-7385
607 Washington St., Blackwater NH 97183
kurtjackson@bogusemail.com

Mary Jacobs
800-555-7100
478 Oak St., Bedrock IA 58176
maryjacobs@bogusemail.com

Michael White
903-555-8277
906 Elm St., Mordor TX 89212
michaelwhite@bogusemail.com

Jennifer Jenkins
196-555-5674
949 Main St., Smalltown SC 96962
jenniferjenkins@bogusemail.com

Sam Wright
900-555-5118
835 Pearl St., Smalltown ND 77737
samwright@bogusemail.com

John Davis
905-555-1630
451 Lake St., Bedrock GA 34615
johndavis@bogusemail.com

Eric Davis
203-555-3475
419 Lake St., Balmora OR 30826
neildavis@bogusemail.com

Laura Jackson
884-555-8444
443 Maple St., Quahog MS 29348
laurajackson@bogusemail.com

John Williams
904-555-8559
756 Hill St., Valyria KY 94854
johnwilliams@bogusemail.com

Michael Martin
889-555-7393
216 High St., Olympus NV 21888
michaelmartin@bogusemail.com

Maggie Brown
195-555-2405
806 Lake St., Lakeview MD 59348
maggiebrown@bogusemail.com

Erik Wilson
321-555-9053
354 Hill St., Mordor FL 74122
kurtwilson@bogusemail.com

Elizabeth Arnold
133-555-1711
805 Maple St., Winterfell NV 99431
elizabetharnold@bogusemail.com

Jane Martin
900-555-5428
418 Park St., Metropolis ID 16576
janemartin@bogusemail.com

Travis Johnson
760-555-7147
749 Washington St., Braavos SD 25668
travisjohnson@bogusemail.com

Laura Jefferson

391-555-6621
122 High St., Metropolis ME 29540
laurajefferson@bogusemail.com

Tom Williams
932-555-7724
610 High St., Old-town FL 60758
tomwilliams@bogusemail.com

Jennifer Taylor
609-555-7908
332 Main St., Pythonville OH 78172
jennifertaylor@bogusemail.com

Erick Wright
800-555-8810
858 Hill St., Blackwater NC 79714
jenniferwright@bogusemail.com

Steve Doe
149-555-7657
441 Elm St., Atlantis MS 87195
stevedoe@bogusemail.com

Kurt Davis
130-555-9709
404 Oak St., Atlantis ND 85386
kurtdavis@bogusemail.com

Corey Harris
143-555-9295
286 Pearl St., Vice City TX 57112
coreyharris@bogusemail.com

Nicole Taylor
903-555-9878
465 Hill St., Old-town LA 64102
nicoletaylor@bogusemail.com

Elizabeth Davis
574-555-3194
151 Lake St., Eerie SD 17880
elizabethdavis@bogusemail.com

Maggie Jenkins
496-555-7533
504 Lake St., Gotham PA 46692
maggiejenkins@bogusemail.com

Linda Davis
210-555-3757
201 Pine St., Vice City AR 78455
lindadavis@bogusemail.com

Dave Moore
900-555-9598
251 Pine St., Old-town OK 29087

davemoore@bogusemail.com

Linda Jenkins
866-555-9844
117 High St., Bedrock NE 11899
lindajenkins@bogusemail.com

Eric White
669-555-7159
650 Oak St., Smalltown TN 43281
samwhite@bogusemail.com

Laura Robinson
152-555-7417
377 Pine St., Valyria NC 78036
laurarobinson@bogusemail.com

Charles Patterson
893-555-9832
416 Pearl St., Valyria AK 62260
charlespatterson@bogusemail.com

Joe Jackson
217-555-7123
683 Cedar St., South Park KS 66724
joejackson@bogusemail.com

Michael Johnson
786-555-6544
288 Hill St., Smalltown AZ 18586
michaeljohnson@bogusemail.com

Corey Miller
780-555-2574
286 High St., Springfield IA 16272
coreymiller@bogusemail.com

James Moore
926-555-8735
278 Main St., Gotham KY 89569
jamesmoore@bogusemail.com

Jennifer Stuart
895-555-3539
766 Hill St., King's Landing GA 54999
jenniferstuart@bogusemail.com

Charles Martin
874-555-3949
775 High St., Faketown PA 89260
charlesmartin@bogusemail.com

Eric Wilks
800-555-2420
885 Main St., Blackwater OH 61275
joewilks@bogusemail.com

Elizabeth Arnold
936-555-6340
528 Hill St., Atlantis NH 88289
elizabetharnold@bogusemail.com

John Miller
372-555-9809
117 Cedar St., Thundera NM 75205
johnmiller@bogusemail.com

Corey Jackson
890-555-5618
115 Oak St., Gotham UT 36433
coreyjackson@bogusemail.com

Sam Thomas
670-555-3005
743 Lake St., Springfield MS 25473
samthomas@bogusemail.com

Patricia Thomas
509-555-5997
381 Hill St., Blackwater CT 30958
patriciathomas@bogusemail.com

Jennifer Davis
721-555-5632
125 Main St., Smalltown MT 62155
jenniferdavis@bogusemail.com

Patricia Brown
900-555-3567
292 Hill St., Gotham WV 57680
patriciabrown@bogusemail.com

Barbara Williams
147-555-6830
514 Park St., Balmora NV 55462
barbarawilliams@bogusemail.com

James Taylor
582-555-3426
776 Hill St., Dawnstar MA 51312
jamestaylor@bogusemail.com

Eric Harris
400-555-1706
421 Elm St., Smalltown NV 72025
barbaraharris@bogusemail.com

Travis Anderson
525-555-1793
937 Cedar St., Thundera WA 78862
travisanderson@bogusemail.com

Sam Robinson
317-555-6700

417 Pine St., Lakeview MD 13147
samrobinson@bogusemail.com

Steve Robinson
974-555-8301
478 Park St., Springfield NM 92369
steverobinson@bogusemail.com

Mary Wilson
800-555-3216
708 Maple St., Braavos UT 29551
marywilson@bogusemail.com

Sam Wilson
746-555-4094
557 Pearl St., Westworld KS 23225
samwilson@bogusemail.com

Charles Jones
922-555-1773
855 Hill St., Olympus HI 81427
charlesjones@bogusemail.com

Laura Brown
711-555-4427
980 Maple St., Smalltown MO 96421
laurabrown@bogusemail.com

Tom Harris
355-555-1872
676 Hill St., Blackwater AR 96698
tomharris@bogusemail.com

Patricia Taylor
852-555-6521
588 Pine St., Olympus FL 98412
patriciataylor@bogusemail.com

Barbara Williams
691-555-5773
351 Elm St., Sunnydale GA 26245
barbarawilliams@bogusemail.com

Maggie Johnson
332-555-5441
948 Cedar St., Quahog DE 56449
maggiejohnson@bogusemail.com

Kurt Miller
900-555-7755
381 Hill St., Quahog AL 97503
kurtmiller@bogusemail.com

Neil Stuart
379-555-3685
496 Cedar St., Sunnydale RI 49113
neilstuart@bogusemail.com

Linda Patterson
127-555-9682
736 Cedar St., Lakeview KY 47472
lindapatterson@bogusemail.com

Charles Davis
789-555-7032
678 Lake St., Mordor MN 11845
charlesdavis@bogusemail.com

Jennifer Jefferson
783-555-5135
289 Park St., Sunnydale WA 74526
jenniferjefferson@bogusemail.com

Erick Taylor
315-555-6507
245 Washington St., Bedrock IL 26941
coreytaylor@bogusemail.com

Robert Wilks
481-555-5835
573 Elm St., Sunnydale IL 47182
robertwilks@bogusemail.com

Travis Jackson
365-555-8287
851 Lake St., Metropolis PA 22772
travisjackson@bogusemail.com

Travis Jackson
911-555-7535
489 Oak St., Atlantis HI 73725
travisjackson@bogusemail.com

Laura Wilks
681-555-2460
371 Pearl St., Smalltown SC 47466
laurawilks@bogusemail.com

Neil Arnold
274-555-9800
504 Oak St., Faketown PA 73860
neilarnold@bogusemail.com

Linda Johnson
800-555-1372
667 High St., Balmora IN 82473
lindajohnson@bogusemail.com

Jennifer Wilson
300-555-7821
266 Pine St., Westworld DC 58720
jenniferwilson@bogusemail.com

Nicole White

133-555-3889
276 High St., Braavos IL 57764
nicolewhite@bogusemail.com

Maria Arnold
705-555-6863
491 Elm St., Metropolis PA 31836
mariaarnold@bogusemail.com

Jennifer Davis
215-555-9449
859 Cedar St., Old-town MT 31169
jenniferdavis@bogusemail.com

Mary Patterson
988-555-6112
956 Park St., Valyria CT 81541
marypatterson@bogusemail.com

Jane Stuart
623-555-3006
983 Oak St., Old-town RI 15445
janestuart@bogusemail.com

Robert Davis
192-555-4977
789 Maple St., Mordor IN 22215
robertdavis@bogusemail.com

James Taylor
178-555-4899
439 Hill St., Olympus NV 39308
jamestaylor@bogusemail.com

Eric Stuart
952-555-3089
777 High St., King's Landing AZ 16547
johnstuart@bogusemail.com

Charles Miller
900-555-6426
207 Washington St., Blackwater MA 24886
charlesmiller@bogusemail.com

In [17]:
```python
matches = pattern.finditer(contents)

for match in matches:
    print(match)
```

```
<re.Match object; span=(12, 23), match='615-555-716'>
<re.Match object; span=(102, 113), match='800-555-566'>
<re.Match object; span=(191, 202), match='560-555-515'>
<re.Match object; span=(281, 292), match='900-555-934'>
<re.Match object; span=(378, 389), match='714-555-740'>
<re.Match object; span=(467, 478), match='800-555-677'>
<re.Match object; span=(557, 568), match='783-555-479'>
<re.Match object; span=(647, 658), match='516-555-461'>
<re.Match object; span=(740, 751), match='127-555-186'>
<re.Match object; span=(829, 840), match='608-555-493'>
<re.Match object; span=(915, 926), match='568-555-605'>
<re.Match object; span=(1003, 1014), match='292-555-187'>
<re.Match object; span=(1091, 1102), match='900-555-320'>
<re.Match object; span=(1180, 1191), match='614-555-116'>
<re.Match object; span=(1269, 1280), match='530-555-267'>
<re.Match object; span=(1355, 1366), match='470-555-275'>
<re.Match object; span=(1439, 1450), match='800-555-608'>
<re.Match object; span=(1526, 1537), match='880-555-831'>
<re.Match object; span=(1614, 1625), match='777-555-837'>
<re.Match object; span=(1697, 1708), match='998-555-738'>
<re.Match object; span=(1790, 1801), match='800-555-710'>
<re.Match object; span=(1874, 1885), match='903-555-827'>
<re.Match object; span=(1962, 1973), match='196-555-567'>
<re.Match object; span=(2051, 2062), match='900-555-511'>
<re.Match object; span=(2135, 2146), match='905-555-163'>
<re.Match object; span=(2216, 2227), match='203-555-347'>
<re.Match object; span=(2300, 2311), match='884-555-844'>
<re.Match object; span=(2387, 2398), match='904-555-855'>
<re.Match object; span=(2475, 2486), match='889-555-739'>
<re.Match object; span=(2562, 2573), match='195-555-240'>
<re.Match object; span=(2647, 2658), match='321-555-905'>
<re.Match object; span=(2734, 2745), match='133-555-171'>
<re.Match object; span=(2826, 2837), match='900-555-542'>
<re.Match object; span=(2915, 2926), match='760-555-714'>
<re.Match object; span=(3012, 3023), match='391-555-662'>
<re.Match object; span=(3103, 3114), match='932-555-772'>
<re.Match object; span=(3192, 3203), match='609-555-790'>
<re.Match object; span=(3284, 3295), match='800-555-881'>
<re.Match object; span=(3372, 3383), match='149-555-765'>
<re.Match object; span=(3452, 3463), match='130-555-970'>
<re.Match object; span=(3535, 3546), match='143-555-929'>
<re.Match object; span=(3624, 3635), match='903-555-987'>
<re.Match object; span=(3714, 3725), match='574-555-319'>
<re.Match object; span=(3802, 3813), match='496-555-753'>
<re.Match object; span=(3887, 3898), match='210-555-375'>
<re.Match object; span=(3971, 3982), match='900-555-959'>
<re.Match object; span=(4056, 4067), match='866-555-984'>
<re.Match object; span=(4140, 4151), match='669-555-715'>
<re.Match object; span=(4225, 4236), match='152-555-741'>
<re.Match object; span=(4317, 4328), match='893-555-983'>
<re.Match object; span=(4407, 4418), match='217-555-712'>
<re.Match object; span=(4498, 4509), match='786-555-654'>
<re.Match object; span=(4588, 4599), match='780-555-257'>
<re.Match object; span=(4676, 4687), match='926-555-873'>
<re.Match object; span=(4762, 4773), match='895-555-353'>
<re.Match object; span=(4859, 4870), match='874-555-394'>
<re.Match object; span=(4945, 4956), match='800-555-242'>
```

```
<re.Match object; span=(5034, 5045), match='936-555-634'>
<re.Match object; span=(5123, 5134), match='372-555-980'>
<re.Match object; span=(5210, 5221), match='890-555-561'>
<re.Match object; span=(5292, 5303), match='670-555-300'>
<re.Match object; span=(5382, 5393), match='509-555-599'>
<re.Match object; span=(5475, 5486), match='721-555-563'>
<re.Match object; span=(5566, 5577), match='900-555-356'>
<re.Match object; span=(5656, 5667), match='147-555-683'>
<re.Match object; span=(5745, 5756), match='582-555-342'>
<re.Match object; span=(5830, 5841), match='400-555-170'>
<re.Match object; span=(5921, 5932), match='525-555-179'>
<re.Match object; span=(6011, 6022), match='317-555-670'>
<re.Match object; span=(6099, 6110), match='974-555-830'>
<re.Match object; span=(6189, 6200), match='800-555-321'>
<re.Match object; span=(6273, 6284), match='746-555-409'>
<re.Match object; span=(6360, 6371), match='922-555-177'>
<re.Match object; span=(6445, 6456), match='711-555-442'>
<re.Match object; span=(6530, 6541), match='355-555-187'>
<re.Match object; span=(6619, 6630), match='852-555-652'>
<re.Match object; span=(6711, 6722), match='691-555-577'>
<re.Match object; span=(6803, 6814), match='332-555-544'>
<re.Match object; span=(6889, 6900), match='900-555-775'>
<re.Match object; span=(6971, 6982), match='379-555-368'>
<re.Match object; span=(7061, 7072), match='127-555-968'>
<re.Match object; span=(7152, 7163), match='789-555-703'>
<re.Match object; span=(7243, 7254), match='783-555-513'>
<re.Match object; span=(7336, 7347), match='315-555-650'>
<re.Match object; span=(7427, 7438), match='481-555-583'>
<re.Match object; span=(7515, 7526), match='365-555-828'>
<re.Match object; span=(7607, 7618), match='911-555-753'>
<re.Match object; span=(7693, 7704), match='681-555-246'>
<re.Match object; span=(7779, 7790), match='274-555-980'>
<re.Match object; span=(7864, 7875), match='800-555-137'>
<re.Match object; span=(7953, 7964), match='300-555-782'>
<re.Match object; span=(8043, 8054), match='133-555-388'>
<re.Match object; span=(8129, 8140), match='705-555-686'>
<re.Match object; span=(8218, 8229), match='215-555-944'>
<re.Match object; span=(8309, 8320), match='988-555-611'>
<re.Match object; span=(8395, 8406), match='623-555-300'>
<re.Match object; span=(8479, 8490), match='192-555-497'>
<re.Match object; span=(8564, 8575), match='178-555-489'>
<re.Match object; span=(8648, 8659), match='952-555-308'>
<re.Match object; span=(8741, 8752), match='900-555-642'>
```

```python
In [21]: pattern = re.compile(r'\d\d\d[-.]\d\d\d[-.]\d\d\d') #character search

         matches = pattern.finditer(text_to_search)

         for match in matches:
             print(match)
```

```
<re.Match object; span=(155, 166), match='321-555-432'>
<re.Match object; span=(168, 179), match='123.555.123'>
```

In [23]:
```python
pattern = re.compile(r'[89]00[-.]\d\d\d[-.]\d\d\d') #character search

matches = pattern.finditer(text_to_search)

for match in matches:
    print(match)
```

```
<re.Match object; span=(194, 205), match='800.555.123'>
<re.Match object; span=(207, 218), match='900.555.123'>
```

In [ ]:
```python
#[^a-zA-Z] ^ negate and - for range
```

In [25]:
```python
pattern = re.compile(r'[^b]at') #character search

matches = pattern.finditer(text_to_search)

for match in matches:
    print(match)
```

```
<re.Match object; span=(272, 275), match='cat'>
<re.Match object; span=(280, 283), match='pat'>
<re.Match object; span=(284, 287), match='mat'>
```

In [27]:
```python
#Quantifiers

pattern = re.compile(r'\d{3}.\d{3}.\d{4}') #character search

matches = pattern.finditer(text_to_search)

for match in matches:
    print(match)
```

```
<re.Match object; span=(155, 167), match='321-555-4321'>
<re.Match object; span=(168, 180), match='123.555.1234'>
<re.Match object; span=(181, 193), match='350*555*1234'>
<re.Match object; span=(194, 206), match='800.555.1234'>
<re.Match object; span=(207, 219), match='900.555.1234'>
```

In [30]:
```python
#Quantifiers

pattern = re.compile(r'M(r|s|rs)\.?\s[A-Z]\w*') #character search (| or) (?0 or more after alpha) \s space \w* 0 or more words after word

matches = pattern.finditer(text_to_search)

for match in matches:
    print(match)
```

```
<re.Match object; span=(221, 232), match='Mr. Schafer'>
<re.Match object; span=(233, 241), match='Mr Smith'>
<re.Match object; span=(242, 250), match='Ms Davis'>
<re.Match object; span=(251, 264), match='Mrs. Robinson'>
<re.Match object; span=(265, 270), match='Mr. T'>
```

In [31]:
```python
emails = '''
CoreyMSchafer@gmail.com
corey.schafer@university.edu
corey-321-schafer@my-work.net
'''
```

In [36]:
```python
pattern = re.compile(r'[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+')   #
(r'[a-zA-Z0-9.-]+@[a-zA-Z-]+\.(com|edu|net)')

matches = pattern.finditer(emails)

for match in matches:
    print(match)
```

```
<re.Match object; span=(1, 24), match='CoreyMSchafer@gmail.com'>
<re.Match object; span=(25, 53), match='corey.schafer@university.edu'>
<re.Match object; span=(54, 83), match='corey-321-schafer@my-work.net'>
```

In [37]:
```python
urls = '''
https://www.google.com
http://coreyms.com
https://youtube.com
https://www.nasa.gov
'''
```

In [41]:
```python
pattern = re.compile(r'https?://(www\.)?(\w+)(\.\w+)')   #?0 or s()group ? \w w
ord

matches = pattern.finditer(urls)

for match in matches:
    print(match.group(2)) #() is group
```

```
google
coreyms
youtube
nasa
```

In [43]:
```python
pattern = re.compile(r'https?://(www\.)?(\w+)(\.\w+)')   #?0 or s()group ? \w w
ord

subbed_urls = pattern.sub(r'\2\3',urls) #substitution
print(subbed_urls)
```

```
google.com
coreyms.com
youtube.com
nasa.gov
```

In [44]:
```python
#findall

pattern = re.compile(r'\d{3}.\d{3}.\d{4}') #character search

matches = pattern.findall(text_to_search)

for match in matches:
    print(match)
```

```
321-555-4321
123.555.1234
350*555*1234
800.555.1234
900.555.1234
```

In [47]:
```python
#match

sentence = 'Start a sentence and then bring it to an end'

pattern = re.compile(r'Start')

matches = pattern.match(sentence) #match finds only first character

print(matches)
```

```
<re.Match object; span=(0, 5), match='Start'>
```

In [46]:
```python
#search

sentence = 'Start a sentence and then bring it to an end'

pattern = re.compile(r'bring')

matches = pattern.search(sentence)

print(matches)
```

```
<re.Match object; span=(26, 31), match='bring'>
```

In [50]:
```python
#Flags

#Ignore case flag

sentence = 'Start a sentence and then bring it to an end'

pattern = re.compile(r'start', re.I) #/IGNORECASE

matches = pattern.search(sentence)

print(matches)
```

```
<re.Match object; span=(0, 5), match='Start'>
```

In [ ]: