Αρχεία

In [1]:
```
!pwd
```

```
/Users/admin/Downloads
```

In [18]:
```
with open('Homo_sapiens.gene_info', 'r') as f:
    l = f.readline()
    print (l)
```

```
#tax_id GeneID  Symbol  LocusTag        Synonyms        dbXrefs chromosome
        map_location    description     type_of_gene    Symbol_from_nomencl
ature_authority Full_name_from_nomenclature_authority   Nomenclature_status
        Other_designations      Modification_date       Feature_type
```

In [5]:
```
with open('Homo_sapiens.gene_info') as f:
    l = f.readlines()
    #print (l)
```

In [6]:
```
len(l)
```

Out[6]:
```
61719
```

In [7]:
```
l[1000]
```

Out[7]:
```
'9606\t1218\tCMD1B\t-\tCMPD1|FDC\tMIM:600884\t9\t9q13-q22\tcardiomyopathy,
dilated 1B (autosomal dominant)\tunknown\t-\t-\t-\t-\t20191002\t-\n'
```

In [8]:
```
with open('Homo_sapiens.gene_info') as f:
    l = f.read()
```

In [9]:
```
len(l)
```

Out[9]:
```
13502145
```

In [10]:
```
!ls -l Homo_sapiens.gene_info
```

```
-rw-r--r--@ 1 admin  staff  13502145 Mar 19 11:54 Homo_sapiens.gene_info
```

In [11]:
```
with open('Homo_sapiens.gene_info') as f:
    for line in f:
        print (line)
        break
```

```
#tax_id GeneID  Symbol  LocusTag        Synonyms        dbXrefs chromosome
        map_location    description     type_of_gene    Symbol_from_nomencl
ature_authority Full_name_from_nomenclature_authority   Nomenclature_status
        Other_designations      Modification_date       Feature_type
```

In [12]:
```
with open('file.txt', 'w') as f:
    f.write('123\n')
    f.write('456\n')
```

In [13]:
```
!cat file.txt
```

```
123
456
```

In [14]:
```
with open('file.txt', 'x') as f:
    f.write('123\n')
    f.write('456\n')
```

```
---------------------------------------------------------------------------
FileExistsError                           Traceback (most recent call last)
<ipython-input-14-c92598bdd549> in <module>
----> 1 with open('file.txt', 'x') as f:
      2     f.write('123\n')
      3     f.write('456\n')

FileExistsError: [Errno 17] File exists: 'file.txt'
```

In [15]:
```
with open('file2.txt', 'x') as f:
    f.write('123\n')
    f.write('456\n')
```

In [16]:
```
with open('file2.txt', 'a') as f:
    f.write('789\n')
    f.write('666\n')
```

In [17]:
```
!cat file2.txt
```

```
123
456
789
666
```

In [19]:
```
def f(x):
    with open('file2.txt', 'a') as f:
        f.write(x + '\n')

f('Mitsos')
```

In [20]:
```
!cat file2.txt
```

```
123
456
789
666
Mitsos
```

In [30]:
```
%%writefile file.txt


this is a test
```

```
Overwriting file.txt
```

In [22]:
```
!cat file.txt
```

```
this is a test
```

In [23]:
```
with open('file.txt') as f:
    d = f.read()
print (d)
```

```
this is a test
```

In [26]:
```
%%timeit

l = [1,2,3,4,5,6,7,8,9,10]
def f(x):
    return x%2==1

list(filter(f, l))
```

```
1.66 µs ± 94.5 ns per loop (mean ± std. dev. of 7 runs, 100000 loops each)
```

In [27]:
```
%%timeit

l = [1,2,3,4,5,6,7,8,9,10]

[x for x in l if x%2==1]
```

```
887 ns ± 55.6 ns per loop (mean ± std. dev. of 7 runs, 1000000 loops each)
```

In [31]:
```
f1 = open('Homo_sapiens.gene_info')
f2 = open('Homo_sapiens.gene_info')
```

In [34]:
```
f1.readline()
```

Out[34]:
```
'9606\t2\tA2M\t-\tA2MD|CPAMD5|FWP007|S863-7\tMIM:103950|HGNC:HGNC:7|Ensemb
l:ENSG00000175899\t12\t12p13.31\talpha-2-macroglobulin\tprotein-coding\tA2
M\talpha-2-macroglobulin\tO\talpha-2-macroglobulin|C3 and PZP-like alpha-2-
macroglobulin domain-containing protein 5|alpha-2-M\t20210316\t-\n'
```

In [35]:
```
f2.readline()
```

Out[35]:
```
'#tax_id\tGeneID\tSymbol\tLocusTag\tSynonyms\tdbXrefs\tchromosome\tmap_loca
tion\tdescription\ttype_of_gene\tSymbol_from_nomenclature_authority\tFull_n
ame_from_nomenclature_authority\tNomenclature_status\tOther_designations\tM
odification_date\tFeature_type\n'
```

In [36]:
```
f1.close()
```

In [37]:
```
f2.readline()
```

Out[37]:
```
'9606\t1\tA1BG\t-\tA1B|ABG|GAB|HYST2477\tMIM:138670|HGNC:HGNC:5|Ensembl:ENS
G00000121410\t19\t19q13.43\talpha-1-B glycoprotein\tprotein-coding\tA1BG\ta
lpha-1-B glycoprotein\tO\talpha-1B-glycoprotein|HEL-S-163pA|epididymis secr
etory sperm binding protein Li 163pA\t20210302\t-\n'
```

In [38]:
```
f2.close()
```

In [39]:
```
f1 = open('file.txt', 'w')
```

In [40]:
```
f2 = open('file.txt', 'w')
```

In [41]:
```
!cat file.txt
```

In [42]:
```
f1.write('111\n')
```

Out[42]:
```
4
```

In [43]:
```
!cat file.txt
```

In [45]:
```
f1.flush()
```

In [46]:
```
!cat file.txt
```

```
111
```

In [47]:
```
f2.write('222\n')
```

Out[47]:
```
4
```

In [48]:
```
!cat file.txt
```

```
111
```

In [49]:
```
f2.flush()
```

In [50]:
```
!cat file.txt
```

```
222
```

In [52]:
```
f1.close()
```

In [53]:
```
f2.close()
```

Serialization

In [51]:
```
a = [1,2,3, {"a": 4, "b": [4,5,6,7]}, [5,6,7,]]
```

In [57]:
```
with open('file.txt', 'w') as f:
    f.write(str(a))
```

In [59]:
```
!cat file.txt
```

```
[1, 2, 3, {'a': 4, 'b': [4, 5, 6, 7]}, [5, 6, 7]]
```

In [60]:
```
b = [1, 2, 3, {'a': 4, 'b': [4, 5, 6, 7]}, [5, 6, 7]]
```

In [61]:
```
import json
```

In [65]:
```
b = json.dumps(a)
```

In [64]:
```
type(b)
```

Out[64]:
```
str
```

In [66]:
```
c = json.loads(b)
```

In [67]:
```
c == a
```

Out[67]:
```
True
```

In [68]:
```
with open('data.json', 'w') as f:
    json.dump(a, f)
```

In [69]:
```
!cat data.json
```

```
[1, 2, 3, {"a": 4, "b": [4, 5, 6, 7]}, [5, 6, 7]]
```

In [70]:
```
with open('data.json', 'r') as f:
    c = json.load(f)
```

In [71]:
```
c == a
```

Out[71]:
```
True
```

In [72]:
```
json.dumps({1,2,3})
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-72-f6a0a593b80d> in <module>
----> 1 json.dumps({1,2,3})

~/anaconda3/lib/python3.8/json/__init__.py in dumps(obj, skipkeys, ensure_a
```

```
scii, check_circular, allow_nan, cls, indent, separators, default, sort_key
s, **kw)
    229         cls is None and indent is None and separators is None and
    230         default is None and not sort_keys and not kw):
--> 231         return _default_encoder.encode(obj)
    232     if cls is None:
    233         cls = JSONEncoder

~/anaconda3/lib/python3.8/json/encoder.py in encode(self, o)
    197         # exceptions aren't as detailed.  The list call should be r
oughly
    198         # equivalent to the PySequence_Fast that ''.join() would d
o.
--> 199         chunks = self.iterencode(o, _one_shot=True)
    200         if not isinstance(chunks, (list, tuple)):
    201             chunks = list(chunks)

~/anaconda3/lib/python3.8/json/encoder.py in iterencode(self, o, _one_shot)
    255                 self.key_separator, self.item_separator, self.sort_
keys,
    256                 self.skipkeys, _one_shot)
--> 257         return _iterencode(o, 0)
    258
    259 def _make_iterencode(markers, _default, _encoder, _indent, _floatst
r,

~/anaconda3/lib/python3.8/json/encoder.py in default(self, o)
    177
    178         """
--> 179         raise TypeError(f'Object of type {o.__class__.__name__} '
    180                         f'is not JSON serializable')
    181

TypeError: Object of type set is not JSON serializable
```

In [73]:
```
json.dumps(lambda x : x+1)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-73-e092f9923bfe> in <module>
----> 1 json.dumps(lambda x : x+1)

~/anaconda3/lib/python3.8/json/__init__.py in dumps(obj, skipkeys, ensure_a
scii, check_circular, allow_nan, cls, indent, separators, default, sort_key
s, **kw)
    229         cls is None and indent is None and separators is None and
    230         default is None and not sort_keys and not kw):
--> 231         return _default_encoder.encode(obj)
    232     if cls is None:
    233         cls = JSONEncoder

~/anaconda3/lib/python3.8/json/encoder.py in encode(self, o)
    197         # exceptions aren't as detailed.  The list call should be r
oughly
    198         # equivalent to the PySequence_Fast that ''.join() would d
o.
--> 199         chunks = self.iterencode(o, _one_shot=True)
    200         if not isinstance(chunks, (list, tuple)):
    201             chunks = list(chunks)

~/anaconda3/lib/python3.8/json/encoder.py in iterencode(self, o, _one_shot)
    255                 self.key_separator, self.item_separator, self.sort_
keys,
    256                 self.skipkeys, _one_shot)
--> 257         return _iterencode(o, 0)
    258
    259 def _make_iterencode(markers, _default, _encoder, _indent, _floatst
r,

~/anaconda3/lib/python3.8/json/encoder.py in default(self, o)
    177
    178         """
--> 179         raise TypeError(f'Object of type {o.__class__.__name__} '
    180                         f'is not JSON serializable')
    181

TypeError: Object of type function is not JSON serializable
```

In [74]:
```
def f(x):
    return x+1
json.dump(x)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-74-d589213418b1> in <module>
      1 def f(x):
      2     return x+1
----> 3 json.dump(x)

NameError: name 'x' is not defined
```

In [75]:
```
import pickle
```

In [77]:
```
a
```

Out[77]:
```
[1, 2, 3, {'a': 4, 'b': [4, 5, 6, 7]}, [5, 6, 7]]
```

In [78]:
```
a.append({3,4,5,6})
```

In [79]:
```
a.append(f)
```

In [86]:
```
a = a[:-2]
```

In [88]:
```
a.append({2,3,4,5})
```

In [89]:
```
a.append(lambda x : x+1)
```

In [81]:
```
json.dumps(a)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-81-2f50cf32d976> in <module>
----> 1 json.dumps(a)

~/anaconda3/lib/python3.8/json/__init__.py in dumps(obj, skipkeys, ensure_a
scii, check_circular, allow_nan, cls, indent, separators, default, sort_key
s, **kw)
    229         cls is None and indent is None and separators is None and
    230         default is None and not sort_keys and not kw):
--> 231         return _default_encoder.encode(obj)
    232     if cls is None:
    233         cls = JSONEncoder

~/anaconda3/lib/python3.8/json/encoder.py in encode(self, o)
    197         # exceptions aren't as detailed.  The list call should be r
oughly
    198         # equivalent to the PySequence_Fast that ''.join() would d
o.
--> 199         chunks = self.iterencode(o, _one_shot=True)
    200         if not isinstance(chunks, (list, tuple)):
    201             chunks = list(chunks)

~/anaconda3/lib/python3.8/json/encoder.py in iterencode(self, o, _one_shot)
    255                 self.key_separator, self.item_separator, self.sort_
keys,
    256                 self.skipkeys, _one_shot)
--> 257         return _iterencode(o, 0)
    258
    259 def _make_iterencode(markers, _default, _encoder, _indent, _floatst
r,

~/anaconda3/lib/python3.8/json/encoder.py in default(self, o)
    177
    178         """
--> 179         raise TypeError(f'Object of type {o.__class__.__name__} '
    180                         f'is not JSON serializable')
    181

TypeError: Object of type set is not JSON serializable
```

In [91]:
```
pickle.dumps(a)
```

```
---------------------------------------------------------------------------
PicklingError                             Traceback (most recent call last)
<ipython-input-91-e346a7b9f67a> in <module>
----> 1 pickle.dumps(a)

PicklingError: Can't pickle <function <lambda> at 0x7faf82583ca0>: attribut
e lookup <lambda> on __main__ failed
```

In [90]:
```
with open('file.pickle', 'w') as f2:
    pickle.dump(a, f2)
```

```
---------------------------------------------------------------------------
PicklingError                             Traceback (most recent call last)
<ipython-input-90-81309c48e28e> in <module>
      1 with open('file.pickle', 'w') as f2:
----> 2     pickle.dump(a, f2)

PicklingError: Can't pickle <function <lambda> at 0x7faf82583ca0>: attribut
e lookup <lambda> on __main__ failed
```

In [92]:
```
pickle.dumps(lambda x : x+1)
```

```
---------------------------------------------------------------------------
PicklingError                                Traceback (most recent call last)
<ipython-input-92-7d5e9e36d753> in <module>
----> 1 pickle.dumps(lambda x : x+1)

PicklingError: Can't pickle <function <lambda> at 0x7faf82583f70>: attribut
e lookup <lambda> on __main__ failed
```

In [93]:
```
def f(x):
    return 3

pickle.dumps(f)
```

Out[93]:
```
b'\x80\x04\x95\x12\x00\x00\x00\x00\x00\x00\x00\x8c\x08__main__\x94\x8c\x01
f\x94\x93\x94.'
```

In [95]:
```
with open('test.pickle', 'w') as f2:
    pickle.dump([1,2,3,f], f2)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-95-f21cfafd1eb0> in <module>
      1 with open('test.pickle', 'w') as f2:
----> 2     pickle.dump([1,2,3,f], f2)

TypeError: write() argument must be str, not bytes
```

itertools

In [96]:
```
from itertools import combinations
```

In [97]:
```
genes = ['aa', 'bb', 'cc', 'dd']
for k,l in combinations(genes, 2):
    print (k,l)
```

```
aa bb
aa cc
aa dd
bb cc
bb dd
cc dd
```

In [100…
```
sum(1 for k in combinations(list(range(10)), 2))
```

Out[100…
```
45
```

In [101…
```
sum(1 for k in combinations(list(range(49)), 6))
```

Out[101…
```
13983816
```

In [104…
```
i=1
for k in combinations(list(range(1,50)), 6):
    i += 1
    print (k)
    if i > 10:
        break
```

```
(1, 2, 3, 4, 5, 6)
(1, 2, 3, 4, 5, 7)
(1, 2, 3, 4, 5, 8)
(1, 2, 3, 4, 5, 9)
(1, 2, 3, 4, 5, 10)
(1, 2, 3, 4, 5, 11)
(1, 2, 3, 4, 5, 12)
(1, 2, 3, 4, 5, 13)
(1, 2, 3, 4, 5, 14)
(1, 2, 3, 4, 5, 15)
```

In [107…
```
a = [10,5,2,]
b = [7,6,5,7]

from itertools import product

for x,y in product(a,b):
    print (x,y)
```

```
10 7
10 6
10 5
10 7
5 7
5 6
5 5
5 7
2 7
2 6
2 5
2 7
```

In [111…
```
import random

a = [random.randint(-40,40) for x in range(100)]
```

In [112…
```
print (a)
```

```
[28, -4, 2, -21, 9, 11, -7, 25, 22, 39, -29, -22, 21, -32, -24, 3, 9, -1, -
4, 26, 11, -8, 20, -11, 8, 26, 33, -1, -28, 22, -21, 23, -5, 26, -19, -29,
27, 38, 5, -9, 16, -17, 16, -35, 6, 33, 37, 18, 21, -21, -40, -18, -17, -1
6, 29, -18, -2, -7, -6, 25, 12, -36, 39, -15, 6, 29, -13, 22, -25, -6, 39,
27, -3, -31, -32, -25, -19, -5, -6, 2, -15, 25, 2, -21, 38, -15, -15, -25,
-9, 4, -33, -1, 20, -15, 29, -21, -16, 7, -5, 37]
```

In [115…
```
b = []
for start, end in combinations(list(range(0, 101)), 2):
    b.append( (sum(a[start:end]), start, end)        )
```

In [118…
```
max(b)
```

Out[118…
```
(254, 0, 49)
```

In [119…
```
sum(a)
```

Out[119…
```
94
```

In [123…
```
from itertools import chain

def powerset(iterable):
    "powerset([1,2,3]) --> () (1,) (2,) (3,) (1,2) (1,3) (2,3) (1,2,3)"
    s = list(iterable)
    return chain.from_iterable(combinations(s, r) for r in range(len(s)+1))
```

In [124…
```
for x in powerset(list('mitsos')):
    print (''.join(x))
```

```
m
i
t
s
o
s
mi
mt
ms
mo
```

```
ms
it
is
io
is
ts
to
ts
so
ss
os
mit
mis
mio
mis
mts
mto
mts
mso
mss
mos
its
ito
its
iso
iss
ios
tso
tss
tos
sos
mits
mito
mits
miso
miss
mios
mtso
mtss
mtos
msos
itso
itss
itos
isos
tsos
mitso
mitss
mitos
misos
mtsos
itsos
```

In [126…
```
import inspect

g = lambda x: x/2

f = lambda x : g(x+1)


inspect.getsource(f)
```

Out[126…
```
'f = lambda x : g(x+1)\n'
```

In [128…
```
pickle.dumps(f)
```

```
---------------------------------------------------------------------------
PicklingError                             Traceback (most recent call last)
<ipython-input-128-6d61f1b895e9> in <module>
----> 1 pickle.dumps(f)

PicklingError: Can't pickle <function <lambda> at 0x7faf82198e50>: attribut
e lookup <lambda> on __main__ failed
```

In [129…
```
a=[1,2,3,4,5,6,7,8,9,0,]
```

In [130…
```
a = [
    "asdASD",
    "asdASDasdgfasef",
    "ASDFASDFASDA",
    "sdfgsdfgsdfg",
]
```

In [131…
```
def f(x):
    return x+1

def f(x,):
    return (x+1)
```

Regular Expression

In [132…
```
a = 'NG_007400.1:g.8638G>T'
```

In [133…
```
def validate(x):
    i = a.find(':')
    if i<0:
        return False

    if a.count(':') != 1:
        return False

    first, second = a.split(':')
```

In [134…
```
import re
```

In [135…
```
'...'
```

Out[135…
```
'...'
```

In [141…
```
a= re.fullmatch('\d.[ab]', '3hj')
print (a)
```

```
None
```

In [142…
```
a= re.fullmatch('\d.[ab]', '3hb')
print (a)
```

```
<re.Match object; span=(0, 3), match='3hb'>
```

In [148…
```
a= re.fullmatch('\d\s[ab]+', '3 bbababababababab')
print (a)
```

```
<re.Match object; span=(0, 16), match='3 bbababababababab'>
```

In [149…
```
a= re.fullmatch('\d\s[ab]+', '3 ')
print (a)
```

```
None
```

In [150…
```
a= re.fullmatch('\d\s[ab]*', '3 ')
print (a)
```

```
<re.Match object; span=(0, 2), match='3 '>
```

In [151…
```
a= re.fullmatch('\d\s[ab]*', '3 abababababbbbb')
print (a)
```
```
<re.Match object; span=(0, 16), match='3 abababababbbbb'>
```

In [152…
```
a= re.fullmatch('\d\s[ab]?', '3 ababb')
print (a)
```
```
None
```

In [153…
```
a= re.fullmatch('\d\s[ab]?', '3 a')
print (a)
```
```
<re.Match object; span=(0, 3), match='3 a'>
```

In [154…
```
a= re.fullmatch('\d\s[ab]?', '3 ')
print (a)
```
```
<re.Match object; span=(0, 2), match='3 '>
```

In [155…
```
a = 'NG_007400.1:g.8638G>T'
```

In [156…
```
'+6912233456789'
```

Out[156…
```
'+6912233456789'
```

In [157…
```
re.fullmatch('\+?\d+', '+324234234')
```

Out[157…
```
<re.Match object; span=(0, 10), match='+324234234'>
```

In [158…
```
re.fullmatch('\+?\d+', '324234234')
```

Out[158…
```
<re.Match object; span=(0, 9), match='324234234'>
```

In [159…
```
re.fullmatch('\+?\d+', 'a324234234')
```

In [160…
```
a = 'NG_007400.1:g.8638G>T'
```

In [161…
```
re.fullmatch('\w\w\w', 'sdd')
```

Out[161…
```
<re.Match object; span=(0, 3), match='sdd'>
```

In [162…
```
re.fullmatch('\w\w\w', 's1d')
```

Out[162…
```
<re.Match object; span=(0, 3), match='s1d'>
```

In [166…
```
re.fullmatch('\w\w\w', 's_d')
```

Out[166…
```
<re.Match object; span=(0, 3), match='s_d'>
```

In [167…
```
a = 'NG_007400.1:g.8638G>T'
```

In [168…
```
re.fullmatch('\w+\.\d+:[cgpn]\.\d+[ACGT]>[ACGT]', a)
```

Out[168…
```
<re.Match object; span=(0, 21), match='NG_007400.1:g.8638G>T'>
```

In [171…
```
a = re.fullmatch('\w+\.\d+:[cgpn]\.\d+[ACGT]>[ACGT]', 'NG_007400.1:g.8638G>
print (a)
```
```
<re.Match object; span=(0, 21), match='NG_007400.1:g.8638G>T'>
```

In [175…
```
a = re.fullmatch('\w+\.\d+:[cgpn]\.\d+[ACGT]>[ACGT]', 'NG_007400.1:g.8638G>
print (a)
```

```
<re.Match object; span=(0, 21), match='NG_007400.1:g.8638G>T'>
```

In [179…
```
a = re.fullmatch('(\w+)\.\d+:[cgpn]\.(\d+)[ACGT]>[ACGT]', 'NG_007400.1:g.86
print (a)
```

```
<re.Match object; span=(0, 21), match='NG_007400.1:g.8638G>T'>
```

In [180…
```
a.group(1)
```

Out[180…
```
'NG_007400'
```

In [181…
```
a.group(2)
```

Out[181…
```
'8638'
```

In [183…
```
a = re.search('\d+', 'Mit8sos')
print (a)
```

```
<re.Match object; span=(3, 4), match='8'>
```

In [184…
```
a = re.fullmatch('\d+', 'Mit8sos')
print (a)
```

```
None
```

In [185…
```
a = re.search('^\d+', 'Mit8sos')
print (a)
```

```
None
```

In [186…
```
a = re.search('^\d+', '88888Mitsos')
print (a)
```

```
<re.Match object; span=(0, 5), match='88888'>
```

In [189…
```
a = re.search('\d+$', 'Mitsos8767')
print (a)
```

```
<re.Match object; span=(6, 10), match='8767'>
```

In [190…
```
bool(a)
```

Out[190…
```
True
```

In [191…
```
a = re.search('\d+$', 'Mitsos8767HH')
print (a)
bool(a)
```

```
None
```

Out[191…
```
False
```

In [192…
```
a = 'a\nb'
print (a)
```

```
a
b
```

In [193…
```
a = r'a\nb'
print (a)
```

```
a\nb
```

In [196…
```
name = 'mitsos'
a = f'my name = {name}'
print (a)
```
```
my name = mitsos
```
In [197…
```
a = re.search(r'\d+$', 'Mitsos8767')
```
In [202…
```
re.match(r'\d', '8afsdfasdf')
```
Out[202…
```
<re.Match object; span=(0, 1), match='8'>
```
In [203…
```
re.search(r'^\d', '8afsdfasdf')
```
Out[203…
```
<re.Match object; span=(0, 1), match='8'>
```
In [204…
```
re.search(r'^\d...\w$', '5j5jo')
```
Out[204…
```
<re.Match object; span=(0, 5), match='5j5jo'>
```
In [206…
```
re.fullmatch(r'\d...\w', '5j5jo')
```
Out[206…
```
<re.Match object; span=(0, 5), match='5j5jo'>
```
In [212…
```
re.fullmatch(r'\d{4}', '1456')
```
Out[212…
```
<re.Match object; span=(0, 4), match='1456'>
```
In [213…
```
re.fullmatch(r'\w{2,3} \d{4}', 'QQQ 6789')
```
Out[213…
```
<re.Match object; span=(0, 8), match='QQQ 6789'>
```
In [216…
```
re.fullmatch(r'[ABEZHIKMNOPTXY]{2,3} \d{4}', 'ABE 6789')
```
Out[216…
```
<re.Match object; span=(0, 8), match='ABE 6789'>
```
In [217…
```
re.fullmatch(r'[ABEZHIKMNOPTXY]{2,3} [0123456789]{4}', 'ABE 6789')
```
Out[217…
```
<re.Match object; span=(0, 8), match='ABE 6789'>
```
In [218…
```
re.fullmatch(r'[ABEZHIKMNOPTXY]{2,3} [0-9]{4}', 'ABE 6789')
```
Out[218…
```
<re.Match object; span=(0, 8), match='ABE 6789'>
```
In [220…
```
re.fullmatch(r'[^ΓΔΘΛΞΠΣΦΨΩ0-9]{2,3} [0-9]{4}', 'ABE 6789')
```
Out[220…
```
<re.Match object; span=(0, 8), match='ABE 6789'>
```
In [223…
```
re.search(r'(a+)', 'aaaaaaaaaaaaaaa')
```
Out[223…
```
<re.Match object; span=(0, 15), match='aaaaaaaaaaaaaaa'>
```
In [224…
```
re.search(r'(a+?)', 'aaaaaaaaaaaaaaa')
```
Out[224…
```
<re.Match object; span=(0, 1), match='a'>
```
In [226…
```
re.search(r'(an)+?', 'banana')
```

Out[226… <re.Match object; span=(1, 3), match='an'>

In [ ]: