Αρχεία

In [1]:
```
!ls -l mim2gene.txt
```

```
-rw-r--r--@ 1 admin  staff  904816 Mar 17 17:12 mim2gene.txt
```

In [2]:
```
!pwd
```

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

In [237…
```
# https://www.omim.org/static/omim/data/mim2gene.txt
with open('mim2gene.txt') as f:
    d = f.read()
```

In [239…
```
print(d[:1000])
```

```
# Copyright (c) 1966-2021 Johns Hopkins University. Use of this file adheres t
o the terms specified at https://omim.org/help/agreement.
# Generated: 2021-03-18
# This file provides links between the genes in OMIM and other gene identifier
s.
# THIS IS NOT A TABLE OF GENE-PHENOTYPE RELATIONSHIPS.
# MIM Number    MIM Entry Type (see FAQ 1.3 at https://omim.org/help/faq)
        Entrez Gene ID (NCBI)   Approved Gene Symbol (HGNC)     Ensembl Gene I
D (Ensembl)
100050  predominantly phenotypes
100070  phenotype       100329167
100100  phenotype
100200  predominantly phenotypes
100300  phenotype
100500  moved/removed
100600  phenotype
100640  gene    216     ALDH1A1 ENSG00000165092
100650  gene/phenotype  217     ALDH2   ENSG00000111275
100660  gene    218     ALDH3A1 ENSG00000108602
100670  gene    219     ALDH1B1 ENSG00000137124
100675  predominantly phenotypes
100678  gene    39      ACAT2   ENSG00000120437
100680  moved/removed
100690  gene    1134    CHRNA1  ENSG00000138435
100700  predominantly phenotypes
100710  gene    1140    CHRNB1  ENSG0000017
```

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

Out[6]:
```
904816
```

In [7]:
```
f = open('mim2gene.txt')
```

In [8]:
```
d = f.read()
```

In [9]:
```
f.close()
```

In [13]:
```
with open('mim2gene.txt') as f:
    for line in f:
        #print (line)
        break
```

In [ ]:

```
with open('mim2gene.txt') as f:
    for line in f:
        print (line.strip())
        #break
```

In [15]:

```
with open('mim2gene.txt') as f:
    for line_number, line in enumerate(f):
        print (line.strip())
        if line_number > 10:
            break
```

```
# Copyright (c) 1966-2021 Johns Hopkins University. Use of this file adheres t
o the terms specified at https://omim.org/help/agreement.
# Generated: 2021-03-17
# This file provides links between the genes in OMIM and other gene identifier
s.
# THIS IS NOT A TABLE OF GENE-PHENOTYPE RELATIONSHIPS.
# MIM Number    MIM Entry Type (see FAQ 1.3 at https://omim.org/help/faq)
        Entrez Gene ID (NCBI)   Approved Gene Symbol (HGNC)     Ensembl Gene I
D (Ensembl)
100050  predominantly phenotypes
100070  phenotype       100329167
100100  phenotype
100200  predominantly phenotypes
100300  phenotype
100500  moved/removed
100600  phenotype
```

In [16]:

```
with open('mim2gene.txt') as f:
    first_line = f.readline()
    print (first_line)
```

```
# Copyright (c) 1966-2021 Johns Hopkins University. Use of this file adheres t
o the terms specified at https://omim.org/help/agreement.

```

In [17]:

```
with open('mim2gene.txt') as f:
    first_line = f.readline()
    print (first_line)
    second_line = f.readline()
    print (second_line)
```

```
# Copyright (c) 1966-2021 Johns Hopkins University. Use of this file adheres t
o the terms specified at https://omim.org/help/agreement.

# Generated: 2021-03-17

```

In [18]:

```
with open('mim2gene.txt') as f:
    for i in range(10):
        line = f.readline()
        print (line.strip())
```

```
# Copyright (c) 1966-2021 Johns Hopkins University. Use of this file adheres t
o the terms specified at https://omim.org/help/agreement.
# Generated: 2021-03-17
# This file provides links between the genes in OMIM and other gene identifier
s.
# THIS IS NOT A TABLE OF GENE-PHENOTYPE RELATIONSHIPS.
# MIM Number    MIM Entry Type (see FAQ 1.3 at https://omim.org/help/faq)
        Entrez Gene ID (NCBI)   Approved Gene Symbol (HGNC)     Ensembl Gene I
D (Ensembl)
100050  predominantly phenotypes
100070  phenotype       100329167
100100  phenotype
100200  predominantly phenotypes
```

```
100300	phenotype
```

In [19]:

```python
with open('mim2gene.txt') as f:
    lines = f.readlines()
```

In [20]:

```python
len(lines)
```

Out[20]:

```
27097
```

In [21]:

```python
print (lines[100])
```

```
102800	phenotype
```

In [22]:

```python
f = open('mim2gene.txt')
l_1 = f.readline()
```

In [23]:

```python
print (l_1)
```

```
# Copyright (c) 1966-2021 Johns Hopkins University. Use of this file adheres t
o the terms specified at https://omim.org/help/agreement.
```

In [24]:

```python
f.readline()
```

Out[24]:

```
'# Generated: 2021-03-17\n'
```

In [25]:

```python
f.readline()
```

Out[25]:

```
'# This file provides links between the genes in OMIM and other gene identifie
rs.\n'
```

In [26]:

```python
f.close()
```

In [27]:

```python
f = open('mim2gene.txt')
```

In [28]:

```python
f.readline()
```

Out[28]:

```
'# Copyright (c) 1966-2021 Johns Hopkins University. Use of this file adheres
to the terms specified at https://omim.org/help/agreement.\n'
```

In [29]:

```python
f = open('mim2gene.txt')
for l in f:
    #print ()
    pass
```

In [30]:

```python
f.readline()
```

Out[30]:

```
''
```

In [31]:

```python
f = open('mim2gene.txt')

while True:
    line = f.readline()
    #if line == ''
    if not line:
        break
f.close()
```

In [36]:

```
s = 'Mitsos'
k = 'Kwstas'
```

In [34]:

```
if '':
    print ('1')
else:
    print ('2')
```

```
2
```

In [41]:

```
with open('my_precious.txt', 'w') as f:
    f.write(s)
    f.write(k)
```

In [42]:

```
!cat my_precious.txt
```

```
MitsosKwstas
```

Για windows:

```
!type my_precious.txt
```

In [43]:

```
with open('my_precious.txt', 'w') as f:
    f.write(s + '\n')
    f.write(k + '\n')
```

In [44]:

```
!cat my_precious.txt
```

```
Mitsos
Kwstas
```

In [48]:

```
a = [1,2,3,4,5,6]

with open('my_precious.txt', 'w') as f:
    for i in a:
        f.write(str(i) + '\n')
```

In [49]:

```
!cat my_precious.txt
```

```
1
2
3
4
5
6
```

In [50]:

```
with open('my_precious.txt', 'w') as f:
    f.write(s + ' ' + k + '\n')
```

In [51]:

```
!cat my_precious.txt
```

```
Mitsos Kwstas
```

In [53]:

```
a = ['kwstas', 'mitsos', 'maria', 'sadfzsdfsadc']

with open('my_precious.txt', 'w') as f:
    f.write(' '.join(a) + '\n')
```

In [54]:

```
!cat my_precious.txt
```

```
kwstas mitsos maria sadfzsdfsadc
```

In [55]:

```
with open('my_precious.txt', 'a') as f:
    f.write('hello world\n')
```

In [56]:

```
!cat my_precious.txt
```

```
kwstas mitsos maria sadfzsdfsadc
hello world
```

In [57]:

```
with open('my_precious.txt', 'r') as f:
    d = f.read()
print (d)
```

```
kwstas mitsos maria sadfzsdfsadc
hello world
```

In [58]:

```
a = [1,2,3,4,5,6]
```

In [61]:

```
with open('double.txt', 'w') as f:
    for i in a:
        f.write(str(i*2) + '\n')


with open('half.txt', 'w') as f:
    for i in a:
        f.write(str(i/2) + '\n')
```

In [60]:

```
!cat double.txt
```

```
2
4
6
8
10
12
```

In [62]:

```
!cat half.txt
```

```
0.5
1.0
1.5
2.0
2.5
3.0
```

In [63]:

```
with open('double.txt', 'w') as f1, open('half.txt', 'w') as f2:
    for i in a:
        f1.write(str(i*2)+'\n')
        f2.write(str(i/1)+'\n')
```

In [64]:

```
with open('half2.txt', 'a') as f:
    f.write('asdf')
```

In [65]:

```
!cat half2.txt
```

```
asdf
```

In [67]:

```
with open('mim2gene.txt') as f:
    for l in f:
        if l[0] == '#':
            continue

        print (l)
        break
```

```
100050  predominantly phenotypes
```

In [72]:

```
l.strip('\n').split('\t')
```

Out[72]:

```
['100050', 'predominantly phenotypes', '', '', '']
```

In [77]:

```
l
```

Out[77]:

```
'100050\tpredominantly phenotypes\t\t\t\n'
```

In [78]:

```
l.strip('\n')
```

Out[78]:

```
'100050\tpredominantly phenotypes\t\t\t'
```

In [79]:

```
l.strip('\n').split('\t')
```

Out[79]:

```
['100050', 'predominantly phenotypes', '', '', '']
```

In [80]:

```
'ababa'.split('a')
```

Out[80]:

```
['', 'b', 'b', '']
```

In [81]:

```
a = [[1,2,3], [4,5,6], [7,8,9]]
```

String formatting

In [82]:

```
name = 'Mitsos'
age = 40
```

In [84]:

```
'my name is ' + name + ' and my age is: ' + str(age)
```

Out[84]:

```
'my name is Mitsos and my age is: 40'
```

In [85]:

```
'my name is {} and my age is: {}'.format(name, age)
```

Out[85]:

```
'my name is Mitsos and my age is: 40'
```

In [86]:

```
'my name is {onoma} and my age is: {hlikia}'.format(onoma=name, hlikia=age)
```

Out[86]:

```
'my name is Mitsos and my age is: 40'
```

In [87]:

```
f'my name is {name} and my age is: {age}'
```

Out[87]:

```
'my name is Mitsos and my age is: 40'
```

```
In [89]:  'my name is %s and my age is: %s' % (name, age)

Out[89]: 'my name is Mitsos and my age is: 40'
```

```
In [94]:  a=234/1345
           a

Out[94]: 0.17397769516728626
```

```
In [95]:  'the result is {} .'.format(a)

Out[95]: 'the result is 0.17397769516728626 .'
```

```
In [99]:  'the result is {0:.3f} .'.format(a)

Out[99]: 'the result is 0.174 .'
```

pass

```
In [102…  a = 1

          if a == 1:
              print ('1')
          else:
              #print ('not 1')
              pass

          1
```

```
In [103…  for i in [1,2,3,5]:
              pass
```

```
In [104…  pass
```

```
In [105…  print (1)
          pass
          print (2)

          1
          2
```

```
In [106…  if a==1: # too complicated
              pass
          else:
              ....

            File "<ipython-input-106-9a1fdd3b174b>", line 4
              ....
                 ^
          SyntaxError: invalid syntax
```

ternary operator

In [109…

```
age = 10
if age >= 18:
    state = 'adult'
else:
    state = 'not adult'
```

In [110…

```
state
```

Out[110…

```
'not adult'
```

In [112…

```
state = 'adult' if age >= 18 else 'not adult'
```

is operator

In [113…

```
a=10
```

In [114…

```
type(a) is int
```

Out[114…

```
True
```

In [120…

```
a=10
b=10
a is b
```

Out[120…

```
True
```

In [116…

```
a==b
```

Out[116…

```
True
```

In [119…

Out[119…

```
True
```

In [121…

```
a=1000
b=1000
a is b
```

Out[121…

```
False
```

In [122…

```
a = "mitsos"
b = "mitsos"
a==b
```

Out[122…

```
True
```

In [123…

```
a is b
```

Out[123…

```
True
```

In [124…
```
a = 1000
b = 1000

a == b
```
Out[124…
```
True
```

In [126…
```
a is b
```
Out[126…
```
False
```

In [127…
```
a=b
```

In [128…
```
a is b
```
Out[128…
```
True
```

In [129…
```
b += 1
```

In [130…
```
a
```
Out[130…
```
1000
```

In [131…
```
b
```
Out[131…
```
1001
```

In [132…
```
a = 1000
b = 1000
a=b

a+=1
print(b)
```
```
1000
```

In [133…
```
a = [1,2,3]
b = [1,2,3]

a=b

b.append(4)
print (a)
```
```
[1, 2, 3, 4]
```

In [134…
```
a = [1,2,3]
b = [1,2,3]

b.append(4)
print (a)
```
```
[1, 2, 3]
```

```
In [135…  a = [1,2,3]
          b = [1,2,3]

          a==b

Out[135…  True

In [136…  a is b

Out[136…  False

In [137…  a = [1,2,3]
          b = [1,2,3]

          a=b

          a is b

Out[137…  True

In [139…  a = 3
          def f(x):
              a=4

          f(10)
          print(a)

          3

In [141…  a=[1,2,3]
          def f(x):
              a.append(5)

          f(10)
          print (a)

          [1, 2, 3, 5]

In [142…  def f(a):
              a=4

          a=2
          f(a)
          print (a)

          2

In [143…  def f(a):
              a.append(4)

          a=[1,2,3]
          f(a)
          print (a)

          [1, 2, 3, 4]

 In [ ]:
```

In [146…

```
int(4).__add__(6)
```

Out[146…

```
10
```

lambda functions

In [147…

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

def f(x):
    return x%2 == 1

list(filter(f, a))
```

Out[147…

```
[1, 3, 5, 7, 9]
```

In [148…

```
def g(x):
    return x*2

list(map(g, a))
```

Out[148…

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

In [149…

```
def g(x):
    return x*2
```

In [150…

```
g(10)
```

Out[150…

```
20
```

In [156…

```
g = lambda x : (x*2, x/2)
```

In [160…

```
a,b = g(10)
print(a)
print (b)
```

```
20
5.0
```

In [162…

```
add = lambda a,b: a+b
```

In [163…

```
add(5,4)
```

Out[163…

```
9
```

In [154…

```
def g(x):
    print ("hello")
    return x*2
```

In [164…

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

In [167…

```
list(map(lambda x:x*2, a))
```

Out[167…

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

```
In [168…  list(filter(lambda x:x%2==1, a))

Out[168…  [1, 3, 5, 7, 9]

In [171…  a = ['edessa', 'Bethleem', 'Thessaloniki']

In [172…  def f(x):
              return x.count('e')

          sorted(a, key=f)

Out[172…  ['Thessaloniki', 'edessa', 'Bethleem']

In [173…  sorted(a, key= lambda x:x.count('e') )

Out[173…  ['Thessaloniki', 'edessa', 'Bethleem']

In [174…  min(a, key=lambda x:x.count('e') )

Out[174…  'Thessaloniki'

In [175…  max(a, key=lambda x:x.count('e') )

Out[175…  'Bethleem'

In [176…  def f(x):

              b = lambda y : y*2
              return b(x)

In [177…  f(10)

Out[177…  20

In [178…  a = [lambda x:x*2, lambda x:x/2]

In [179…  a[0](10)

Out[179…  20

In [180…  a[1](10)

Out[180…  5.0

In [181…  a = {}

In [222…  a[lambda x:x*2] = 4

In [193…  hash(lambda x:x*2)

Out[193…  8781085966026
```

In [194…

```
list(a.keys())
```

Out[194…

```
[<function __main__.<lambda>(x)>,
 <function __main__.<lambda>(x)>,
 <function __main__.<lambda>(x)>]
```

In [223…

```
a
```

Out[223…

```
{<function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4,
 <function __main__.<lambda>(x)>: 4}
```

In [224…

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

In [225…

```
[2*x for x in a]
```

Out[225…

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

In [226…

```
list(map(lambda x:x*2, a))
```

Out[226…

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

In [227…

```
a=[[1,2], [2,3]]
```

In [228…

```
b = [tuple(x) for x in a]
```

In [229…

```
print (b)
```

```
[(1, 2), (2, 3)]
```

In [231…

```
sorted(b, key=lambda x:x[1], reverse=True)
```

Out[231…

```
[(2, 3), (1, 2)]
```

In [232…

```
sorted(a, key=lambda x:x[1], reverse=True)
```

```
Out[232… [[2, 3], [1, 2]]

In [233… [sum(x) for x in a]

Out[233… [3, 5]

In [234… [sum(x) for x in b]

Out[234… [3, 5]

In [ ]:
```