

```

In [2]: !cd
        C:\Users\user

In [ ]: f = open('mitsos.txt')

In [4]: f = open('C:/Users/user/Downloads/gwas_catalog_v1.0-associations_e105_r202')

In [5]: f.close()

In [6]: f = open(r'C:\Users\user\Downloads\')
        Input In [6]
        f = open(r'C:\Users\user\Downloads\')
        ^
SyntaxError: EOL while scanning string literal

In [7]: f = open('C:/Users/user/results.txt')

In [8]: f

Out[8]: <_io.TextIOWrapper name='C:/Users/user/results.txt' mode='r' encoding='cp1253'>

In [9]: type(f)

Out[9]: _io.TextIOWrapper

In [10]: s = f.read()

In [11]: s

Out[11]: 'this is a fantastic file\nvery precious data\nmuch science\nnobel\n'

In [12]: print (s)

this is a fantastic file
very precious data
much science
nobel

In [13]: g = f.read()

In [14]: print (g)

In [15]: len(g)

Out[15]: 0

In [16]: f.close()

In [17]: f = open('C:/Users/user/results.txt')

In [18]: g = f.read()

```

```
In [19]: g
Out[19]: 'this is a fantastic file\nvery precious data\nmuch science\nnobel\n'

In [20]: f.close()

In [21]: f = open('C:/Users/user/results.txt')

In [22]: line = f.readline()

In [23]: print (line)
this is a fantastic file

In [24]: g = f.read()

In [25]: print (g)
very precious data
much science
nobel

In [26]: f.close()

In [27]: f = open('C:/Users/user/results.txt')

In [28]: a = f.readline()

In [29]: b = f.readline()

In [30]: print (b)
very precious data

In [31]: print (f.read())
much science
nobel

In [32]: f.seek(10)
Out[32]: 10

In [33]: f.readline()
Out[33]: 'fantastic file\n'

In [34]: f.close()

In [35]: f = open('C:/Users/user/results.txt')
```

```
In [44]: f = open('C:/Users/user/results.txt')
while True:
    line = f.readline()
    print (line)
    if not line:
        break
f.close()
```

this is a fantastic file

very precious data

much science

nobel

```
In [45]: f = open('C:/Users/user/results.txt')
while True:
    line = f.readline()
    print (line)
    if len(line) == 0:
        break
f.close()
```

this is a fantastic file

very precious data

much science

nobel

```
In [46]: f = open('C:/Users/user/results.txt')
while True:
    line = f.readline()
    print (line)
    if line == '':
        break
f.close()
```

this is a fantastic file

very precious data

much science

nobel

```
In [47]: for x in 'SDAFGFDS':
    print (x)
```

S  
D  
A  
F  
G  
F  
D

c

```
In [48]: for x in [5,6,7,]:  
        print (x)
```

5  
6  
7

```
In [70]: f = open('C:/Users/user/results.txt')  
        for line in f:  
            print (line)  
        f.close()
```

this is a fantastic file  
  
very precious data  
  
much science  
  
nobel

```
In [71]: with open('C:/Users/user/results.txt') as f:  
        for l in f:  
            print (l)
```

this is a fantastic file  
  
very precious data  
  
much science  
  
nobel

```
In [72]: l = []
```

```
In [73]: with open('C:/Users/user/results.txt') as f:  
        lc = 0  
        for l in f:  
            lc += 1  
            if lc<3:  
                continue  
            print (l)  
            break
```

much science

```
In [91]: with open('C:/Users/user/results.txt') as f:  
        for lc, l in enumerate(f):  
            if lc < 6:  
                continue  
            print (l)  
            break
```

dyjffjghjfkjhjthjhkhkhjfgfgmgdd.fgjdaalkgjshdlfkghsdldkfjghsldkfjghsldkjghskl  
dfjghlsdkfjghlsdkdfjghlsdkdfjghskldjfgghsldkfjghsldkjgh sljgsldkj ghsldjgh sl  
dkgjh skldfjgh skldfjghsldkjghsldkfjgh skldjg hslfdkjghsldkjg hslfdkjgh sld  
kjgh sldkgjh sldkjg hsdldfkjgh sdfkjg hslldkfjg hslldkfjg hslldkfjg hslldkfjg hs  
ldkfjgh sldkfjg sldkfjgh sldkjgh sldfjgh sldkfjgh sldkfjgh sldkfjg hslldkfjg  
hslldkfjgh sldkfjgh sldkfjgh fjkldh

In [ ]:

```
In [79]: my_list = []
with open('C:/Users/user/results.txt') as f:

    for lc, l in enumerate(f):
        if lc in {1,2}:
            my_list.append(l)
```

In [78]: my\_list

Out[78]: ['very precious data\n', 'much science. bravo!\n']

```
In [80]: with open('C:/Users/user/results.txt') as f:

        my_list = [l for lc, l in enumerate(f) if lc in {1,2}]
```

In [81]: my\_list

Out[81]: ['very precious data\n', 'much science. bravo!\n']

```
In [82]: my_list = [l for lc, l in enumerate(open('C:/Users/user/results.txt')) if
```

In [83]: my\_list

Out[83]: ['very precious data\n', 'much science. bravo!\n']

```
In [84]: a = 'ghjkgfghj\n'
```

In [85]: a

Out[85]: 'ghjkgfghj\n'

```
In [86]: a.strip()
```

Out[86]: 'ghjkgfghj'

```
In [87]: ' sfdagsagszdfg '.strip()
```

Out[87]: 'sfdagsagszdfg'

```
In [88]: 'sdfasdfasdf\n'[:-1]
```

Out[88]: 'sdfasdfasdf'

```
In [89]: 'sdfasdfasdf\n'.replace('\n', '')
```

Out[89]: 'sdfasdfasdf'

In [ ]:

In [ ]:

In [ ]:

```
In [41]: print ('mitsos', end=' - ')
        print ('alex')
```

mitsos - alex

```
In [42]: bool('sdfsdfg')
```

Out[42]: True

```
In [43]: bool('')
```

Out[43]: False

```
In [92]: my_list = []
        with open('C:/Users/user/results.txt') as f:

            data = f.read()
            print (len(data))
```

488

```
In [93]: my_list = []
        with open('C:/Users/user/alex.docx') as f:

            data = f.read()
            print (len(data))
```

---

**UnicodeDecodeError** Traceback (most recent call last)

Input In [93], in <cell line: 2>()

```
1 my_list = []
2 with open('C:/Users/user/alex.docx') as f:
----> 4     data = f.read()
      5 print (len(data))
```

File ~\miniconda3\lib\encodings\cp1253.py:23, in IncrementalDecoder.decode(self, input, final)

```
22 def decode(self, input, final=False):
--> 23     return codecs.charmap_decode(input,self.errors,decoding_tabl
e)[0]
```

**UnicodeDecodeError**: 'charmap' codec can't decode byte 0xd2 in position 16: character maps to <undefined>

```
In [107... my_list = []
        with open('C:/Users/user/alex2.txt') as f:

            data = f.read()
            print (len(data))
```

379

```
In [108... print (data)
```

Xf/lgk jh;flsgkhjsfkj ghsfkldghsldkjghsldkfjg hldakfjgh sldfkj ghslldkfjgh s  
kldfjgh skldfjgh skldfjgh ldskfj ghslldkfj ghslldkfjg sldkfjgh sldjgh sldjgh  
sldkfjgh sldkjgh sldkjgh sldkfjg sldkfjg hslldkjgh sldkfjg hslldfjkg hslldkjgh  
ldfjgh sldkfjgh

1. Xghfg
2. Hfd
3. Hfd
4. Ghfd
5. Ghd

6. Fghfg  
7. h  
sldkfjg sldkfjgh slkdfjg sdfklgh skldfgh sldkjg hsldfjgh sldkfjgh sldj ghls  
dkf h

```
In [94]: def f(l):  
         return sum(l)
```

```
In [95]: f([10, 20, 30])
```

Out[95]: 60

```
In [99]: def f(l):  
         if not len(l):  
             return 0  
         return l[0] + f(l[1:])  
  
         f([10, 20, 30])
```

Out[99]: 60

```
In [105... l = [['Helen', 23, 8], ['Kostas', 25, 9], ['Alex', 22, 9], ['Maria', 24, 7]
```

```
In [106... s = 'a ορετςερετςρε οδφγ zfdgsdfg'
```

```
In [101... l = [4,5,4,3,4,5,6,7,6,1,5,4]
```

```
In [102... min(l)
```

Out[102... 1

```
In [104... def f(l):  
         if len(l) ==2:  
             if l[0] < l[1]:  
                 return l[0]  
             return l[1]  
  
         if l[0] < l[1]:  
             return f( [l[0]] + l[2:] )  
  
         return f([l[1]] + l[2:])  
  
         f(l)
```

Out[104... 1

```
In [110... my_list = []  
with open('C:/Users/user/findings.txt', 'w') as f:  
  
    f.write('cvghjklcvbnjkfyucvhjkcvcgh')
```

```
In [ ]:
```

```
In [ ]:
```

In [ ]:

```
In [115]: my_list = []
with open('C:/Users/user/findings.txt', 'x') as f:

    f.write('SUCH RESEARCH! WOW!!!!')
```

```
-----
FileExistsError                                Traceback (most recent call last)
Input In [115], in <cell line: 2>()
      1 my_list = []
----> 2 with open('C:/Users/user/findings.txt', 'x') as f:
      4     f.write('SUCH RESEARCH! WOW!!!!')
```

**FileExistsError:** [Errno 17] File exists: 'C:/Users/user/findings.txt'

```
In [118]: my_list = []
with open('C:/Users/user/findings.txt', 'w') as f:

    f.write('SUCH RESEARCH! WOW!!!!\n')
    f.write('cannot get better than this!\n')
```

In [ ]:

```
In [36]: data = []
with open('C:/Users/user/Downloads/gwas_catalog_v1.0-associations_e105_r201
          encoding='iso-8859-1') as f:
    line = f.readline()
    header = line.strip().split('\t')

    for line_counter, line in enumerate(f):
        l = f.readline()

        if line_counter % 10_000 == 0:
            print (f'Progress: {line_counter}')

        l_list = l.strip().split('\t')

        data.append(l_list)

print (f' read {line_counter} lines')
```

```
Progress: 0
Progress: 10000
Progress: 20000
Progress: 30000
Progress: 40000
Progress: 50000
Progress: 60000
Progress: 70000
Progress: 80000
Progress: 90000
Progress: 100000
Progress: 110000
Progress: 120000
Progress: 130000
Progress: 140000
Progress: 150000
Progress: 160000
Progress: 170000
read 175920 lines
```



```
In [8]: data[100]
```

```
Out[8]: ['2019-02-14',
        '29507422',
        'Hoffmann TJ',
        '2018-03-05',
        'Nat Genet',
        'www.ncbi.nlm.nih.gov/pubmed/29507422',
        'A large electronic-health-record-based genome-wide study of serum lipid
s.',
        'High density lipoprotein cholesterol levels',
        '76,627 European ancestry individuals, 7,795 Hispanic individuals, 6,855 E
ast Asian ancestry individuals, 2,958 African American individuals, 439 Sou
th Asian ancestry individuals',
        'NA',
        '2p24.1',
        '2',
        '21041028',
        'NR',
        'APOB',
        '',
        '',
        '',
        'ENSG00000084674',
        '',
        '',
        '',
        'rs1367117-G',
        'rs1367117',
        '0',
        '1367117',
        'missense_variant',
        '0',
        'NR',
        '3E-6',
        '5.522878745280337',
        '',
        '0.018',
        'unit increase',
        'Affymetrix [at least 7091467] (imputed)',
        'N']
```

```
In [134]: ''.strip().split('\t')
```

```
Out[134]: ['']
```

```
In [11]: dict(zip(header, data[100]))
```

```
Out[11]: {'DATE ADDED TO CATALOG': '2019-02-14',
        'PUBMEDID': '29507422',
        'FIRST AUTHOR': 'Hoffmann TJ',
        'DATE': '2018-03-05',
        'JOURNAL': 'Nat Genet',
        'LINK': 'www.ncbi.nlm.nih.gov/pubmed/29507422',
        'STUDY': 'A large electronic-health-record-based genome-wide study of seru
m lipids.',
        'DISEASE/TRAIT': 'High density lipoprotein cholesterol levels',
        'INITIAL SAMPLE SIZE': '76,627 European ancestry individuals, 7,795 Hispan
ic individuals, 6,855 East Asian ancestry individuals, 2,958 African Americ
an individuals, 439 South Asian ancestry individuals',
        'REPLICATION SAMPLE SIZE': 'NA',
        'REGION': '2p24.1',
        'CHR_ID': '2',
        'CHR_POS': '21041028',
```

```

'REPORTED_GENE(S)': 'NR',
'MAPPED_GENE': 'APOB',
'UPSTREAM_GENE_ID': '',
'DOWNSTREAM_GENE_ID': '',
'SNP_GENE_IDS': 'ENSG00000084674',
'UPSTREAM_GENE_DISTANCE': '',
'DOWNSTREAM_GENE_DISTANCE': '',
'STRONGEST SNP-RISK ALLELE': 'rs1367117-G',
'SNPS': 'rs1367117',
'MERGED': '0',
'SNP_ID_CURRENT': '1367117',
'CONTEXT': 'missense_variant',
'INTERGENIC': '0',
'RISK ALLELE FREQUENCY': 'NR',
'P-VALUE': '3E-6',
'PVALUE_MLOG': '5.522878745280337',
'P-VALUE (TEXT)': '',
'OR or BETA': '0.018',
'95% CI (TEXT)': 'unit increase',
'PLATFORM [SNPS PASSING QC]': 'Affymetrix [at least 7091467] (imputed)',

```

```

In [15]: d = {}

for item in data:
    for h, v in zip(header, item):
        if not h in d:
            d[h] = []

        d[h].append(v)

```

```

In [27]: len(set(d['FIRST AUTHOR']))

```

```

Out[27]: 123

```

```

In [28]: set(d['FIRST AUTHOR'])

```

```

Out[28]: {'Ahmed S',
'Allen RJ',
'Almgren P',
'Almli LM',
'Anney RJL',
'Arpawong TE',
'Astle WJ',
'Bei JX',
'Benyamin B',
'Biernacka JM',
'Bonas-Guarch S',
'Cha S',
'Chan JP',
'Chang X',
'Chaturvedi S',
'Chen H',
'Chenoweth MJ',
'Christophersen IE',
'Clarke TK',
'Coleman JRI',
'Conti DV',
'Cooper JD',
'Corre T',
'Darlow JM',
'Day FR',
'Delgado DA',

```

'Dong C',  
'Dudenkov TM',  
'Ferreira MA',  
'Gao B',  
'Gorski M',  
'Graff M',  
'Guo Q',  
'Hammerschlag AR',  
'Haryono SJ',  
'Hinks A',  
'Hofer P',  
'Hoffmann TJ',  
'Hong X',  
'Hu X',  
'Ikram MA',  
'Ilboudo Y',  
'Jia P',  
'Jonsson L',  
'Jun GR',  
'Justice AE',  
'Kawaguchi T',  
'Kerr KF',  
'Kim KW',  
'Kim M',  
'Kimura M',  
'Konte B',  
'Kristiansen W',  
'Kunz M',  
'Lee MH',  
'Lee MK',  
'Lee TH',  
'Lencer R',  
'Lessard CJ',  
'Li C',  
'Li D',  
'Li J',  
'Li M',  
'Litchfield K',  
'Liu JZ',  
'Liu Y',  
'Lu AT',  
'Lutz SM',  
'Lv H',  
'Mack S',  
'Magvanjav O',  
'Marenholz I',  
'McKay JD',  
'Michailidou K',  
'Milne RL',  
'Miron J',  
'Moore CB',  
'Moore KN',  
'Morris AP',  
'Morton LM',  
'Munz M',  
'Nakada TA',  
'Ng E',  
'Nolte IM',  
'Persad PJ',  
'Qian DC',  
'Randall CL',  
'Ravenhall M',  
'Ren HY',  
'Saccone NL',

```
'Sakamoto Y',
'Sanchez-Juan P',
'Sanchez-Roige S',
'Scelo G',
'Seyerle AA',
'Shah AA',
'Shen X',
'Sobota RS',
'Sud A',
'Sugier PE',
'Suh Y',
'Suhre K',
'Sun Y',
'Tachmazidou I',
'Tapper W',
'Thompson AG',
'Tian C',
'Tomer Y',
'Turley P',
'Wang Z',
'Ward-Caviness CK',
'Wattacheril J',
'Winkler TW',
'Witt SH',
'Xu W',
'Yashin AI',
'Yeo A',
'Yin X',
'Yucesoy B',
'Zai CC',
'Zhang Y',
'Zhang YB',
'Zhou H'}
```

```
In [30]: set([x for x in d['FIRST AUTHOR'] if 'idou' in x])
```

```
Out[30]: {'Michailidou K', 'Tachmazidou I'}
```

```
In [120... line
```

```
Out[120... 'DATE ADDED TO CATALOG\tPUBMEDID\tFIRST AUTHOR\tDATE\tJOURNAL\tLINK\tSTUDY\t
DISEASE/TRAIT\tINITIAL SAMPLE SIZE\tREPLICATION SAMPLE SIZE\tREGION\tCHR_I
D\tCHR_POS\tREPORTED GENE(S)\tMAPPED_GENE\tUPSTREAM_GENE_ID\tDOWNSTREAM_GEN
E_ID\tSNP_GENE_IDS\tUPSTREAM_GENE_DISTANCE\tDOWNSTREAM_GENE_DISTANCE\tSTRON
GEST SNP-RISK ALLELE\tSNPS\tMERGED\tSNP_ID_CURRENT\tCONTEXT\tINTERGENIC\tRI
SK ALLELE FREQUENCY\tP-VALUE\tPVALUE_MLOG\tP-VALUE (TEXT)\tOR or BETA\t95%
CI (TEXT)\tPLATFORM [SNPS PASSING QC]\tCNV\n'
```

```
In [122... line.strip().split('\t')
```

```
Out[122... ['DATE ADDED TO CATALOG',
'PUBMEDID',
'FIRST AUTHOR',
'DATE',
'JOURNAL',
'LINK',
'STUDY',
'DISEASE/TRAIT',
'INITIAL SAMPLE SIZE',
'REPLICATION SAMPLE SIZE',
'REGION',
'CHR_ID',
'CHR_POS',
```

'REPORTED\_GENE(S) ',  
'MAPPED\_GENE ',  
'UPSTREAM\_GENE\_ID ',  
'DOWNSTREAM\_GENE\_ID ',  
'SNP\_GENE\_IDS ',  
'UPSTREAM\_GENE\_DISTANCE ',  
'DOWNSTREAM\_GENE\_DISTANCE ',  
'STRONGEST SNP-RISK ALLELE ',  
'SNPS ',  
'MERGED ',  
'SNP\_ID\_CURRENT ',  
'CONTEXT ',  
'INTERGENIC ',  
'RISK ALLELE FREQUENCY ',  
'P-VALUE ',  
'PVALUE\_MLOG ',  
'P-VALUE (TEXT) ',  
'OR or BETA ',  
'95% CI (TEXT) ',  
'PLATFORM [SNPS PASSING QC] ',

In [ ]:

In [ ]:

In [ ]:

In [ ]: