

Introduction to



python

for scientific computing

- Lecture 3

Review of previous lectures

Variables and Types

1. Which of the following is of the type `float` ?:

`"hello"`

`7.4`

`3`

`-2`

`"2.1"`

2. Match the following variables with their type:

`var1 = 54`

`var2 = [1, 2, 7, 1, 24]`

`var3 = 2.98`

`var4 = True`

Variables and Types

1. Which of the following is of the type `float` ?:

7.4

2. Match the following variables with their type:

<code>var1 = 54</code>	<code>integer</code>
<code>var2 = [1, 2, 7, 1, 24]</code>	<code>list</code>
<code>var3 = 2.98</code>	<code>float</code>
<code>var4 = True</code>	<code>boolean</code>

Literals

All literals have a type:

- Strings (str) 'Hello' "Hi"
- Integers (int) 5
- Floats (float) 3.14
- Boolean (bool) True or False

```
In [11]: type(5)
```

```
Out[11]: int
```

Variables

Used to store values and to assign them a name.

```
In [12]: a = 3.14  
a
```

```
Out[12]: 3.14
```

Lists

A collection of values.

```
In [13]: x = [1,5,3,7,8]
          y = ['a','b','c']
          type(x)
```

```
Out[13]: list
```

Operations

4. What happens if you do `[1, 2, 5, 11] + [87, 2, 43, 3]` ?

5. How do you find out if the variable `x` is present in a the list `mylist` ?

6. How do you find out if 5 is larger than 3 and the integer 4 is the same as the float 4?

Operations

4. What happens if you do `[1, 2, 5, 11] + [87, 2, 43, 3]` ?

`[1, 2, 5, 11, 87, 2, 43, 3]` The lists will be concatenated

5. How do you find out if the variable `x` is present in a the list `mylist` ?

Two answers correct:

1. `x in mylist`

```
2. for l in mylist:
    if l == x:
        print('Found a match')
```

6. How do you find out if 5 is larger than 3 and the integer 4 is the same as the float 4?

`5 > 3 and 4 == 4.0`

Basic operations

Type	Operations
int	+ - / ** % // ...
float	+ - / * % // ...
string	+

Comparison/Logical/Membership operators

Operation	Meaning
<	less than
<=	less than or equal
>	greater than
>=	greater than or equal
==	equal
!=	not equal

Operation	Meaning
and	connects two statements, both conditions having to be fulfilled
or	connects two statements, either conditions having to be fulfilled
not	reverses and/or

Operation	Meaning
in	value in object
not in	value not in object

Sequences

7. How do you select the second element in the variable `mylist = [4,3,8,10]`?

8. Pair the following variables with whether they are mutable or immutable

`var1 = 'my pretty string'`

`var2 = [1,2,3,4,5]`

`var3 = "hello world"`

`var4 = ['a', 'b', 'c', 'd']`

9. Which of the following types are iterable?

- Lists
- Tuples
- Integers
- Strings
- Floats

Sequences

7. How do you select the second element in the variable `mylist = [4,3,8,10]`?

`mylist[1]`

8. Pair the following variables with whether they are mutable or immutable

<code>var1 = 'my pretty string'</code>	immutable
<code>var2 = [1,2,3,4,5]</code>	mutable
<code>var3 = "hello world"</code>	immutable
<code>var4 = ['a', 'b', 'c', 'd']</code>	mutable

9. Which of the following types are iterable?

Lists, tuples, strings

Indexing

Lists (and strings) are an ORDERED collection of elements where every element can be access through an index.

`a[0]` : first item in list a

REMEMBER! Indexing starts at 0 in python

```
In [4]: a = [1,2,3,4,5]
        b = ['a','b','c']
        c = 'a random string'
```

Mutable / Immutable sequences and iterables

Lists are mutable object, meaning you can use an index to change the list, while strings are immutable and therefore not changeable.

An iterable sequence is anything you can loop over, ie, lists and strings.

```
In [ ]: a = [1,2,3,4,5]           # mutable  
        b = ['a','b','c']       # mutable  
        c = 'a random string'   # immutable  
  
#c[0] = 'A'  
#a[0] = 42
```

Another data type: tuples

- A tuple is an immutable sequence of objects
- Unlike a list, nothing can be changed in a tuple
- Still iterable

```
In [ ]: myTuple = (1,2,3,4, 'a', 'b', 'c', [42,43,44])
        #myTuple[0] = 42
        #print(myTuple)
        #print(len(myTuple))
        #for i in myTuple:
        #    print(i)
```

If/ Else statements

10. How do you do to print 'Yes' if x is bigger than y?

In []:

```
x = 10  
y = 5
```

in/not in

How do you tell if an element is in a list/string or not?

In []:

```
a = 2  
b = [1, 2, 3, 4]
```

Loops

How do you print each element in a list:

- With a for loop
- With a while loop

```
In [82]: list_of_things = [5, 6.1, 7, "hello", (2, 3), False, None]
```

```
for thing in list_of_things:  
    print(thing)
```

```
i = 0  
while i < len(list_of_things):  
    print(list_of_things[i])  
    i += 1 # same as i = i + 1
```

```
5  
6.1  
7  
hello  
(2, 3)  
False  
None  
5  
6.1  
7  
hello  
(2, 3)  
False  
None
```


A good Day

1. Your task here is to decide if today is a good day. There are certain criteria that have to be fulfilled for a day to be classified as good:

- Days on a weekend are always good
- Weekdays are good if it's no earlier than 10
- If there's coffee in the kitchen, it makes a weekday good

Complete the below code using AND and OR statements to find out if today is a good day, and print the results

1. Modify your code so coffee only makes a weekday good if there is no rain

```
In [84]: day      = "Monday"
         time     = 7
         coffee   = True
         rain     = False

         todayIsAGoodDay = day in ["Saturday", "Sunday"] or (time > 10 and coffee and not
         rain)
         if todayIsAGoodDay:
             print("Today is a good day!")
         else:
             print("Today is a bad day")
```

Today is a bad day

Questions?

Working on files

```
In [ ]: fruits = ['apple', 'pear', 'banana', 'orange']  
  
for fruit in fruits:  
    print(fruit)
```

```
apple  
pear  
banana  
orange  
fruits.txt (END)
```

```
In [1]: fh = open('../files/fruits.txt', 'r', encoding = 'utf-8')  
  
        for line in fh:  
            print(line)  
  
        fh.close()
```

apple

pear

banana

orange

Additional useful methods:

<code>'string'.strip()</code>	Removes whitespace
<code>'string'.split()</code>	Splits on whitespace into list

```
In [3]: s = '  an example string to split with whitespace in end  '
        sw = s.strip()
        sw
        #l = sw.split()
        #l
        #l = s.strip().split('\t')
        #l
```

```
Out[3]: 'an example string to split with whitespace in end'
```

```
apple  
pear  
banana  
orange  
fruits.txt (END)
```

```
In [5]: fh = open('../files/fruits.txt', 'r', encoding = 'utf-8')  
  
for line in fh:  
    print(line.strip())  
  
fh.close()
```

```
apple  
pear  
banana  
orange
```

Another example

```
ICA      254
Icecream      65
Coop      25.45
ICA      654.21
Pharmacy     39.90
IKEA      2365
ATM        500
SevenEleven  62.60
ICA      278.50
Åhlens     645.20
bank_statement.txt (END)
```

How much money is spent on ICA?


```
fh    = open("../files/bank_statement.txt", "r", encoding = "utf-8")

total = 0

for line in fh:
    expenses = line.strip().split() # split line into list
    store    = expenses[0]          # save what store
    price    = float(expenses[1])   # save the price
    if store == 'ICA':               # only count the price if store is ICA
        total = total + price
fh.close()

print('Total amount spent on ICA is: '+str(total))
```

Slightly more complex...

store	year	month	day	sum	
ICA	2018	08	30	254	
Icecream		2018	09	05	65
Coop	2018	09	08	25.45	
ICA	2018	09	22	654.21	
Pharmacy		2018	09	23	39.90
IKEA	2018	09	25	2365	
ATM	2018	09	28	500	
SevenEleven		2018	09	29	62.60
ICA	2018	09	29	278.50	
Åhlens	2018	10	02	645.20	

bank_statement_extended.txt (END)

How much money is spent on ICA in September?

```

fh    = open("../files/bank_statement_extended.txt", "r", encoding = "utf-8")

total = 0

for line in fh:
    if not line.startswith('store'):
        expenses = line.strip().split()
        store    = expenses[0]
        year     = expenses[1]
        month    = expenses[2]
        day      = expenses[3]
        price    = float(expenses[4])
        if store == 'ICA' and month == '09':    # store has to be ICA and month s
eptember
            total = total + price
fh.close()

out = open("../files/bank_statement_results.txt", "w", encoding = "utf-8")    # o
pen a file for writing the results to
out.write('Total amount spent on ICA in september is: '+str(total))
out.close()

```

```

Total amount spent on ICA in september is: 932.71
bank_statement_results.txt (END)

```

Pseudocode

How to approach a coding task

Problem:

You have a file with a large list of movies from IMDb. You are interested in only one of the movie genres ("Adventure") and the movie has to be very good (rating > 8.5). What you want to know is the title and length (in minutes) of the movie.

```
# Votes | Rating | Year | Runtime | URL | Genres | Title
126807| 8.5|1957|5280|https://images-na.ssl-images-amazon.com/images/M/MV5B0TI5Nzc0OTM5YzBkMS00NjkkLTNmM2U0tjM200dGxN2M5NjNkXkEyXkFqcGdeQXVyNjQ2Mjc5NzM0._V1_.jpg|Drama,War|Paths of Glory
71379| 8.2|1925|4320|https://images-na.ssl-images-amazon.com/images/M/MV5BMzYzMDQyNzA4MjV5BM15BanBnXkFtZTYwNDU5NDU5._V1_.jpg|Adventure,Comedy,Drama,Family|The Gold Rush
700599| 8.3|2009|5760|https://images-na.ssl-images-amazon.com/images/M/MV5BMTk3NDU5NzA4MjV5BM15BanBnXkFtZTYwNDU5NDU5._V1_.jpg|Animation,Adventure,Comedy,Family|Up
27632| 8.3|1928|6840|https://images-na.ssl-images-amazon.com/images/M/MV5BMjQwMjQ0MzgwNjV5BM15BanBnXkFtZTYwNDU5NDU5._V1_.jpg|Biography,Drama,History|The Passion of Joan of Arc
228498| 8.4|1959|8160|https://images-na.ssl-images-amazon.com/images/M/MV5BMjQwMjQ0MzgwNjV5BM15BanBnXkFtZTYwNDU5NDU5._V1_.jpg|Action,Adventure,Crime,Mystery,Thriller|North by Northwest
105342| 8.2|1954|6480|https://images-na.ssl-images-amazon.com/images/M/MV5BMTcwMTU3MjI0V5BM15BanBnXkFtZTYwNDU5NDU5._V1_.jpg|Crime,Drama,Thriller|On the Waterfront
115523| 8.1|1993|7980|https://images-na.ssl-images-amazon.com/images/M/MV5BMTcwMTU3MjI0V5BM15BanBnXkFtZTYwNDU5NDU5._V1_.jpg|Biography,Drama|In the Name of the Father
225098| 8.3|1962|7740|https://images-na.ssl-images-amazon.com/images/M/MV5BMjA4MzI1NDY2NjV5BM15BanBnXkFtZTYwNDU5NDU5._V1_.jpg|Crime,Drama|To Kill a Mockingbird
```

Always write pseudocode!

Pseudocode is a description of what you want to do without actually using proper syntax

```
# Votes | Rating | Year | Runtime | URL | Genres | Title
126807| 8.5 | 1957 | 5280 | https://images-na.ssl-images-amazon.com/images/M/MV5B0TI5Nzc0MTYzBkMS00NjkkLTNmM2UtNjM2ODgxN2M5NjNkXkEyXkFqcGdeQXVyNjQ2MjQ5NzM0._V1_.jpg | Drama, War | Paths of Glory
71379| 8.2 | 1925 | 4320 | https://images-na.ssl-images-amazon.com/images/M/MV5BMzYzMDQyNzA4NV5BM15BanBnXkFtZTYwNDU5NDU5._V1_.jpg | Adventure, Comedy, Drama, Family | The Gold Rush
700599| 8.3 | 2009 | 5760 | https://images-na.ssl-images-amazon.com/images/M/MV5BMTk3NDQ2NzI4NF5BM15BanBnXkFtZTgwNzE1MzEyMTE0._V1_.jpg | Animation, Adventure, Comedy, Family | Up
27632| 8.3 | 1928 | 6840 | https://images-na.ssl-images-amazon.com/images/M/MV5BMjQxMjk5ODUxMl5BM15BanBnXkFtZTgwNDI2MjUwMzE0._V1_.jpg | Biography, Drama, History | The Passion of Joan of Arc
228498| 8.4 | 1959 | 8160 | https://images-na.ssl-images-amazon.com/images/M/MV5BMjQwMTQ0Mzg1Ml5BM15BanBnXkFtZTgwNjE2OTI3MjE0._V1_.jpg | Action, Adventure, Crime, Mystery, Thriller | North by Northwest
105342| 8.2 | 1954 | 6480 | https://images-na.ssl-images-amazon.com/images/M/MV5BMTcwMTU0MjI1OV5BM15BanBnXkFtZTgwNjE2OTI3MjE0._V1_.jpg | Crime, Drama, Thriller | On the Waterfront
115523| 8.1 | 1993 | 7980 | https://images-na.ssl-images-amazon.com/images/M/MV5BMTcwMTU0MjI1OV5BM15BanBnXkFtZTgwMDU0NDkzMTE0._V1_.jpg | Biography, Drama | In the Name of the Father
225098| 8.3 | 1962 | 7740 | https://images-na.ssl-images-amazon.com/images/M/MV5BMjA4MzI1NDY2Nl5BM15BanBnXkFtZTcwMTcyODc5Mw00._V1_.jpg | Crime, Drama | To Kill a Mockingbird
```

Basic Pseudocode:

- Open file
- For each line (movie):
 - Does the line start with #? -> ignore it
 - Otherwise: split the line in columns
 - Isolate the column that contains the genre
 - Is it the right genre?
 - If so, isolate the column that contains the rating
 - Is the rating good enough?
 - If so, get title and timing
 - Then convert timing to minutes
 - Finally, print results
- Close file

```
# Votes | Rating | Year | Runtime | URL | Genres | Title
126807| 8.5|1957|5280|https://images-na.ssl-images-amazon.com/images/M/MV5B0TI5Nzc0MTYzBkMS00NjkkLTNmM2U0tjM2ODgxN2M5NjNkXkEyXkFqcGdeQXVyNjQ2Mjc0NzZM@._V1_.jpg|Drama,War|Paths of Glory
71379| 8.2|1925|4320|https://images-na.ssl-images-amazon.com/images/M/MV5BMzYzMDQyNzA4NV5BM15BanBnXkFtZTYwNDU5NDU5._V1_.jpg|Adventure,Comedy,Drama,Family|The Gold Rush
700599| 8.3|2009|5760|https://images-na.ssl-images-amazon.com/images/M/MV5BMTk3NDE2NzI4NF5BM15BanBnXkFtZTgwNDZlMzEyMTE@._V1_.jpg|Animation,Adventure,Comedy,Family|Up
27632| 8.3|1928|6840|https://images-na.ssl-images-amazon.com/images/M/MV5BMjQxMjk5ODUxM15BM15BanBnXkFtZTgwNDI2MjUwMzE@._V1_.jpg|Biography,Drama,History|The Passion of Joan of Arc
228498| 8.4|1959|8160|https://images-na.ssl-images-amazon.com/images/M/MV5BMjQwMTQ0MzgwN15BM15BanBnXkFtZTgwNjc4ODE4MzE@._V1_.jpg|Action,Adventure,Crime,Mystery,Thriller|North by Northwest
105342| 8.2|1954|6480|https://images-na.ssl-images-amazon.com/images/M/MV5BMTcwMTU3MjI1OV5BM15BanBnXkFtZTgwNjc4ODE4MzE@._V1_.jpg|Crime,Drama,Thriller|On the Waterfront
115523| 8.1|1993|7980|https://images-na.ssl-images-amazon.com/images/M/MV5BMTcwMTU3MjI1OV5BM15BanBnXkFtZTgwMDU4NjkzMTE@._V1_.jpg|Biography,Drama|In the Name of the Father
225098| 8.3|1962|7740|https://images-na.ssl-images-amazon.com/images/M/MV5BMjA4MzI1NDY2N15BM15BanBnXkFtZTcwMTcyODc5Mw@@._V1_.jpg|Crime,Drama|To Kill a Mockingbird
```

Now start building the program bit by bit

- Open file
- For each line (movie):
 - Does the line start with #? -> ignore it

```
# Votes | Rating | Year | Runtime | URL | Genres | Title
126807| 8.5| 1957| 5280| https://images-na.ssl-images-amazon.com/images/M/MV5B0TI5Nzc0MTYzBkMS00NjkkLTNmM2UtNjM2ODgxN2M5NjNkXkEyXkFqcGdeQXVyNjQ2MjQ5NzM@._V1_.jpg| Drama, War| Paths of Glory
71379| 8.2| 1925| 4320| https://images-na.ssl-images-amazon.com/images/M/MV5BMzYzMDQyNzA4NV5BM15BanBnXkFtZTYwNDU5NDU5._V1_.jpg| Adventure, Comedy, Drama, Family| The Gold Rush
700599| 8.3| 2009| 5760| https://images-na.ssl-images-amazon.com/images/M/MV5BMTk3NDE2NzI4NF5BM15BanBnXkFtZTgwNzE1MzEyMTE@._V1_.jpg| Animation, Adventure, Comedy, Family| Up
27632| 8.3| 1928| 6840| https://images-na.ssl-images-amazon.com/images/M/MV5BMjQxMjk5ODUxM15BM15BanBnXkFtZTgwNDI2MjUwMzE@._V1_.jpg| Biography, Drama, History| The Passion of Joan of Arc
228498| 8.4| 1959| 8160| https://images-na.ssl-images-amazon.com/images/M/MV5BMjQwMTQ0Mzg4N15BM15BanBnXkFtZTgwNjc4ODU4MzE@._V1_.jpg| Action, Adventure, Crime, Mystery, Thriller| North by Northwest
105342| 8.2| 1954| 6480| https://images-na.ssl-images-amazon.com/images/M/MV5BMTcwMTU3MjI1OV5BM15BanBnXkFtZTgwNjc4ODU4MzE@._V1_.jpg| Crime, Drama, Thriller| On the Waterfront
115523| 8.1| 1993| 7980| https://images-na.ssl-images-amazon.com/images/M/MV5BMTcwMTU3MjI1OV5BM15BanBnXkFtZTgwNjc4ODU4MzE@._V1_.jpg| Biography, Drama| In the Name of the Father
225098| 8.3| 1962| 7740| https://images-na.ssl-images-amazon.com/images/M/MV5BMjA4MzI1NDY2N15BM15BanBnXkFtZTcwMTcyODc5Mw@._V1_.jpg| Crime, Drama| To Kill a Mockingbird
```

```
In [10]: movies_file = open("../downloads/250.imdb")

for ...
```



```
# Votes | Rating | Year | Runtime | URL | Genres | Title
126807 | 8.5 | 1957 | 5280 | https://images-na.ssl-images-amazon.com/images/M/MV5B0TI5Nzc0OTM0YzBkMS00NjkwLTNmM2U0NjM2ODgxN2M5NjNkXkEyXkFqcGdeQXVyNjQ2MjQ5NzM0._V1_.jpg | Drama, War | Paths of Glory
71379 | 8.2 | 1925 | 4320 | https://images-na.ssl-images-amazon.com/images/M/MV5BMzYzMDQyNzA4NV5BM15BanBnXkFtZTYwNDU5NDU5._V1_.jpg | Adventure, Comedy, Drama, Family | The Gold Rush
700599 | 8.3 | 2009 | 5760 | https://images-na.ssl-images-amazon.com/images/M/MV5BMTk3NDE2NzI4NF5BM15BanBnXkFtZTgwNzE1MzEyMTE0._V1_.jpg | Animation, Adventure, Comedy, Family | Up
27632 | 8.3 | 1928 | 6840 | https://images-na.ssl-images-amazon.com/images/M/MV5BMjQxMjk5ODUxM15BM15BanBnXkFtZTgwNDI2MjUwMzE0._V1_.jpg | Biography, Drama, History | The Passion of Joan of Arc
228498 | 8.4 | 1959 | 8160 | https://images-na.ssl-images-amazon.com/images/M/MV5BMjQwMTQ0MzgwN15BM15BanBnXkFtZTgwNjc4ODc4MzE0._V1_.jpg | Action, Adventure, Crime, Mystery, Thriller | North by Northwest
105342 | 8.2 | 1954 | 6480 | https://images-na.ssl-images-amazon.com/images/M/MV5BMTcwMTU3MjE1OV5BM15BanBnXkFtZTgwNjE2OTI3MjE0._V1_.jpg | Crime, Drama, Thriller | On the Waterfront
115523 | 8.1 | 1993 | 7980 | https://images-na.ssl-images-amazon.com/images/M/MV5BMTcwNjMyMzI0OV5BM15BanBnXkFtZTgwMDU4NjgzMTE0._V1_.jpg | Biography, Drama | In the Name of the Father
225098 | 8.3 | 1962 | 7740 | https://images-na.ssl-images-amazon.com/images/M/MV5BMjA4MzI1NDY2N15BM15BanBnXkFtZTcwMTcyODc5Mw00._V1_.jpg | Crime, Drama | To Kill a Mockingbird
```

```
In [ ]: movies_file = open("../downloads/250.imdb")
```

```
for line in movies_file:
    if not line[0] == "#":
        ...
```

```
# Votes | Rating | Year | Runtime | URL | Genres | Title
126807 | 8.5 | 1957 | 5280 | https://images-na.ssl-images-amazon.com/images/M/MV5B0TI5Nzc0OTMxYzBkMS00NjkwLTNmM2U0NjM2ODgxN2M5NjNkXkEyXkFqcGdeQXVyNjQ2MjQ5NzM@._V1_.jpg | Drama, War | Paths of Glory
71379 | 8.2 | 1925 | 4320 | https://images-na.ssl-images-amazon.com/images/M/MV5BMzYzMDQyNzA4NV5BMl5BanBnXkFtZTYwNDU5NDU5._V1_.jpg | Adventure, Comedy, Drama, Family | The Gold Rush
700599 | 8.3 | 2009 | 5760 | https://images-na.ssl-images-amazon.com/images/M/MV5BMTk3NDk2NzI4NF5BMl5BanBnXkFtZTgwNzE1MzEyMTE@._V1_.jpg | Animation, Adventure, Comedy, Family | Up
27632 | 8.3 | 1928 | 6840 | https://images-na.ssl-images-amazon.com/images/M/MV5BMjQxMjk5ODUxMl5BMl5BanBnXkFtZTgwNDI2MjUwMzE@._V1_.jpg | Biography, Drama, History | The Passion of Joan of Arc
228498 | 8.4 | 1959 | 8160 | https://images-na.ssl-images-amazon.com/images/M/MV5BMjQwMTQ0MzgwNl5BMl5BanBnXkFtZTgwNjc4ODk4MzE@._V1_.jpg | Action, Adventure, Crime, Mystery, Thriller | North by Northwest
105342 | 8.2 | 1954 | 6480 | https://images-na.ssl-images-amazon.com/images/M/MV5BMTcwMTU3MjE1OV5BMl5BanBnXkFtZTgwNjE2OTI3MjE@._V1_.jpg | Crime, Drama, Thriller | On the Waterfront
115523 | 8.1 | 1993 | 7980 | https://images-na.ssl-images-amazon.com/images/M/MV5BMTcwMTU3MjE1OV5BMl5BanBnXkFtZTgwMDU4NjgzMTE@._V1_.jpg | Biography, Drama | In the Name of the Father
225098 | 8.3 | 1962 | 7740 | https://images-na.ssl-images-amazon.com/images/M/MV5BMjA4MzI1NDY2Nl5BMl5BanBnXkFtZTcwMTcyODc5Mw@._V1_.jpg | Crime, Drama | To Kill a Mockingbird
```

```
In [ ]: movies_file = open("../downloads/250.imdb")
```

```
for line in movies_file:
    if not line[0] == "#":
        line_split = line.split("|")
        ...
```

```
# Votes | Rating | Year | Runtime | URL | Genres | Title
126807 | 8.5 | 1957 | 5280 | https://images-na.ssl-images-amazon.com/images/M/MV5B0TI5Nzc0OTMtYzBkMS00NjkkLTNmM2U0NjM2ODgxN2M5NjNkXkEyXkFqcGdeQXVyNjQ2MjQ5NzM@._V1_.jpg | Drama, War | Paths of Glory
71379 | 8.2 | 1925 | 4320 | https://images-na.ssl-images-amazon.com/images/M/MV5BMzYzMDQyNzA4NV5BM15BanBnXkFtZTYwNDU5NDU5._V1_.jpg | Adventure, Comedy, Drama, Family | The Gold Rush
700599 | 8.3 | 2009 | 5760 | https://images-na.ssl-images-amazon.com/images/M/MV5BMTk3NDk2NzI4NF5BM15BanBnXkFtZTgwNzE1MzE5MTE@._V1_.jpg | Animation, Adventure, Comedy, Family | Up
27632 | 8.3 | 1928 | 6840 | https://images-na.ssl-images-amazon.com/images/M/MV5BMjQxMjk5ODUxM15BM15BanBnXkFtZTgwNDI2MjUwMzE@._V1_.jpg | Biography, Drama, History | The Passion of Joan of Arc
228498 | 8.4 | 1959 | 8160 | https://images-na.ssl-images-amazon.com/images/M/MV5BMjQwMTQ0MzgwN15BM15BanBnXkFtZTgwNjc4ODk0MzE@._V1_.jpg | Action, Adventure, Crime, Mystery, Thriller | North by Northwest
105342 | 8.2 | 1954 | 6480 | https://images-na.ssl-images-amazon.com/images/M/MV5BMTcwMTU3MjI1OV5BM15BanBnXkFtZTgwNjE2OTI3MjE@._V1_.jpg | Crime, Drama, Thriller | On the Waterfront
115523 | 8.1 | 1993 | 7980 | https://images-na.ssl-images-amazon.com/images/M/MV5BMTcwMTU3MjI1OV5BM15BanBnXkFtZTgwMDU4NjgzMTE@._V1_.jpg | Biography, Drama | In the Name of the Father
225098 | 8.3 | 1962 | 7740 | https://images-na.ssl-images-amazon.com/images/M/MV5BMjA4MzI1NDY2N15BM15BanBnXkFtZTcwMTcyODc5Mw@._V1_.jpg | Crime, Drama | To Kill a Mockingbird
```

```
In [ ]: movies_file = open("../downloads/250.imdb")
```

```
for line in movies_file:
    if not line[0] == "#":
        line_split = line.split("|")
        genres = line_split[5]
        if "Adventure" in genres:
            ...
```

```
# Votes | Rating | Year | Runtime | URL | Genres | Title
126807 | 8.5 | 1957 | 5280 | https://images-na.ssl-images-amazon.com/images/M/MV5B0TI5Nzc0OTMxYzBkMS00NjYxLThmM2U0NjM2ODgxN2M5NjNkXkEyXkFqcGdeQXVyNjQ2MjQ5NzM@._V1_.jpg | Drama, War | Paths of Glory
71379 | 8.2 | 1925 | 4320 | https://images-na.ssl-images-amazon.com/images/M/MV5BMzYzMDQyNzA4NV5BM15BanBnXkFtZTYwNDU5NDU5._V1_.jpg | Adventure, Comedy, Drama, Family | The Gold Rush
700599 | 8.3 | 2009 | 5760 | https://images-na.ssl-images-amazon.com/images/M/MV5BMTk3NDE2NzI4NF5BM15BanBnXkFtZTgwNzE1MzEyMTE@._V1_.jpg | Animation, Adventure, Comedy, Family | Up
27632 | 8.3 | 1928 | 6840 | https://images-na.ssl-images-amazon.com/images/M/MV5BMjQxMjk5ODUxM15BM15BanBnXkFtZTgwNDI2MjUwMzE@._V1_.jpg | Biography, Drama, History | The Passion of Joan of Arc
228498 | 8.4 | 1959 | 8160 | https://images-na.ssl-images-amazon.com/images/M/MV5BMjQwMTQ0MzgwN15BM15BanBnXkFtZTgwNjc4ODc4MzE@._V1_.jpg | Action, Adventure, Crime, Mystery, Thriller | North by Northwest
105342 | 8.2 | 1954 | 6480 | https://images-na.ssl-images-amazon.com/images/M/MV5BMTcwMTU3MjI1OV5BM15BanBnXkFtZTgwNjE2OTI3MjE@._V1_.jpg | Crime, Drama, Thriller | On the Waterfront
115523 | 8.1 | 1993 | 7980 | https://images-na.ssl-images-amazon.com/images/M/MV5BMTcwMTU3MjI1OV5BM15BanBnXkFtZTgwMDU4NjZMTE@._V1_.jpg | Biography, Drama | In the Name of the Father
225098 | 8.3 | 1962 | 7740 | https://images-na.ssl-images-amazon.com/images/M/MV5BMjA4MzI1NDY2N15BM15BanBnXkFtZTcwMTcyODc5Mw@._V1_.jpg | Crime, Drama | To Kill a Mockingbird
```

```
In [ ]: movies_file = open("../downloads/250.imdb")
```

```
for line in movies_file:
    if not line[0] == "#":
        line_split = line.split("|")
        genres = line_split[5]
        if "Adventure" in genres:
            rating = float(line_split[1])
            if rating > 8.5:
                ...
```

```
In [ ]: movies_file = open("../downloads/250.imdb")

for line in movies_file:
    if not line[0] == "#":
        line_split = line.split("|")
        genres = line_split[5]
        if "Adventure" in genres:
            rating = float(line_split[1])
            if rating > 8.5:
                timing_seconds = int(line_split[3])
                title = line_split[6]
                ...
```