In [39]:

```
import numpy as np #linear algebra
import pandas as pd #data processing
# step 1 get top line
from bs4 import BeautifulSoup as bs
import bs4
import requests as rq # get url
import re
from datetime import date, time, datetime

pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.expand_frame_repr', False)
pd.set_option('max_colwidth', None)

```

In [40]:

```
the_getString = 'https://www.the-numbers.com/bankability'
r=rq.get(the_getString)
# 'html.parser'
# p=bs(r.text,'lxml')
p=bs(r.text,'html.parser')
# p.prettifyl
# print(p)
jobtitleslist = []
TheNamestitleslist = []
TheURLlist = []
TheAmountlist = []
TheBornOnlist = []

# allCol2Divs=p.find_all('div', id="col2outer")
# allCol2Divs=p.find_all('div', id="col2side")
allCol2Divs=p.find_all(True, {'id':['col2mid', 'col2side']})

for allCol2Div in allCol2Divs:
#     print(allCol2Div)
    if allCol2Div.find(style="font-size:200%;"):
        bigtext = allCol2Div.find_all(style="font-size:200%;")
#         print(f'BIGTEXT = {bigtext}')
        for anchor in bigtext:
            if anchor.name =="span":
                thename = anchor.get_text()
                TheNamestitleslist.append(thename)
#                 print(f'the Name = {thename}')
                thehref = anchor.find("a")
                theURL = thehref.get("href")
                TheURLlist.append(theURL)
#                 print (f' THe href =  {theURL}')
            if anchor.name =="div":
                theAmount = anchor.get_text()
                TheAmountlist.append(theAmount)
#                 print (f' THe amount =  {theAmount}')

    if allCol2Div.find(itemprop="jobTitle"):
        jobTitles = allCol2Div.find(itemprop="jobTitle")
        for jobTitle in jobTitles:
#             the_jobTitle = jobTitle.text()
            jobtitleslist.append(jobTitle)
#         print(f'Job Title {jobTitle.get_text()}')
#         print(jobTitle)

    if allCol2Div.find_all("a", href=re.compile("on-this-day")):
        bornOns = allCol2Div.find("a", href=re.compile("on-this-day"))
        for bornon in bornOns:
            #print(bornon)
            TheBornOnlist.append(str(bornon))
#         for bornon in bornOns:
#             bornonDate = bornOn
#             print(bornonDate)
#         jobTitles = allCol2Div.find(itemprop="jobTitle")
#         for jobTitle in jobTitles:
# #             the_jobTitle = jobTitle.text()
#             jobtitleslist.append(jobTitle)
```

```
# #         print(f'Job Title {jobTitle.get_text()}')
# #         print(jobTitle)



dfjobTitles = pd.DataFrame(jobtitleslist,columns=["PrimaryTitle"])
# print(dfjobTitles)
dfNames = pd.DataFrame(TheNamestitleslist,columns=["Name"])
# print(dfNames)
dfURL = pd.DataFrame(TheURLlist,columns=["URL"])
# print(dfURL)
dfAmount = pd.DataFrame(TheAmountlist,columns=["Amount"])
# print(dfAmount)

dfjBornOn = pd.DataFrame(TheBornOnlist,columns=["BornOn"])
# print(dfjobTitles)

DFBankaability = pd.concat([dfjobTitles, dfNames, dfURL,dfAmount, dfjBornOn]

DFBankaability["BornOnDateType"] = pd.to_datetime(DFBankaability['BornOn'])
DFBankaability["NameBorn"] = DFBankaability.apply(lambda row: (row.Name) + (
```

In [41]:
```
DFBankaability.to_excel("DFBankaability.xlsx")
```

In [33]:
```

```

In [ ]:
```

```

In [35]:
```
DFBankaability.head()
```

Out[35]:

	PrimaryTitle	Name	URL	Amount	BornOn	BornOnDateType	NameBorn
0	Leading Actor	Tom Cruise	/person/540401-Tom-Cruise	$24,142,354	July 3, 1962	1962-07-03	Tom Cruise1962
1	Leading Actor	Robert Downey, Jr.	/person/41500401-Robert-Downey-Jr	$18,374,416	April 4, 1965	1965-04-04	Robert Downey Jr.1965
2	Leading Actor	Will Smith	/person/770401-Will-Smith	$17,267,666	September 25, 1968	1968-09-25	Will Smith1968
3	Producer	Kathleen Kennedy	/person/189790401-Kathleen-Kennedy	$16,885,391	June 5, 1953	1953-06-05	Kathleen Kennedy1953
4	Supporting Actor	Jon Favreau	/person/46750401-Jon-Favreau	$14,891,341	October 19, 1966	1966-10-19	Jon Favreau1966

In [ ]:
```
row.rd_string_IMDB_step3.strftime('%b %d %Y').strip()
 row.rd_string_IMDB_step3.strftime('%b %d %Y').strip()
```