Sprint 10: Tasca M10 T01

Exercici 1

Realitza web scraping de dues de les tres pàgines web proposades utilitzant BeautifulSoup primer i Selenium després.

- http://quotes.toscrape.com
- https://www.bolsamadrid.es
- www.wikipedia.es (fes alguna cerca primer i escrapeja algun contingut)

1.1 Scraping con BeautifulSoup de http://quotes.toscrape.com/page/2/

In [77]:
```
# Importar módulos
import requests
import csv
from bs4 import BeautifulSoup

# Dirección de la página web
url = "http://quotes.toscrape.com/page/2/" #http://quotes.toscrape.com/page/2/ (hemos comprobado que funciona en la segunda pa
response = requests.get(url) # Ejecutar GET-Request

# Analizar sintácticamente el archivo HTML de BeautifulSoup del texto fuente
html = BeautifulSoup(response.text, 'html.parser')

# Extraer todas las citas y autores del archivo HTML
quotes_html = html.find_all('span', class_="text")
authors_html = html.find_all('small', class_="author")

# Crear una lista de las citas
quotes = list()
for quote in quotes_html:
    quotes.append(quote.text)

# Crear una lista de los autores
authors = list()
for author in authors_html:
    authors.append(author.text)
# verificar los resultados
for t in zip(quotes, authors):
    print(t)
    print("")

# Guardar las citas y los autores en un archivo CSV en el directorio actual
with open('./citas.csv', 'w') as csv_file:
    csv_writer = csv.writer(csv_file, dialect='excel')
    csv_writer.writerows(zip(quotes, authors))
```

```
("“This life is what you make it. No matter what, you're going to mess up sometimes, it's a universal truth. But the good par
t is you get to decide how you're going to mess it up. Girls will be your friends - they'll act like it anyway. But just reme
mber, some come, some go. The ones that stay with you through everything - they're your true best friends. Don't let go of th
em. Also remember, sisters make the best friends in the world. As for lovers, well, they'll come and go too. And baby, I hate
to say it, most of them - actually pretty much all of them are going to break your heart, but you can't give up because if yo
u give up, you'll never find your soulmate. You'll never find that half who makes you whole and that goes for everything. Jus
t because you fail once, doesn't mean you're gonna fail at everything. Keep trying, hold on, and always, always, always belie
ve in yourself, because if you don't, then who will, sweetie? So keep your head high, keep your chin up, and most importantl
y, keep smiling, because life's a beautiful thing and there's so much to smile about.”", 'Marilyn Monroe')

('“It takes a great deal of bravery to stand up to our enemies, but just as much to stand up to our friends.”', 'J.K. Rowlin
g')

("“If you can't explain it to a six year old, you don't understand it yourself.”", 'Albert Einstein')

("“You may not be her first, her last, or her only. She loved before she may love again. But if she loves you now, what else
matters? She's not perfect—you aren't either, and the two of you may never be perfect together but if she can make you laugh,
cause you to think twice, and admit to being human and making mistakes, hold onto her and give her the most you can. She may
not be thinking about you every second of the day, but she will give you a part of her that she knows you can break—her hear
t. So don't hurt her, don't change her, don't analyze and don't expect more than she can give. Smile when she makes you happ
y, let her know when she makes you mad, and miss her when she's not there.”", 'Bob Marley')

('“I like nonsense, it wakes up the brain cells. Fantasy is a necessary ingredient in living.”', 'Dr. Seuss')

('“I may not have gone where I intended to go, but I think I have ended up where I needed to be.”', 'Douglas Adams')

("“The opposite of love is not hate, it's indifference. The opposite of art is not ugliness, it's indifference. The opposite
of faith is not heresy, it's indifference. And the opposite of life is not death, it's indifference.”", 'Elie Wiesel')

('“It is not a lack of love, but a lack of friendship that makes unhappy marriages.”', 'Friedrich Nietzsche')

('“Good friends, good books, and a sleepy conscience: this is the ideal life.”', 'Mark Twain')

('“Life is what happens to us while we are making other plans.”', 'Allen Saunders')
```

1.2 Scraping con BeautifulSoup de https://www.bolsamadrid.es/esp/aspx/Mercados/Precios.aspx?indice=ESI100000000

In [78]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd

#Leer url
url = "https://www.bolsamadrid.es/esp/aspx/Mercados/Precios.aspx?indice=ESI100000000"
page = requests.get(url)
soup = BeautifulSoup(page.content, "html.parser")

# busca la tabla de datos de cotizaciones del IBEX35 que queremos obtener
table = soup.find("table", id = "ctl00_Contenido_tblAcciones")
```

In [79]:
```
#Extrae el contenido
contenido = table.find_all("td")

content = list()

for i in contenido:
    content.append(i.text)

#nombre de las columnas
columns = list(("Name", "Last", "% Diff", "Max", "Min", "Volum", "Cash", "Date", "Time"))

#crea un dictionario
dic = {}

# la tabla está compuesta por 9 columnas por tanto lee la lista y extrae casa 9 elementos
for i in range(9):
    dic[columns[i]] = content[i::9]
```

In [80]:
```
#creamos un dataframe a partir del dictionario
df_cotizaciones = pd.DataFrame(data = dic)
df_cotizaciones
```

Out[80]:

	Name	Last	% Diff	Max	Min	Volum	Cash	Date	Time
0	ACCIONA	206,0000	-1,06	210,4000	205,6000	69.130	14.301,93	26/08/2022	Cierre
1	ACCIONA ENER	43,5000	0,05	44,2000	43,4000	213.361	9.340,67	26/08/2022	Cierre
2	ACERINOX	9,1640	-1,48	9,4140	9,1380	642.377	5.945,99	26/08/2022	Cierre
3	ACS	22,4200	-1,19	23,0000	22,3500	493.316	11.122,03	26/08/2022	Cierre
4	AENA	124,4000	-0,80	127,0000	124,2500	73.558	9.213,77	26/08/2022	Cierre
5	AMADEUS	54,8000	-1,76	56,1800	54,7800	263.180	14.497,26	26/08/2022	Cierre
6	ARCELORMIT.	23,5300	-1,01	24,3000	23,4850	276.827	6.633,81	26/08/2022	Cierre
7	B.SANTANDER	2,4075	-1,55	2,4830	2,3960	25.583.772	62.055,73	26/08/2022	Cierre
8	BA.SABADELL	0,6446	1,00	0,6570	0,6394	23.829.984	15.461,27	26/08/2022	Cierre
9	BANKINTER	4,8270	-0,27	4,9240	4,8130	1.238.485	6.006,38	26/08/2022	Cierre
10	BBVA	4,4530	-1,51	4,5865	4,4305	10.322.165	46.353,18	26/08/2022	Cierre
11	CAIXABANK	2,8800	-0,86	2,9420	2,8610	13.254.244	38.482,11	26/08/2022	Cierre
12	CELLNEX	39,9200	-3,50	41,6100	39,7000	928.605	37.231,32	26/08/2022	Cierre
13	ENAGAS	19,3050	0,10	19,4400	19,2000	637.267	12.305,23	26/08/2022	Cierre
14	ENDESA	17,8650	-1,35	18,2000	17,8050	668.732	11.987,45	26/08/2022	Cierre
15	FERROVIAL	25,8700	-0,61	26,3300	25,7400	427.746	11.106,76	26/08/2022	Cierre
16	FLUIDRA	15,9700	-3,15	16,7900	15,9700	664.196	10.831,54	26/08/2022	Cierre
17	GRIFOLS CL.A	12,4200	-2,74	12,9500	12,3500	1.242.748	15.522,41	26/08/2022	Cierre
18	IAG	1,2500	-3,06	1,3040	1,2500	8.448.797	10.688,98	26/08/2022	Cierre
19	IBERDROLA	10,9250	-0,73	11,1500	10,8650	9.477.435	103.755,62	26/08/2022	Cierre
20	INDITEX	22,2700	-3,76	23,2500	22,1500	2.755.260	62.094,37	26/08/2022	Cierre
21	INDRA A	8,0900	-2,18	8,3500	8,0900	370.334	3.035,60	26/08/2022	Cierre
22	INM.COLONIAL	5,9500	-2,14	6,1700	5,9500	687.556	4.146,92	26/08/2022	Cierre
23	MAPFRE	1,6260	-0,85	1,6470	1,6260	1.445.419	2.358,39	26/08/2022	Cierre
24	MELIA HOTELS	6,0200	-4,06	6,2900	6,0200	480.664	2.957,20	26/08/2022	Cierre
25	MERLIN	9,3500	-1,68	9,5750	9,3100	494.146	4.649,18	26/08/2022	Cierre
26	NATURGY	29,5800	-0,50	29,8900	29,4900	272.670	8.089,26	26/08/2022	Cierre
27	PHARMA MAR	59,0200	-3,91	61,3800	58,8800	66.187	3.971,02	26/08/2022	Cierre
28	R.E.C.	19,4200	-1,07	19,6750	19,2950	561.392	10.923,81	26/08/2022	Cierre
29	REPSOL	13,4300	0,19	13,5450	13,3900	3.721.033	50.035,03	26/08/2022	Cierre
30	ROVI	49,9800	-2,67	51,6500	49,8400	73.167	3.692,16	26/08/2022	Cierre
31	SACYR	2,2240	-2,03	2,2880	2,2080	988.312	2.220,23	26/08/2022	Cierre
32	SIEMENS GAME	17,9300	-0,06	17,9650	17,9100	550.830	9.879,47	26/08/2022	Cierre
33	SOLARIA	23,3900	-3,78	24,3300	23,2600	451.965	10.727,13	26/08/2022	Cierre
34	TELEFONICA	4,1440	-0,84	4,2070	4,1110	12.219.616	50.687,29	26/08/2022	Cierre

In [81]:
```
df_cotizaciones.to_csv("cotizaciones_bt_BM.csv", index = False) # guardamos las cotizaciones en un fichero.csv
```

1.3 Scraping con Selenimun de http://quotes.toscrape.com/page/2/

In [82]:
```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.by import By
from selenium.webdriver.support import expected_conditions as EC

options = Options()
options.add_argument("start-maximized")
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()), options=options)
driver.get("http://quotes.toscrape.com/page/2/")

all_boxes = WebDriverWait(driver, 20).until(EC.visibility_of_all_elements_located((By.XPATH, "//div[@class='quote']/span[@clas
for each in all_boxes:
    print(each.text)
    print("")
```

```
“This life is what you make it. No matter what, you're going to mess up sometimes, it's a universal truth. But the good part
is you get to decide how you're going to mess it up. Girls will be your friends - they'll act like it anyway. But just rememb
er, some come, some go. The ones that stay with you through everything - they're your true best friends. Don't let go of the
m. Also remember, sisters make the best friends in the world. As for lovers, well, they'll come and go too. And baby, I hate
to say it, most of them - actually pretty much all of them are going to break your heart, but you can't give up because if yo
u give up, you'll never find your soulmate. You'll never find that half who makes you whole and that goes for everything. Jus
t because you fail once, doesn't mean you're gonna fail at everything. Keep trying, hold on, and always, always, always belie
ve in yourself, because if you don't, then who will, sweetie? So keep your head high, keep your chin up, and most importantl
y, keep smiling, because life's a beautiful thing and there's so much to smile about.”

“It takes a great deal of bravery to stand up to our enemies, but just as much to stand up to our friends.”

“If you can't explain it to a six year old, you don't understand it yourself.”

“You may not be her first, her last, or her only. She loved before she may love again. But if she loves you now, what else ma
tters? She's not perfect—you aren't either, and the two of you may never be perfect together but if she can make you laugh, c
ause you to think twice, and admit to being human and making mistakes, hold onto her and give her the most you can. She may n
ot be thinking about you every second of the day, but she will give you a part of her that she knows you can break—her heart.
So don't hurt her, don't change her, don't analyze and don't expect more than she can give. Smile when she makes you happy, l
et her know when she makes you mad, and miss her when she's not there.”

“I like nonsense, it wakes up the brain cells. Fantasy is a necessary ingredient in living.”

“I may not have gone where I intended to go, but I think I have ended up where I needed to be.”

“The opposite of love is not hate, it's indifference. The opposite of art is not ugliness, it's indifference. The opposite of
faith is not heresy, it's indifference. And the opposite of life is not death, it's indifference.”

“It is not a lack of love, but a lack of friendship that makes unhappy marriages.”

“Good friends, good books, and a sleepy conscience: this is the ideal life.”

“Life is what happens to us while we are making other plans.”
```

In [83]:
```
driver.close()
```

1.4 Scraping con Selenium https://www.bolsamadrid.es/esp/aspx/Mercados/Precios.aspx?indice=ESI100000000

In [84]:
```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.by import By

options = Options()
options.add_argument("start-maximized")
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()), options=options)
driver.get("https://www.bolsamadrid.es/esp/aspx/Mercados/Precios.aspx?indice=ESI100000000")
```

In [85]:
```
table = driver.find_element(By.ID, "ctl00_Contenido_tblAcciones")

contenido = []

for i in table.find_elements(By.TAG_NAME, "td"):
    contenido.append(i.text)
```

In [86]:
```
#crea un diccionario
dicc = {}

for i in range(9):
    dicc[columns[i]] = contenido[i::9]
```

In [87]:
```
#crea un dataframe desde el diccionario
df2_cotizaciones = pd.DataFrame(data = dicc)
df2_cotizaciones
```

Out[87]:

	Name	Last	% Diff	Max	Min	Volum	Cash	Date	Time
0	ACCIONA	206,0000	-1,06	210,4000	205,6000	69.130	14.301,93	26/08/2022	Cierre
1	ACCIONA ENER	43,5000	0,05	44,2000	43,4000	213.361	9.340,67	26/08/2022	Cierre
2	ACERINOX	9,1640	-1,48	9,4140	9,1380	642.377	5.945,99	26/08/2022	Cierre
3	ACS	22,4200	-1,19	23,0000	22,3500	493.316	11.122,03	26/08/2022	Cierre
4	AENA	124,4000	-0,80	127,0000	124,2500	73.558	9.213,77	26/08/2022	Cierre
5	AMADEUS	54,8000	-1,76	56,1800	54,7800	263.180	14.497,26	26/08/2022	Cierre
6	ARCELORMIT.	23,5300	-1,01	24,3000	23,4850	276.827	6.633,81	26/08/2022	Cierre
7	B.SANTANDER	2,4075	-1,55	2,4830	2,3960	25.583.772	62.055,73	26/08/2022	Cierre
8	BA.SABADELL	0,6446	1,00	0,6570	0,6394	23.829.984	15.461,27	26/08/2022	Cierre
9	BANKINTER	4,8270	-0,27	4,9240	4,8130	1.238.485	6.006,38	26/08/2022	Cierre
10	BBVA	4,4530	-1,51	4,5865	4,4305	10.322.165	46.353,18	26/08/2022	Cierre
11	CAIXABANK	2,8800	-0,86	2,9420	2,8610	13.254.244	38.482,11	26/08/2022	Cierre
12	CELLNEX	39,9200	-3,50	41,6100	39,7000	928.605	37.231,32	26/08/2022	Cierre
13	ENAGAS	19,3050	0,10	19,4400	19,2000	637.267	12.305,23	26/08/2022	Cierre
14	ENDESA	17,8650	-1,35	18,2000	17,8050	668.732	11.987,45	26/08/2022	Cierre
15	FERROVIAL	25,8700	-0,61	26,3300	25,7400	427.746	11.106,76	26/08/2022	Cierre
16	FLUIDRA	15,9700	-3,15	16,7900	15,9700	664.196	10.831,54	26/08/2022	Cierre
17	GRIFOLS CL.A	12,4200	-2,74	12,9500	12,3500	1.242.748	15.522,41	26/08/2022	Cierre
18	IAG	1,2500	-3,06	1,3040	1,2500	8.448.797	10.688,98	26/08/2022	Cierre
19	IBERDROLA	10,9250	-0,73	11,1500	10,8650	9.477.435	103.755,62	26/08/2022	Cierre
20	INDITEX	22,2700	-3,76	23,2500	22,1500	2.755.260	62.094,37	26/08/2022	Cierre
21	INDRA A	8,0900	-2,18	8,3500	8,0900	370.334	3.035,60	26/08/2022	Cierre
22	INM.COLONIAL	5,9500	-2,14	6,1700	5,9500	687.556	4.146,92	26/08/2022	Cierre
23	MAPFRE	1,6260	-0,85	1,6470	1,6260	1.445.419	2.358,39	26/08/2022	Cierre
24	MELIA HOTELS	6,0200	-4,06	6,2900	6,0200	480.664	2.957,20	26/08/2022	Cierre
25	MERLIN	9,3500	-1,68	9,5750	9,3100	494.146	4.649,18	26/08/2022	Cierre
26	NATURGY	29,5800	-0,50	29,8900	29,4900	272.670	8.089,26	26/08/2022	Cierre
27	PHARMA MAR	59,0200	-3,91	61,3800	58,8800	66.187	3.971,02	26/08/2022	Cierre
28	R.E.C.	19,4200	-1,07	19,6750	19,2950	561.392	10.923,81	26/08/2022	Cierre
29	REPSOL	13,4300	0,19	13,5450	13,3900	3.721.033	50.035,03	26/08/2022	Cierre
30	ROVI	49,9800	-2,67	51,6500	49,8400	73.167	3.692,16	26/08/2022	Cierre
31	SACYR	2,2240	-2,03	2,2880	2,2080	988.312	2.220,23	26/08/2022	Cierre
32	SIEMENS GAME	17,9300	-0,06	17,9650	17,9100	550.830	9.879,47	26/08/2022	Cierre
33	SOLARIA	23,3900	-3,78	24,3300	23,2600	451.965	10.727,13	26/08/2022	Cierre
34	TELEFONICA	4,1440	-0,84	4,2070	4,1110	12.219.616	50.687,29	26/08/2022	Cierre

In [88]:
```
df2_cotizaciones.to_csv("cotizaciones_se_BM.csv", index = False)
driver.close()
```

Exercici 2

Documenta en un Word el teu conjunt de dades generat amb la informació que tenen els diferents arxius de Kaggle.

In [89]:
```
df2_cotizaciones.head()
```

Out[89]:

	Name	Last	% Diff	Max	Min	Volum	Cash	Date	Time
0	ACCIONA	206,0000	-1,06	210,4000	205,6000	69.130	14.301,93	26/08/2022	Cierre
1	ACCIONA ENER	43,5000	0,05	44,2000	43,4000	213.361	9.340,67	26/08/2022	Cierre
2	ACERINOX	9,1640	-1,48	9,4140	9,1380	642.377	5.945,99	26/08/2022	Cierre
3	ACS	22,4200	-1,19	23,0000	22,3500	493.316	11.122,03	26/08/2022	Cierre
4	AENA	124,4000	-0,80	127,0000	124,2500	73.558	9.213,77	26/08/2022	Cierre

Descripción de los diferentes campos del Data Frame:

- Name : Name of the stock index,

- Last : Index price at the time of table creation
- % Diff. : Precentual difference of the current price with respect the last quotation
- Max : Maximum price of the stock market in the current session
- Min : Minimum price of the stock market in the current session
- Volum : Number of transactions of the asset carried out until the close of the trading session
- Cash (miles €): Total amount negotiated,
- Date : Date of the session
- Time : Time session

In [91]:
```
from pandas_profiling import ProfileReport

# genera el informe
indices_profile = ProfileReport(df2_cotizaciones,
                  title = 'Information on the price, volume, of the IBEX35 the main index of the Spanish stock market (08/28/2
                  dataset = {'description' : 'This dataset contains information about the IBEX35, the main index of the Spanis
                            'url': 'https://www.bolsamadrid.es/esp/aspx/Mercados/Precios.aspx?indice=ESI100000000'},
                  variables = {'descriptions': {
                      'Name' : 'Name of the stock index',
                      'Last' : 'Index price at the time of table creation',
                      '% Dif.' : 'Precentual difference of the current price with respect the last quotation',
                      'Max' : 'Maximum price of the stock market in the current session',
                      'Min' : 'Minimum price of the stock market in the current session',
                      'Volum' : 'Number of transactions of the asset carried out until the close of the trading session',
                      'Cash (miles €)' : 'total amount negotiated',
                      'Data' : 'Date of the session',
                      'Time' : 'Time session'
                  }})
#genera la página html
indices_profile.to_file('Cotizaciones_Bolsa_Madrid.html')
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
Export report to file:   0%|          | 0/1 [00:00<?, ?it/s]
```

https://amariasm.github.io/web_scrapping/Cotizaciones_Bolsa_Madrid.html

Exercici 3

Tria una pàgina web que tu vulguis i realitza web scraping mitjançant la llibreria Selenium primer i Scrapy després.

Vamos a realizar scraping sobre diferentes fuentes de información relacionadas con el proyecto final del curso, como son la agencia tributaria (aet.com) y el portal Idealista.com

3.1 Agencia tributaria información sobre renta y viviendas con Selenium

In [59]:
```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.by import By

options = Options()
options.add_argument("start-maximized")
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()), options=options)
driver.get("https://sede.agenciatributaria.gob.es/AEAT/Contenidos_Comunes/La_Agencia_Tributaria/Estadisticas/Publicaciones/sit
```

In [60]:
```
table = driver.find_element(By.ID, "table01")

contenido = []
contenido1=[]

for i in table.find_elements(By.TAG_NAME, "td"):
    contenido.append(i.text)
for i in table.find_elements(By.TAG_NAME, "th"):
    contenido1.append(i.text)
```

In [61]:
```
columna1=contenido1[9:]
df1 = pd.DataFrame(columna1, columns = ['CCAA de residencia del declarante'])
df1
```

Out[61]:

	CCAA de residencia del declarante
0	Andalucía
1	Aragón
2	Asturias, Principado de
3	Balears,Illes
4	Canarias
5	Cantabria
6	Castilla y León
7	Castilla - La Mancha
8	Cataluña
9	Comunitat Valenciana
10	Extremadura
11	Galicia
12	Madrid, Comunidad de
13	Murcia, Región de
14	Rioja, La

In [62]:
```
columns = list(("Número de viviendas","Renta imputada","Ingresos arrendamientos","Gastos arrendamientos",
"Rendimiento neto","Rendimiento neto positivo","Rendimiento neto negativo","Rendimiento neto reducido"))
#creamos un diccionario
dicc = {}

for i in range(8):
    dicc[columns[i]] = contenido[i::8]

#creamos a dataframe desde el diccionario
df2 = pd.DataFrame(data = dicc)

df_vivienda = pd.concat([df1, df2], axis=1)
df_vivienda
```

Out[62]:

	CCAA de residencia del declarante	Número de viviendas	Renta imputada	Ingresos arrendamientos	Gastos arrendamientos	Rendimiento neto	Rendimiento neto positivo	Rendimiento neto negativo	Rendimiento neto reducido
0	Andalucía	2.938.463	734.331.487	1.674.887.787	704.654.534	970.229.544	997.043.952	26.814.408	517.882.566
1	Aragón	636.106	121.577.668	455.164.234	217.329.949	237.820.836	247.831.667	10.010.831	106.836.220
2	Asturias, Principado de	456.832	81.287.697	295.202.561	135.788.971	159.413.422	164.019.916	4.606.494	75.572.831
3	Balears,Illes	433.644	125.235.720	770.541.611	284.259.096	486.279.888	495.799.981	9.520.094	239.998.837
4	Canarias	624.450	128.147.300	780.194.736	304.419.270	475.774.304	487.433.820	11.659.516	244.779.999
5	Cantabria	240.470	48.253.007	161.426.132	80.074.338	81.350.722	87.235.155	5.884.433	37.097.405
6	Castilla y León	1.158.667	226.069.213	633.985.369	312.403.130	321.579.412	334.297.131	12.717.719	150.747.765
7	Castilla - La Mancha	832.351	183.063.511	406.911.341	190.260.729	216.650.421	225.290.672	8.640.250	104.081.894
8	Cataluña	2.835.052	663.173.772	4.310.082.788	1.812.405.216	2.497.658.893	2.567.689.325	70.030.431	1.130.885.461
9	Comunitat Valenciana	2.076.475	519.889.156	1.313.631.900	596.727.181	716.901.470	743.181.611	26.280.141	354.721.763
10	Extremadura	417.355	77.808.026	158.495.310	74.342.835	84.152.476	87.494.163	3.341.687	43.665.952
11	Galicia	1.053.026	206.858.072	772.613.129	334.781.409	437.821.600	453.501.089	15.679.489	217.596.919
12	Madrid, Comunidad de	2.958.893	830.181.093	4.296.071.833	1.782.386.894	2.513.673.500	2.573.736.194	60.062.694	1.076.077.188
13	Murcia, Región de	559.741	139.410.413	256.700.700	123.567.590	133.127.327	138.596.685	5.469.358	63.205.990
14	Rioja, La	152.608	31.435.283	83.370.721	49.302.987	34.067.735	38.268.676	4.200.940	14.814.021

In [63]:
```
df_vivienda.to_csv("Viviendas por Declarantes y por Coumunidades Autónomas.csv", index = False)
```

3.2 Idealista información vivienda en venta con Scrapy

In [64]:
```
import scrapy
from scrapy.crawler import CrawlerProcess
```

In [65]:
```
import json

# define class to save results as json file
class JsonWriterPipeline(object):

    def open_spider(self, spider):
        self.file = open('casa_result.jl', 'w')
```

```
    def close_spider(self, spider):
        self.file.close()

    def process_item(self, item, spider):
        line = json.dumps(dict(item)) + "\n"
        self.file.write(line)
        return item
```

In [66]:
```
import logging

# construct the spider to get the info we want
class CasaSpider(scrapy.Spider):
    name = "casas"
    start_urls = [
        'https://www.idealista.com/buscar/venta-viviendas/08005/?ordenado-por=precios-desc',
        'https://www.idealista.com/buscar/venta-viviendas/08005/pagina-2.htm?ordenado-por=precios-desc'
    ]
    custom_settings = {
        'LOG_LEVEL': logging.WARNING,
        'ITEM_PIPELINES': {'__main__.JsonWriterPipeline': 1}, # Used for pipeline 1
        #'FEED_FORMAT':'json',                                 # Used for pipeline 2
        #'FEED_URI': 'guitar_result.json'                      # Used for pipeline 2
         'FEEDS':{'casa_result.json':{'format':'json'}}
    }
    def parse(self, response):
        for casa in response.css('div.item-info-container'):
            yield {
                 'Link': casa.css('a.item-link::text').extract_first(),
                 'Price': casa.css('span.item-price.h2-simulated::text').extract_first(),
                 'Parking': casa.css('span.item-parking::text').extract_first(),
                 'Description_1': casa.css('span.item-detail:nth-child(1)::text').extract_first(),
                 'Description_2': casa.css('span.item-detail:nth-child(2)::text').extract_first(),
                 'Description_3': casa.css('span.item-detail:nth-child(3)::text').extract_first(),
            }
```

In [67]:
```
import sys
if "twisted.internet.reactor" in sys.modules:
    del sys.modules["twisted.internet.reactor"]

process = CrawlerProcess({
    'LOG_LEVEL': 30,
    'USER_AGENT': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.102 Safa
})

process.crawl(CasaSpider)
process.start()
```

Out[67]:
```
<Deferred at 0x2146022b6d0>
```

In [92]:
```
import pandas as pd
# we can open the json file to a dataframe
df_casas= pd.read_json('casa_result.jl', lines=True)
df_casas.head(10)
```

Out[92]:

	Link	Price	Parking	Description_1	Description_2	Description_3
0	Piso en La Vila Olímpica del Poblenou, Barcelona	890.000	Garaje incluido	4 hab.	127 m²	Planta 7ª exterior con ascensor
1	Chalet adosado en La Vila Olímpica del Pobleno...	855.000	Garaje incluido	3 hab.	146 m²	None
2	Piso en La Vila Olímpica del Poblenou, Barcelona	825.000	None	3 hab.	112 m²	Planta 8ª exterior con ascensor
3	Piso en La Vila Olímpica del Poblenou, Barcelona	815.000	None	3 hab.	112 m²	Planta 7ª con ascensor
4	Piso en La Vila Olímpica del Poblenou, Barcelona	815.000	None	3 hab.	112 m²	Planta 8ª exterior con ascensor
5	Piso en Passatge de Saladrigas, 4, El Poblenou...	815.000	None	3 hab.	101 m²	Bajo exterior con ascensor
6	Piso en Passatge de Saladrigas, 4, El Poblenou...	807.000	None	3 hab.	112 m²	Bajo exterior con ascensor
7	Piso en calle de l'Arquitecte Sert, 31, La Vil...	799.000	Garaje	4 hab.	238 m²	Planta 1ª exterior con ascensor
8	Piso en calle de Salvador Espriu, La Vila Olím...	799.000	Garaje incluido	2 hab.	98 m²	Planta 5ª exterior con ascensor
9	Dúplex en SALVADOR ESPRIU, La Vila Olímpica de...	798.000	None	4 hab.	178 m²	Planta 2ª exterior con ascensor

In [69]:
```
df_casas.to_csv("Viviendas de Idealiata.csv", index = False)
```

3.3 Agencia Tributaria información sobre vivienda y valores catratrales con BeautifulSoap

In [70]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

In [71]:
```
#read url
url = "https://sede.agenciatributaria.gob.es/AEAT/Contenidos_Comunes/La_Agencia_Tributaria/Estadisticas/Publicaciones/sites/ir
page = requests.get(url)
soup = BeautifulSoup(page.content, "html.parser")
```

In [72]:
```
# Obtenemos la tabla por un ID específico
table = soup.find('table', attrs={'id': 'table01'})
```

In [73]:
```
#get the content
cont = table.find_all("th")
cont1= table.find_all("td")

content = list()
content1= list()

for i in cont:
    content.append(i.text)
for i in cont1:
    content1.append(i.text)
```

In [74]:
```
columna1=content[4:]
df1 = pd.DataFrame(columna1, columns = ['Localización de la vivienda'])
columna2=content1[::3]
df2 = pd.DataFrame(columna2, columns = ['Número de viviendas con valor catastral'])
columna3=content1[1::3]
df3 = pd.DataFrame(columna3, columns = ['Valor catastral'])
columna4=content1[2::3]
df4 = pd.DataFrame(columna4, columns = ['Valor catastral medio'])

df_valorCatastral = pd.concat([df1, df2,df3,df4], axis=1)
df_valorCatastral
```

Out[74]:

	Localización de la vivienda	Número de viviendas con valor catastral	Valor catastral	Valor catastral medio
0	Andalucía	3.027.330	180.646.091.513	59.672
1	Almería	272.136	16.354.522.072	60.097
2	Cádiz	416.770	25.459.994.309	61.089
3	Córdoba	289.553	14.798.931.117	51.110
4	Granada	384.382	20.357.495.848	52.962
5	Huelva	211.152	10.750.357.914	50.913
6	Jaén	262.029	12.666.129.231	48.339
7	Málaga	596.148	46.474.498.345	77.958
8	Sevilla	595.160	33.784.162.677	56.765
9	Aragón	619.327	35.174.089.732	56.794
10	Huesca	118.011	5.181.261.631	43.905
11	Teruel	85.489	3.171.328.775	37.096
12	Zaragoza	415.827	26.821.499.326	64.502
13	Asturias, Principado de	452.175	25.939.756.973	57.367
14	Balears,Illes	411.383	32.236.852.682	78.362
15	Canarias	603.152	34.147.647.921	56.615
16	Las Palmas	310.194	17.909.142.047	57.735
17	S. C. Tenerife	292.958	16.238.505.874	55.429
18	Cantabria	248.939	18.267.673.830	73.382
19	Castilla y León	1.276.208	62.294.220.110	48.812
20	Ávila	119.156	5.264.869.880	44.185
21	Burgos	180.350	10.927.254.155	60.589
22	León	237.674	10.111.484.794	42.544
23	Palencia	82.719	3.292.759.261	39.807
24	Salamanca	175.392	7.417.434.510	42.291
25	Segovia	93.246	5.769.784.842	61.877
26	Soria	61.664	2.937.493.107	47.637
27	Valladolid	227.577	12.085.256.386	53.104
28	Zamora	98.430	4.487.883.175	45.595
29	Castilla - La Mancha	928.014	52.544.953.523	56.621
30	Albacete	167.814	9.287.040.697	55.341
31	Ciudad Real	209.948	10.545.206.808	50.228
32	Cuenca	111.997	4.807.639.898	42.927
33	Guadalajara	139.470	8.804.382.318	63.127
34	Toledo	298.785	19.100.683.802	63.928
35	Cataluña	2.685.604	205.991.152.237	76.702
36	Barcelona	1.811.567	157.538.759.671	86.963
37	Girona	324.348	20.165.082.091	62.171
38	Lleida	170.372	8.615.878.146	50.571
39	Tarragona	379.317	19.671.432.329	51.860
40	Comunitat Valenciana	2.191.430	117.116.481.365	53.443
41	Alicante	789.667	39.926.208.046	50.561
42	Castellón	322.388	17.176.252.892	53.278
43	Valencia	1.079.375	60.014.020.426	55.601
44	Extremadura	453.419	18.331.300.816	40.429
45	Badajoz	262.102	11.494.225.835	43.854
46	Cáceres	191.317	6.837.074.980	35.737
47	Galicia	1.072.192	51.433.602.157	47.971
48	A Coruña	451.203	24.256.830.082	53.760
49	Lugo	149.641	5.301.445.532	35.428
50	Ourense	141.604	6.331.231.568	44.711
51	Pontevedra	329.744	15.544.094.974	47.140

	Localización de la vivienda	Número de viviendas con valor catastral	Valor catastral	Valor catastral medio
52	Madrid, Comunidad de	2.292.318	255.335.952.104	111.388
53	Murcia, Región de	565.378	32.423.371.607	57.348
54	Rioja, La	145.880	6.975.523.681	47.817

In [75]:
```
df_valorCatastral.to_csv("Valor catastral por Coumunidades y provincias.csv", index = False)
```