{

"cells": [

{

"cell\_type": "code",

"execution\_count": 23,

"id": "63cc6bee",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Requirement already satisfied: selenium in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (4.8.0)\n",

"Requirement already satisfied: certifi>=2021.10.8 in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (from selenium) (2021.10.8)\n",

"Requirement already satisfied: urllib3[socks]~=1.26 in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (from selenium) (1.26.9)\n",

"Requirement already satisfied: trio~=0.17 in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (from selenium) (0.22.0)\n",

"Requirement already satisfied: trio-websocket~=0.9 in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (from selenium) (0.9.2)\n",

"Requirement already satisfied: idna in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (from trio~=0.17->selenium) (3.3)\n",

"Requirement already satisfied: attrs>=19.2.0 in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (from trio~=0.17->selenium) (21.4.0)\n",

"Requirement already satisfied: outcome in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (from trio~=0.17->selenium) (1.2.0)\n",

"Requirement already satisfied: async-generator>=1.9 in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (from trio~=0.17->selenium) (1.10)\n",

"Requirement already satisfied: exceptiongroup>=1.0.0rc9 in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (from trio~=0.17->selenium) (1.1.0)\n",

"Requirement already satisfied: cffi>=1.14 in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (from trio~=0.17->selenium) (1.15.0)\n",

"Requirement already satisfied: sniffio in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (from trio~=0.17->selenium) (1.2.0)\n",

"Requirement already satisfied: sortedcontainers in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (from trio~=0.17->selenium) (2.4.0)\n",

"Requirement already satisfied: pycparser in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (from cffi>=1.14->trio~=0.17->selenium) (2.21)\n",

"Requirement already satisfied: wsproto>=0.14 in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (from trio-websocket~=0.9->selenium) (1.2.0)\n",

"Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (from urllib3[socks]~=1.26->selenium) (1.7.1)\n",

"Requirement already satisfied: h11<1,>=0.9.0 in c:\\users\\su2ny\\anaconda3\\lib\\site-packages (from wsproto>=0.14->trio-websocket~=0.9->selenium) (0.14.0)\n"

]

}

],

"source": [

"!pip install selenium"

]

},

{

"cell\_type": "code",

"execution\_count": 24,

"id": "b4e38142",

"metadata": {},

"outputs": [],

"source": [

"import selenium\n",

"import pandas as pd\n",

"from selenium import webdriver\n",

"import warnings"

]

},

{

"cell\_type": "code",

"execution\_count": 25,

"id": "b2369e06",

"metadata": {},

"outputs": [],

"source": [

"warnings.filterwarnings('ignore')\n",

"from selenium.common.exceptions import StaleElementReferenceException, NoSuchElementException\n",

"from selenium.webdriver.common.by import By\n",

"import time"

]

},

{

"cell\_type": "markdown",

"id": "fbb155ef",

"metadata": {},

"source": [

"Now we will download the webDriver for the Web Browser.Steps for download are-\n",

"1.check the version of your browser\n",

"2.go to the link https://chromedriver.chromium.org/downloads\n",

"3.Download the webdriver for your version of your browser."

]

},

{

"cell\_type": "code",

"execution\_count": 26,

"id": "b5182eba",

"metadata": {},

"outputs": [],

"source": [

"#Let first connect to the driver\n",

"driver=webdriver.Chrome(r\"C:\\Users\\su2ny\\Downloads\\chromedriver\_win32\\chromedriver.exe\")"

]

},

{

"cell\_type": "code",

"execution\_count": 27,

"id": "3c82712d",

"metadata": {},

"outputs": [],

"source": [

"#Opening the naukri page on automated chrome brower\n",

"driver.get('https://www.naukri.com/')"

]

},

{

"cell\_type": "markdown",

"id": "6593ffa8",

"metadata": {},

"source": [

"Now we will download the webDriver for the Web Browser.Steps for download are-\n",

"1.check the version of your browser\n",

"2.go to the link https://chromedriver.chromium.org/downloads\n",

"3.Download the webdriver for your version of your browser."

]

},

{

"cell\_type": "code",

"execution\_count": 28,

"id": "27304543",

"metadata": {},

"outputs": [],

"source": [

"# entering designation and location as required in the question-\n",

"\n",

"designation=driver.find\_element(By.CLASS\_NAME,\"suggestor-input \")\n",

"designation.send\_keys('Data Analyst')"

]

},

{

"cell\_type": "code",

"execution\_count": 29,

"id": "8a8d0dd8",

"metadata": {},

"outputs": [],

"source": [

"location=driver.find\_element(By.XPATH,\"/html/body/div[1]/div[6]/div/div/div[5]/div/div/div/input\")\n",

"location.send\_keys('Banglore')"

]

},

{

"cell\_type": "code",

"execution\_count": 30,

"id": "ab2eb864",

"metadata": {},

"outputs": [],

"source": [

"search=driver.find\_element(By.XPATH,\"/html/body/div[1]/div[6]/div/div/div[6]\")\n",

"search.click()"

]

},

{

"cell\_type": "code",

"execution\_count": 31,

"id": "f825be5c",

"metadata": {},

"outputs": [],

"source": [

"job\_title=[]\n",

"job\_location=[]\n",

"company\_name=[]\n",

"experience\_required=[]"

]

},

{

"cell\_type": "code",

"execution\_count": 32,

"id": "d68ddd81",

"metadata": {},

"outputs": [],

"source": [

"# scraping job title from the given page\n",

"title\_tags=driver.find\_elements(By.XPATH,'//a[@class=\"title ellipsis\"]')\n",

"for i in title\_tags[0:10]:\n",

" title=i.text\n",

" job\_title.append(title)\n",

" \n",

"# scraping job location from the given page\n",

"location\_tags=driver.find\_elements(By.XPATH,'//span[@class=\"ellipsis fleft locWdth\"]')\n",

"for i in location\_tags[0:10]:\n",

" location=i.text\n",

" job\_location.append(location)\n",

" \n",

"# scraping company name from the given page\n",

"company\_tags=driver.find\_elements(By.XPATH,'//a[@class=\"subTitle ellipsis fleft\"]')\n",

"for i in company\_tags[0:10]:\n",

" company=i.text\n",

" company\_name.append(company)\n",

" \n",

"#scraping job experience from the given page\n",

"experience\_tags=driver.find\_elements(By.XPATH,'//span[@class=\"ellipsis fleft expwdth\"]')\n",

"for i in experience\_tags[0:10]:\n",

" exp=i.text\n",

" experience\_required.append(exp)\n"

]

},

{

"cell\_type": "code",

"execution\_count": 33,

"id": "8a8d331c",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"0 0 0 0\n"

]

}

],

"source": [

"print(len(job\_title),len(job\_location),len(company\_name),len(experience\_required))"

]

},

{

"cell\_type": "code",

"execution\_count": 34,

"id": "94a7e418",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>title</th>\n",

" <th>location</th>\n",

" <th>company\_name</th>\n",

" <th>experience</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

"Empty DataFrame\n",

"Columns: [title, location, company\_name, experience]\n",

"Index: []"

]

},

"execution\_count": 34,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"import pandas as pd\n",

"df=pd.DataFrame({'title':job\_title,'location':job\_location,'company\_name':company\_name,'experience':experience\_required})\n",

"df"

]

},

{

"cell\_type": "code",

"execution\_count": 35,

"id": "aeef8af2",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"[]"

]

},

"execution\_count": 35,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"# to fetcg the url -\n",

"url=driver.find\_elements(By.XPATH,'//a[@class=\"title ellipsis\"]')\n",

"url[0:10]"

]

},

{

"cell\_type": "code",

"execution\_count": 36,

"id": "3fd43758",

"metadata": {},

"outputs": [],

"source": [

"# Lets provide range to print only top 10 data\n",

"for i in url[0:10]:\n",

" print(i.get\_attribute('href'))"

]

},

{

"cell\_type": "code",

"execution\_count": 37,

"id": "0aed44ae",

"metadata": {},

"outputs": [],

"source": [

"job\_titles=[]"

]

},

{

"cell\_type": "code",

"execution\_count": 38,

"id": "06f80a01",

"metadata": {},

"outputs": [],

"source": [

"start=0\n",

"end=2\n",

"for page in range(start,end):\n",

" title=driver.find\_elements(By.XPATH,\"//a[@class='title fw500 ellipsis']\")\n",

" for i in title [0:10]:\n",

" job\_title.append(i.text)\n",

" \n",

" next\_button=driver.find\_elements(By.XPATH,\"//a[@class='1lkt03']\")"

]

},

{

"cell\_type": "code",

"execution\_count": 39,

"id": "c36294ec",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"0"

]

},

"execution\_count": 39,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"len(job\_titles)"

]

},

{

"cell\_type": "code",

"execution\_count": 40,

"id": "e7399f66",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"[]"

]

},

"execution\_count": 40,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"job\_titles"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"id": "e5d3c67c",

"metadata": {},

"outputs": [],

"source": []

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3 (ipykernel)",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.9.12"

}

},

"nbformat": 4,

"nbformat\_minor": 5

}