⇒ We will try to scrap a Job advertisement site to scrap details of Job having Python as criteria.

⇒ We will use the requests library, to see the HTML code of a webpage.

```
pip install requests
```

⇒ requests library just requests information from a website. Information can be a webpage or other file. Webpage will be a html or text file simply.

⇒ Using requests library :-

```
import requests

html_response = requests.get("url")
print(html_response)
```

⇒ requests.get() method takes complete URL of a webpage, and returns the response status code of a GET request to that webpage.

⇒ To get the HTML webpage code of the requested web page, we need to use the text member attribute, of the returned object from get().

```
html_page = requests.get("<url>").text
print (html_page)
```

attribute member
contains the webpage's HTML code

Prints the HTML code of the webpage.
[the entire code shown in page-source].

⇒ The find() & find_all() method can be used on any HTML element object :-

```
jobs = soup.find_all ("div")
for job in jobs:
    company = job.find ("h3")
    print (company.text)
```

Used for soup as well as for child HTML objects

37:20

⇒ Using the above knowledge & basic python knowledge, we can filter out & extract needed piece of info from webpages.

50:00

⇒ Adding more functionalities :-

⇒ Formatting of texts fetched by removing white spaces :-

⇒ Texts fetched from elements having leading and trailing spaces, tabs, multiple line breaks.

Those leading spaces & line breaks in a string can be removed by strip() command :-

```
str = "   .   I am there !   ."
print( str.strip() )
```

⇒ If any parameter, i.e, string is passed as its param, then it truncates that string from the beginning and end of the caller object string.

53:30

⇒ To fetch the value of any attribute of an HTML tag :-

⇒ Let us take a scenario :-

While web scraping a webpage, suppose we get multiple cards regarding a topic, based on our search. Each has information, and hyperlinks for detailed info for each topic.
Now, we need to scrap the links, that contains more details.
The hyperlinks are actually stored in attributes of tags, i.e, <a> tag.

⇒ The values of attributes of a tag can be accessed in following way :-

Suppose, in the element contained in "job" below, we have and <h2> tag, and inside it, we have <a> tag, and we want to fetch the link in its href attribute. Then this can be done as:-

```
job = soup.find("div", class_="bx-card")
link = job.h2.a["href"]
print(link)
```

link will contain the link as string

tag["attribute"]

Thus each tags attribute can be thought of as dictionaries, and the keys in the dictionary.