Hands-On Activity 7.2: Webscraping using BeautifulSoup and Requests

Data Collection from Webcam

In [12]:
```
pip install opencv-python
```

```
Requirement already satisfied: opencv-python in c:\users\lenovo\anaconda3\lib\site-p
ackages (4.9.0.80)Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: numpy>=1.21.2 in c:\users\lenovo\anaconda3\lib\site-p
ackages (from opencv-python) (1.26.4)
```

In [33]:
```
import cv2
#from google.colab.patches import cv2_imshow
key = cv2.waitKey(1)
webcam = cv2.VideoCapture(0)
while True:
    try:
        check, frame = webcam.read()
        print(check)
        print(frame)
        cv2.imshow("Capturing", frame)
        key = cv2.waitKey(1)
        if key == ord('s'):
            cv2.imwrite(filename='saved_img.jpg', img=frame)
            webcam.release()
            img_new = cv2.imread('saved_img.jpg', cv2.IMREAD_GRAYSCALE)
            img_new = cv2. imshow("Captured Image", img_new)
            cv2.waitKey(1650)
            cv2.destroyAllWindows()
            print("Processing image...")
            img_ = cv2.imread('saved_img.jpg', cv2.IMREAD_ANYCOLOR)
            print("Converting RGB image to grayscale...")
            gray = cv2.cvtColor(img_, cv2.COLOR_BGR2GRAY)
            print("Resizing image to 28x28 scale...")
            img_ = cv2.resize(gray,(28,28))
            print("Resized...")
            img_resized = cv2.imwrite(filename = 'saved_img.jpg', img=img_)
            print("Image saved!")

            break
        elif key == ord('q'):
            print("Turning off camera.")
            webcam.release()
            print("Camera off.")
            print("Program ended.")
            cv2.destroyAllWindows()
            break
    except(KeyboardInterrupt):
        print("Turning off camera.")
        webcam.release()
```

```
        print("Camera off.")
        print("Program ended.")
        cv2.destroyAllWindows()
        break
```

```
True
[[[75 74 76]
  [75 75 78]
  [80 80 82]
  ...
  [58 44 46]
  [58 42 43]
  [59 43 44]]

 [[73 73 73]
  [73 73 74]
  [77 77 77]
  ...
  [53 42 44]
  [54 42 43]
  [54 42 43]]

 [[73 73 73]
  [73 73 74]
  [73 73 73]
  ...
  [52 42 43]
  [52 42 43]
  [52 42 43]]

 ...

 [[17 14 19]
  [20 16 21]
  [20 16 21]
  ...
  [13 13 14]
  [14 14 14]
  [14 14 14]]

 [[17 16 23]
  [17 16 22]
  [15 14 19]
  ...
  [11 11 11]
  [11 11 11]
  [12 12 12]]

 [[16 15 20]
  [17 16 21]
  [16 15 20]
  ...
  [ 9  9  9]
  [ 9  9  9]
  [ 9  9  9]]]
True
[[[85 84 89]
  [89 88 91]
  [77 76 79]
  ...
  [45 49 44]
```

```
  [37 42 37]
  [40 46 41]]

 [[82 81 83]
  [83 82 84]
  [76 75 77]
  ...
  [44 44 41]
  [42 42 42]
  [43 43 43]]

 [[71 71 73]
  [75 74 77]
  [76 75 79]
  ...
  [44 43 43]
  [44 44 44]
  [44 44 44]]

 ...

 [[23 18 20]
  [24 18 21]
  [25 17 20]
  ...
  [12 11 15]
  [ 9  8 13]
  [11  9 14]]

 [[19 18 20]
  [18 17 19]
  [21 18 20]
  ...
  [13 12 15]
  [ 8  8 10]
  [ 9  9 11]]

 [[16 16 18]
  [16 16 18]
  [17 17 19]
  ...
  [13 12 15]
  [ 9  9 11]
  [ 9  9 11]]]
True
[[[85 84 89]
  [89 88 91]
  [77 76 79]
  ...
  [45 49 44]
  [37 42 37]
  [40 46 41]]

 [[82 81 83]
  [83 82 84]
  [76 75 77]
```

```
  ...
  [44 44 41]
  [42 42 42]
  [43 43 43]]

 [[71 71 73]
  [75 74 77]
  [76 75 79]
  ...
  [44 43 43]
  [44 44 44]
  [44 44 44]]

 ...

 [[23 18 20]
  [24 18 21]
  [25 17 20]
  ...
  [12 11 15]
  [ 9  8 13]
  [11  9 14]]

 [[19 18 20]
  [18 17 19]
  [21 18 20]
  ...
  [13 12 15]
  [ 8  8 10]
  [ 9  9 11]]

 [[16 16 18]
  [16 16 18]
  [17 17 19]
  ...
  [13 12 15]
  [ 9  9 11]
  [ 9  9 11]]]
True
[[[85 84 89]
  [89 88 91]
  [77 76 79]
  ...
  [45 49 44]
  [37 42 37]
  [40 46 41]]

 [[82 81 83]
  [83 82 84]
  [76 75 77]
  ...
  [44 44 41]
  [42 42 42]
  [43 43 43]]

 [[71 71 73]
```

```
  [75 74 77]
  [76 75 79]
  ...
  [44 43 43]
  [44 44 44]
  [44 44 44]]

 ...

 [[23 18 20]
  [24 18 21]
  [25 17 20]
  ...
  [12 11 15]
  [ 9  8 13]
  [11  9 14]]

 [[19 18 20]
  [18 17 19]
  [21 18 20]
  ...
  [13 12 15]
  [ 8  8 10]
  [ 9  9 11]]

 [[16 16 18]
  [16 16 18]
  [17 17 19]
  ...
  [13 12 15]
  [ 9  9 11]
  [ 9  9 11]]]
True
[[[85 84 89]
  [89 88 91]
  [77 76 79]
  ...
  [45 49 44]
  [37 42 37]
  [40 46 41]]

 [[82 81 83]
  [83 82 84]
  [76 75 77]
  ...
  [44 44 41]
  [42 42 42]
  [43 43 43]]

 [[71 71 73]
  [75 74 77]
  [76 75 79]
  ...
  [44 43 43]
  [44 44 44]
  [44 44 44]]
```

```
 ...

 [[23 18 20]
  [24 18 21]
  [25 17 20]
  ...
  [12 11 15]
  [ 9  8 13]
  [11  9 14]]

 [[19 18 20]
  [18 17 19]
  [21 18 20]
  ...
  [13 12 15]
  [ 8  8 10]
  [ 9  9 11]]

 [[16 16 18]
  [16 16 18]
  [17 17 19]
  ...
  [13 12 15]
  [ 9  9 11]
  [ 9  9 11]]]
True
[[[77 73 78]
  [79 74 80]
  [80 74 81]
  ...
  [49 46 49]
  [53 45 50]
  [51 42 48]]

 [[75 72 77]
  [80 75 81]
  [81 75 80]
  ...
  [51 46 51]
  [51 45 50]
  [54 49 54]]

 [[79 73 78]
  [80 75 78]
  [80 75 77]
  ...
  [53 48 52]
  [52 46 51]
  [54 49 54]]

 ...

 [[19 14 17]
  [19 14 17]
  [21 15 18]
```

```
  ...
  [ 9  9 11]
  [ 9  9 11]
  [12 11 13]]

 [[16 13 16]
  [17 15 16]
  [16 14 14]
  ...
  [ 9  9 11]
  [11 10 12]
  [ 9  9 11]]

 [[15 13 13]
  [15 12 14]
  [15 12 15]
  ...
  [ 9  9 11]
  [11 10 12]
  [11 10 12]]]
True
[[[77 73 78]
  [79 74 80]
  [80 74 81]
  ...
  [49 46 49]
  [53 45 50]
  [51 42 48]]

 [[75 72 77]
  [80 75 81]
  [81 75 80]
  ...
  [51 46 51]
  [51 45 50]
  [54 49 54]]

 [[79 73 78]
  [80 75 78]
  [80 75 77]
  ...
  [53 48 52]
  [52 46 51]
  [54 49 54]]

 ...

 [[19 14 17]
  [19 14 17]
  [21 15 18]
  ...
  [ 9  9 11]
  [ 9  9 11]
  [12 11 13]]

 [[16 13 16]
```

```
  [17 15 16]
  [16 14 14]
  ...
  [ 9  9 11]
  [11 10 12]
  [ 9  9 11]]

 [[15 13 13]
  [15 12 14]
  [15 12 15]
  ...
  [ 9  9 11]
  [11 10 12]
  [11 10 12]]]
True
[[[77 73 78]
  [79 74 80]
  [80 74 81]
  ...
  [49 46 49]
  [53 45 50]
  [51 42 48]]

 [[75 72 77]
  [80 75 81]
  [81 75 80]
  ...
  [51 46 51]
  [51 45 50]
  [54 49 54]]

 [[79 73 78]
  [80 75 78]
  [80 75 77]
  ...
  [53 48 52]
  [52 46 51]
  [54 49 54]]

 ...

 [[19 14 17]
  [19 14 17]
  [21 15 18]
  ...
  [ 9  9 11]
  [ 9  9 11]
  [12 11 13]]

 [[16 13 16]
  [17 15 16]
  [16 14 14]
  ...
  [ 9  9 11]
  [11 10 12]
  [ 9  9 11]]
```

```
 [[15 13 13]
  [15 12 14]
  [15 12 15]
  ...
  [ 9  9 11]
  [11 10 12]
  [11 10 12]]]
True
[[[81 72 80]
  [80 70 76]
  [83 73 76]
  ...
  [50 42 59]
  [46 40 59]
  [46 40 59]]

 [[79 70 76]
  [78 68 73]
  [83 73 76]
  ...
  [47 42 56]
  [46 41 55]
  [52 47 61]]

 [[74 68 73]
  [73 69 73]
  [74 71 74]
  ...
  [48 44 57]
  [48 45 59]
  [50 47 61]]

 ...

 [[20 13 16]
  [19 13 16]
  [19 14 17]
  ...
  [ 8 13 19]
  [ 8 13 19]
  [ 9 14 21]]

 [[16 13 18]
  [16 13 18]
  [16 13 18]
  ...
  [ 5 15 20]
  [ 5 15 21]
  [ 6 16 22]]

 [[14 13 18]
  [13 12 16]
  [13 12 16]
  ...
  [ 5 14 21]
```

```
  [ 5 15 21]
  [ 6 16 22]]]
True
[[[81 72 80]
  [80 70 76]
  [83 73 76]
  ...
  [50 42 59]
  [46 40 59]
  [46 40 59]]

 [[79 70 76]
  [78 68 73]
  [83 73 76]
  ...
  [47 42 56]
  [46 41 55]
  [52 47 61]]

 [[74 68 73]
  [73 69 73]
  [74 71 74]
  ...
  [48 44 57]
  [48 45 59]
  [50 47 61]]

 ...

 [[20 13 16]
  [19 13 16]
  [19 14 17]
  ...
  [ 8 13 19]
  [ 8 13 19]
  [ 9 14 21]]

 [[16 13 18]
  [16 13 18]
  [16 13 18]
  ...
  [ 5 15 20]
  [ 5 15 21]
  [ 6 16 22]]

 [[14 13 18]
  [13 12 16]
  [13 12 16]
  ...
  [ 5 14 21]
  [ 5 15 21]
  [ 6 16 22]]]
True
[[[81 72 80]
  [80 70 76]
  [83 73 76]
```

```
  ...
  [50 42 59]
  [46 40 59]
  [46 40 59]]

 [[79 70 76]
  [78 68 73]
  [83 73 76]
  ...
  [47 42 56]
  [46 41 55]
  [52 47 61]]

 [[74 68 73]
  [73 69 73]
  [74 71 74]
  ...
  [48 44 57]
  [48 45 59]
  [50 47 61]]

 ...

 [[20 13 16]
  [19 13 16]
  [19 14 17]
  ...
  [ 8 13 19]
  [ 8 13 19]
  [ 9 14 21]]

 [[16 13 18]
  [16 13 18]
  [16 13 18]
  ...
  [ 5 15 20]
  [ 5 15 21]
  [ 6 16 22]]

 [[14 13 18]
  [13 12 16]
  [13 12 16]
  ...
  [ 5 14 21]
  [ 5 15 21]
  [ 6 16 22]]]
True
[[[81 72 80]
  [80 70 76]
  [83 73 76]
  ...
  [50 42 59]
  [46 40 59]
  [46 40 59]]

 [[79 70 76]
```

```
  [78 68 73]
  [83 73 76]
  ...
  [47 42 56]
  [46 41 55]
  [52 47 61]]

 [[74 68 73]
  [73 69 73]
  [74 71 74]
  ...
  [48 44 57]
  [48 45 59]
  [50 47 61]]

 ...

 [[20 13 16]
  [19 13 16]
  [19 14 17]
  ...
  [ 8 13 19]
  [ 8 13 19]
  [ 9 14 21]]

 [[16 13 18]
  [16 13 18]
  [16 13 18]
  ...
  [ 5 15 20]
  [ 5 15 21]
  [ 6 16 22]]

 [[14 13 18]
  [13 12 16]
  [13 12 16]
  ...
  [ 5 14 21]
  [ 5 15 21]
  [ 6 16 22]]]
True
[[[61 63 85]
  [63 63 80]
  [68 65 77]
  ...
  [42 38 40]
  [39 34 37]
  [39 34 37]]

 [[68 63 82]
  [68 65 77]
  [66 65 70]
  ...
  [42 35 38]
  [42 35 38]
  [43 35 38]]
```

```
 [[65 61 70]
  [67 63 71]
  [66 65 72]
  ...
  [44 36 38]
  [46 36 37]
  [47 38 38]]

 ...

 [[15 10 20]
  [15 10 20]
  [15 10 20]
  ...
  [13 11 18]
  [13 11 18]
  [14 12 19]]

 [[15 10 20]
  [15 10 20]
  [17 13 22]
  ...
  [12 10 17]
  [12 10 17]
  [12 10 17]]

 [[15 10 20]
  [16 12 21]
  [17 13 22]
  ...
  [11  9 16]
  [ 9  8 15]
  [ 9  8 15]]]
True
[[[61 63 85]
  [63 63 80]
  [68 65 77]
  ...
  [42 38 40]
  [39 34 37]
  [39 34 37]]

 [[68 63 82]
  [68 65 77]
  [66 65 70]
  ...
  [42 35 38]
  [42 35 38]
  [43 35 38]]

 [[65 61 70]
  [67 63 71]
  [66 65 72]
  ...
  [44 36 38]
```

```
  [46 36 37]
  [47 38 38]]

 ...

 [[15 10 20]
  [15 10 20]
  [15 10 20]
  ...
  [13 11 18]
  [13 11 18]
  [14 12 19]]

 [[15 10 20]
  [15 10 20]
  [17 13 22]
  ...
  [12 10 17]
  [12 10 17]
  [12 10 17]]

 [[15 10 20]
  [16 12 21]
  [17 13 22]
  ...
  [11  9 16]
  [ 9  8 15]
  [ 9  8 15]]]
True
[[[61 63 85]
  [63 63 80]
  [68 65 77]
  ...
  [42 38 40]
  [39 34 37]
  [39 34 37]]

 [[68 63 82]
  [68 65 77]
  [66 65 70]
  ...
  [42 35 38]
  [42 35 38]
  [43 35 38]]

 [[65 61 70]
  [67 63 71]
  [66 65 72]
  ...
  [44 36 38]
  [46 36 37]
  [47 38 38]]

 ...

 [[15 10 20]
```

```
  [15 10 20]
  [15 10 20]
  ...
  [13 11 18]
  [13 11 18]
  [14 12 19]]

 [[15 10 20]
  [15 10 20]
  [17 13 22]
  ...
  [12 10 17]
  [12 10 17]
  [12 10 17]]

 [[15 10 20]
  [16 12 21]
  [17 13 22]
  ...
  [11  9 16]
  [ 9  8 15]
  [ 9  8 15]]]
True
[[[75 70 62]
  [72 68 60]
  [74 71 64]
  ...
  [45 41 43]
  [43 43 45]
  [42 41 44]]

 [[71 68 61]
  [69 65 59]
  [73 67 61]
  ...
  [42 38 38]
  [44 42 42]
  [44 42 42]]

 [[64 66 59]
  [69 67 60]
  [70 65 59]
  ...
  [43 41 41]
  [44 42 42]
  [44 42 42]]

 ...

 [[18 14 14]
  [17 12 13]
  [16 11 12]
  ...
  [ 8  8  9]
  [ 8  8 10]
  [ 8  8 10]]
```

```
 [[14 11 12]
  [15 13 13]
  [14 11 12]
  ...
  [ 8  8 10]
  [ 8  8 10]
  [ 8  8 10]]

 [[13 10 11]
  [13 10 11]
  [13 10 11]
  ...
  [ 8  8 10]
  [ 8  8 10]
  [ 8  8 10]]]
True
[[[75 70 62]
  [72 68 60]
  [74 71 64]
  ...
  [45 41 43]
  [43 43 45]
  [42 41 44]]

 [[71 68 61]
  [69 65 59]
  [73 67 61]
  ...
  [42 38 38]
  [44 42 42]
  [44 42 42]]

 [[64 66 59]
  [69 67 60]
  [70 65 59]
  ...
  [43 41 41]
  [44 42 42]
  [44 42 42]]

 ...

 [[18 14 14]
  [17 12 13]
  [16 11 12]
  ...
  [ 8  8  9]
  [ 8  8 10]
  [ 8  8 10]]

 [[14 11 12]
  [15 13 13]
  [14 11 12]
  ...
  [ 8  8 10]
```

```
  [ 8  8 10]
  [ 8  8 10]]

 [[13 10 11]
  [13 10 11]
  [13 10 11]
  ...
  [ 8  8 10]
  [ 8  8 10]
  [ 8  8 10]]]
True
[[[75 70 62]
  [72 68 60]
  [74 71 64]
  ...
  [45 41 43]
  [43 43 45]
  [42 41 44]]

 [[71 68 61]
  [69 65 59]
  [73 67 61]
  ...
  [42 38 38]
  [44 42 42]
  [44 42 42]]

 [[64 66 59]
  [69 67 60]
  [70 65 59]
  ...
  [43 41 41]
  [44 42 42]
  [44 42 42]]

 ...

 [[18 14 14]
  [17 12 13]
  [16 11 12]
  ...
  [ 8  8  9]
  [ 8  8 10]
  [ 8  8 10]]

 [[14 11 12]
  [15 13 13]
  [14 11 12]
  ...
  [ 8  8 10]
  [ 8  8 10]
  [ 8  8 10]]

 [[13 10 11]
  [13 10 11]
  [13 10 11]
```

```
  ...
  [ 8  8 10]
  [ 8  8 10]
  [ 8  8 10]]]
True
[[[75 70 62]
  [72 68 60]
  [74 71 64]
  ...
  [45 41 43]
  [43 43 45]
  [42 41 44]]

 [[71 68 61]
  [69 65 59]
  [73 67 61]
  ...
  [42 38 38]
  [44 42 42]
  [44 42 42]]

 [[64 66 59]
  [69 67 60]
  [70 65 59]
  ...
  [43 41 41]
  [44 42 42]
  [44 42 42]]

 ...

 [[18 14 14]
  [17 12 13]
  [16 11 12]
  ...
  [ 8  8  9]
  [ 8  8 10]
  [ 8  8 10]]

 [[14 11 12]
  [15 13 13]
  [14 11 12]
  ...
  [ 8  8 10]
  [ 8  8 10]
  [ 8  8 10]]

 [[13 10 11]
  [13 10 11]
  [13 10 11]
  ...
  [ 8  8 10]
  [ 8  8 10]
  [ 8  8 10]]]
True
[[[82 63 56]
```

```
  [84 65 58]
  [84 65 60]
  ...
  [47 44 50]
  [45 44 51]
  [44 43 50]]

 [[81 62 55]
  [82 64 57]
  [86 69 63]
  ...
  [47 45 51]
  [47 45 52]
  [47 45 52]]

 [[80 63 56]
  [80 63 57]
  [84 67 64]
  ...
  [45 44 50]
  [47 45 52]
  [49 47 54]]

 ...

 [[19 18 22]
  [20 19 23]
  [17 16 21]
  ...
  [11 15 16]
  [10 14 14]
  [11 16 15]]

 [[19 18 22]
  [19 17 23]
  [20 18 25]
  ...
  [10 14 14]
  [10 14 14]
  [10 14 14]]

 [[17 16 21]
  [17 16 21]
  [17 16 21]
  ...
  [ 9 13 13]
  [ 9 13 13]
  [10 14 14]]]
True
[[[82 63 56]
  [84 65 58]
  [84 65 60]
  ...
  [47 44 50]
  [45 44 51]
  [44 43 50]]
```

```
 [[81 62 55]
  [82 64 57]
  [86 69 63]
  ...
  [47 45 51]
  [47 45 52]
  [47 45 52]]

 [[80 63 56]
  [80 63 57]
  [84 67 64]
  ...
  [45 44 50]
  [47 45 52]
  [49 47 54]]

 ...

 [[19 18 22]
  [20 19 23]
  [17 16 21]
  ...
  [11 15 16]
  [10 14 14]
  [11 16 15]]

 [[19 18 22]
  [19 17 23]
  [20 18 25]
  ...
  [10 14 14]
  [10 14 14]
  [10 14 14]]

 [[17 16 21]
  [17 16 21]
  [17 16 21]
  ...
  [ 9 13 13]
  [ 9 13 13]
  [10 14 14]]]
True
[[[82 63 56]
  [84 65 58]
  [84 65 60]
  ...
  [47 44 50]
  [45 44 51]
  [44 43 50]]

 [[81 62 55]
  [82 64 57]
  [86 69 63]
  ...
  [47 45 51]
```

```
  [47 45 52]
  [47 45 52]]

 [[80 63 56]
  [80 63 57]
  [84 67 64]
  ...
  [45 44 50]
  [47 45 52]
  [49 47 54]]

 ...

 [[19 18 22]
  [20 19 23]
  [17 16 21]
  ...
  [11 15 16]
  [10 14 14]
  [11 16 15]]

 [[19 18 22]
  [19 17 23]
  [20 18 25]
  ...
  [10 14 14]
  [10 14 14]
  [10 14 14]]

 [[17 16 21]
  [17 16 21]
  [17 16 21]
  ...
  [ 9 13 13]
  [ 9 13 13]
  [10 14 14]]]
True
[[[82 63 56]
  [84 65 58]
  [84 65 60]
  ...
  [47 44 50]
  [45 44 51]
  [44 43 50]]

 [[81 62 55]
  [82 64 57]
  [86 69 63]
  ...
  [47 45 51]
  [47 45 52]
  [47 45 52]]

 [[80 63 56]
  [80 63 57]
  [84 67 64]
```

```
  ...
  [45 44 50]
  [47 45 52]
  [49 47 54]]

 ...

 [[19 18 22]
  [20 19 23]
  [17 16 21]
  ...
  [11 15 16]
  [10 14 14]
  [11 16 15]]

 [[19 18 22]
  [19 17 23]
  [20 18 25]
  ...
  [10 14 14]
  [10 14 14]
  [10 14 14]]

 [[17 16 21]
  [17 16 21]
  [17 16 21]
  ...
  [ 9 13 13]
  [ 9 13 13]
  [10 14 14]]]
True
[[[69 56 52]
  [70 56 55]
  [73 59 61]
  ...
  [45 40 31]
  [45 40 29]
  [44 39 28]]

 [[69 56 52]
  [70 56 55]
  [71 57 58]
  ...
  [45 37 29]
  [46 39 28]
  [47 40 29]]

 [[66 56 52]
  [66 55 54]
  [68 56 57]
  ...
  [46 40 30]
  [45 40 29]
  [43 39 28]]

 ...
```

```
 [[17 13 21]
  [17 12 19]
  [21 14 22]
  ...
  [14 13 18]
  [15 14 19]
  [14 13 18]]

 [[20 15 25]
  [19 14 23]
  [21 14 22]
  ...
  [14 13 18]
  [15 14 19]
  [15 14 19]]

 [[20 16 23]
  [18 15 22]
  [18 15 22]
  ...
  [14 13 18]
  [14 13 18]
  [15 14 19]]]
True
[[[69 56 52]
  [70 56 55]
  [73 59 61]
  ...
  [45 40 31]
  [45 40 29]
  [44 39 28]]

 [[69 56 52]
  [70 56 55]
  [71 57 58]
  ...
  [45 37 29]
  [46 39 28]
  [47 40 29]]

 [[66 56 52]
  [66 55 54]
  [68 56 57]
  ...
  [46 40 30]
  [45 40 29]
  [43 39 28]]

 ...

 [[17 13 21]
  [17 12 19]
  [21 14 22]
  ...
  [14 13 18]
```

```
  [15 14 19]
  [14 13 18]]

 [[20 15 25]
  [19 14 23]
  [21 14 22]
  ...
  [14 13 18]
  [15 14 19]
  [15 14 19]]

 [[20 16 23]
  [18 15 22]
  [18 15 22]
  ...
  [14 13 18]
  [14 13 18]
  [15 14 19]]]
True
[[[69 56 52]
  [70 56 55]
  [73 59 61]
  ...
  [45 40 31]
  [45 40 29]
  [44 39 28]]

 [[69 56 52]
  [70 56 55]
  [71 57 58]
  ...
  [45 37 29]
  [46 39 28]
  [47 40 29]]

 [[66 56 52]
  [66 55 54]
  [68 56 57]
  ...
  [46 40 30]
  [45 40 29]
  [43 39 28]]

 ...

 [[17 13 21]
  [17 12 19]
  [21 14 22]
  ...
  [14 13 18]
  [15 14 19]
  [14 13 18]]

 [[20 15 25]
  [19 14 23]
  [21 14 22]
```

```
  ...
  [14 13 18]
  [15 14 19]
  [15 14 19]]

 [[20 16 23]
  [18 15 22]
  [18 15 22]
  ...
  [14 13 18]
  [14 13 18]
  [15 14 19]]]
True
[[[61 61 61]
  [60 59 59]
  [63 60 61]
  ...
  [52 40 45]
  [52 39 47]
  [53 39 48]]

 [[59 59 59]
  [59 59 59]
  [62 62 62]
  ...
  [49 41 43]
  [48 38 41]
  [49 39 43]]

 [[62 59 59]
  [63 60 61]
  [65 63 63]
  ...
  [47 40 41]
  [46 39 40]
  [49 42 43]]

 ...

 [[24 16 19]
  [25 16 19]
  [26 16 19]
  ...
  [16 15 21]
  [17 16 23]
  [16 15 22]]

 [[25 17 20]
  [24 16 19]
  [24 16 19]
  ...
  [16 15 22]
  [16 15 22]
  [16 15 22]]

 [[24 16 19]
```

```
  [24 16 19]
  [24 16 19]
  ...
  [16 15 22]
  [16 15 22]
  [16 15 22]]]
True
[[[61 61 61]
  [60 59 59]
  [63 60 61]
  ...
  [52 40 45]
  [52 39 47]
  [53 39 48]]

 [[59 59 59]
  [59 59 59]
  [62 62 62]
  ...
  [49 41 43]
  [48 38 41]
  [49 39 43]]

 [[62 59 59]
  [63 60 61]
  [65 63 63]
  ...
  [47 40 41]
  [46 39 40]
  [49 42 43]]

 ...

 [[24 16 19]
  [25 16 19]
  [26 16 19]
  ...
  [16 15 21]
  [17 16 23]
  [16 15 22]]

 [[25 17 20]
  [24 16 19]
  [24 16 19]
  ...
  [16 15 22]
  [16 15 22]
  [16 15 22]]

 [[24 16 19]
  [24 16 19]
  [24 16 19]
  ...
  [16 15 22]
  [16 15 22]
  [16 15 22]]]
```

```
True
[[[61 61 61]
  [60 59 59]
  [63 60 61]
  ...
  [52 40 45]
  [52 39 47]
  [53 39 48]]

 [[59 59 59]
  [59 59 59]
  [62 62 62]
  ...
  [49 41 43]
  [48 38 41]
  [49 39 43]]

 [[62 59 59]
  [63 60 61]
  [65 63 63]
  ...
  [47 40 41]
  [46 39 40]
  [49 42 43]]

 ...

 [[24 16 19]
  [25 16 19]
  [26 16 19]
  ...
  [16 15 21]
  [17 16 23]
  [16 15 22]]

 [[25 17 20]
  [24 16 19]
  [24 16 19]
  ...
  [16 15 22]
  [16 15 22]
  [16 15 22]]

 [[24 16 19]
  [24 16 19]
  [24 16 19]
  ...
  [16 15 22]
  [16 15 22]
  [16 15 22]]]
True
[[[61 61 61]
  [60 59 59]
  [63 60 61]
  ...
  [52 40 45]
```

```
  [52 39 47]
  [53 39 48]]

 [[59 59 59]
  [59 59 59]
  [62 62 62]
  ...
  [49 41 43]
  [48 38 41]
  [49 39 43]]

 [[62 59 59]
  [63 60 61]
  [65 63 63]
  ...
  [47 40 41]
  [46 39 40]
  [49 42 43]]

 ...

 [[24 16 19]
  [25 16 19]
  [26 16 19]
  ...
  [16 15 21]
  [17 16 23]
  [16 15 22]]

 [[25 17 20]
  [24 16 19]
  [24 16 19]
  ...
  [16 15 22]
  [16 15 22]
  [16 15 22]]

 [[24 16 19]
  [24 16 19]
  [24 16 19]
  ...
  [16 15 22]
  [16 15 22]
  [16 15 22]]]
True
[[[68 66 68]
  [68 66 68]
  [71 68 71]
  ...
  [52 49 49]
  [51 49 49]
  [50 48 48]]

 [[70 67 69]
  [68 66 68]
  [70 67 69]
```

```
  ...
  [49 47 45]
  [50 48 48]
  [49 46 47]]

 [[70 67 69]
  [67 65 66]
  [71 68 69]
  ...
  [49 46 48]
  [50 47 50]
  [48 45 47]]

 ...

 [[23 24 20]
  [24 24 21]
  [25 23 23]
  ...
  [17 21 22]
  [17 21 21]
  [18 23 22]]

 [[23 24 20]
  [24 25 23]
  [24 24 24]
  ...
  [18 22 23]
  [17 21 23]
  [19 23 25]]

 [[23 24 20]
  [23 24 20]
  [23 24 20]
  ...
  [17 21 22]
  [18 23 22]
  [18 23 22]]]
True
[[[68 66 68]
  [68 66 68]
  [71 68 71]
  ...
  [52 49 49]
  [51 49 49]
  [50 48 48]]

 [[70 67 69]
  [68 66 68]
  [70 67 69]
  ...
  [49 47 45]
  [50 48 48]
  [49 46 47]]

 [[70 67 69]
```

```
  [67 65 66]
  [71 68 69]
  ...
  [49 46 48]
  [50 47 50]
  [48 45 47]]

 ...

 [[23 24 20]
  [24 24 21]
  [25 23 23]
  ...
  [17 21 22]
  [17 21 21]
  [18 23 22]]

 [[23 24 20]
  [24 25 23]
  [24 24 24]
  ...
  [18 22 23]
  [17 21 23]
  [19 23 25]]

 [[23 24 20]
  [23 24 20]
  [23 24 20]
  ...
  [17 21 22]
  [18 23 22]
  [18 23 22]]]
True
[[[68 66 68]
  [68 66 68]
  [71 68 71]
  ...
  [52 49 49]
  [51 49 49]
  [50 48 48]]

 [[70 67 69]
  [68 66 68]
  [70 67 69]
  ...
  [49 47 45]
  [50 48 48]
  [49 46 47]]

 [[70 67 69]
  [67 65 66]
  [71 68 69]
  ...
  [49 46 48]
  [50 47 50]
  [48 45 47]]
```

```
 ...

 [[23 24 20]
  [24 24 21]
  [25 23 23]
  ...
  [17 21 22]
  [17 21 21]
  [18 23 22]]

 [[23 24 20]
  [24 25 23]
  [24 24 24]
  ...
  [18 22 23]
  [17 21 23]
  [19 23 25]]

 [[23 24 20]
  [23 24 20]
  [23 24 20]
  ...
  [17 21 22]
  [18 23 22]
  [18 23 22]]]
True
[[[68 66 68]
  [68 66 68]
  [71 68 71]
  ...
  [52 49 49]
  [51 49 49]
  [50 48 48]]

 [[70 67 69]
  [68 66 68]
  [70 67 69]
  ...
  [49 47 45]
  [50 48 48]
  [49 46 47]]

 [[70 67 69]
  [67 65 66]
  [71 68 69]
  ...
  [49 46 48]
  [50 47 50]
  [48 45 47]]

 ...

 [[23 24 20]
  [24 24 21]
  [25 23 23]
```

```
  ...
  [17 21 22]
  [17 21 21]
  [18 23 22]]

 [[23 24 20]
  [24 25 23]
  [24 24 24]
  ...
  [18 22 23]
  [17 21 23]
  [19 23 25]]

 [[23 24 20]
  [23 24 20]
  [23 24 20]
  ...
  [17 21 22]
  [18 23 22]
  [18 23 22]]]
True
[[[64 60 78]
  [67 62 81]
  [65 58 77]
  ...
  [32 39 43]
  [32 39 40]
  [31 37 39]]

 [[64 58 74]
  [63 57 72]
  [59 54 66]
  ...
  [30 38 41]
  [28 36 40]
  [29 37 41]]

 [[59 53 60]
  [58 52 58]
  [60 54 59]
  ...
  [32 41 44]
  [33 42 46]
  [33 42 46]]

 ...

 [[19 22 29]
  [22 23 30]
  [23 22 29]
  ...
  [15 20 22]
  [13 20 22]
  [15 21 23]]

 [[24 23 30]
```

```
  [23 22 29]
  [23 22 29]
  ...
  [15 22 24]
  [16 22 24]
  [16 22 24]]

 [[22 21 28]
  [21 19 26]
  [21 19 26]
  ...
  [15 20 22]
  [13 20 22]
  [15 21 23]]]
True
[[[64 60 78]
  [67 62 81]
  [65 58 77]
  ...
  [32 39 43]
  [32 39 40]
  [31 37 39]]

 [[64 58 74]
  [63 57 72]
  [59 54 66]
  ...
  [30 38 41]
  [28 36 40]
  [29 37 41]]

 [[59 53 60]
  [58 52 58]
  [60 54 59]
  ...
  [32 41 44]
  [33 42 46]
  [33 42 46]]

 ...

 [[19 22 29]
  [22 23 30]
  [23 22 29]
  ...
  [15 20 22]
  [13 20 22]
  [15 21 23]]

 [[24 23 30]
  [23 22 29]
  [23 22 29]
  ...
  [15 22 24]
  [16 22 24]
  [16 22 24]]
```

```
 [[22 21 28]
  [21 19 26]
  [21 19 26]
  ...
  [15 20 22]
  [13 20 22]
  [15 21 23]]]
True
[[[64 60 78]
  [67 62 81]
  [65 58 77]
  ...
  [32 39 43]
  [32 39 40]
  [31 37 39]]

 [[64 58 74]
  [63 57 72]
  [59 54 66]
  ...
  [30 38 41]
  [28 36 40]
  [29 37 41]]

 [[59 53 60]
  [58 52 58]
  [60 54 59]
  ...
  [32 41 44]
  [33 42 46]
  [33 42 46]]

 ...

 [[19 22 29]
  [22 23 30]
  [23 22 29]
  ...
  [15 20 22]
  [13 20 22]
  [15 21 23]]

 [[24 23 30]
  [23 22 29]
  [23 22 29]
  ...
  [15 22 24]
  [16 22 24]
  [16 22 24]]

 [[22 21 28]
  [21 19 26]
  [21 19 26]
  ...
  [15 20 22]
```

```
  [13 20 22]
  [15 21 23]]]
True
[[[54 59 58]
  [56 57 57]
  [62 59 59]
  ...
  [41 37 36]
  [30 39 40]
  [31 41 43]]

 [[59 57 57]
  [60 59 59]
  [62 62 62]
  ...
  [37 39 41]
  [29 38 39]
  [29 39 40]]

 [[58 56 56]
  [62 59 59]
  [63 60 61]
  ...
  [37 40 40]
  [37 41 41]
  [37 42 42]]

 ...

 [[21 19 26]
  [22 21 27]
  [22 21 26]
  ...
  [17 18 17]
  [16 16 16]
  [16 16 16]]

 [[21 20 25]
  [21 20 25]
  [19 18 22]
  ...
  [20 17 17]
  [16 16 16]
  [17 18 17]]

 [[19 18 22]
  [20 19 23]
  [20 19 23]
  ...
  [17 16 16]
  [16 16 16]
  [16 16 16]]]
True
[[[54 59 58]
  [56 57 57]
  [62 59 59]
```

```
  ...
  [41 37 36]
  [30 39 40]
  [31 41 43]]

 [[59 57 57]
  [60 59 59]
  [62 62 62]
  ...
  [37 39 41]
  [29 38 39]
  [29 39 40]]

 [[58 56 56]
  [62 59 59]
  [63 60 61]
  ...
  [37 40 40]
  [37 41 41]
  [37 42 42]]

 ...

 [[21 19 26]
  [22 21 27]
  [22 21 26]
  ...
  [17 18 17]
  [16 16 16]
  [16 16 16]]

 [[21 20 25]
  [21 20 25]
  [19 18 22]
  ...
  [20 17 17]
  [16 16 16]
  [17 18 17]]

 [[19 18 22]
  [20 19 23]
  [20 19 23]
  ...
  [17 16 16]
  [16 16 16]
  [16 16 16]]]
True
[[[86 53 49]
  [79 49 48]
  [77 54 56]
  ...
  [56 37 48]
  [56 38 46]
  [54 37 45]]

 [[86 57 53]
```

```
  [82 57 55]
  [84 63 65]
  ...
  [50 39 44]
  [47 37 40]
  [48 38 41]]

 [[77 60 57]
  [78 62 61]
  [73 62 63]
  ...
  [48 38 42]
  [46 36 37]
  [48 39 39]]

 ...

 [[24 21 24]
  [24 21 24]
  [24 21 24]
  ...
  [19 21 22]
  [17 21 23]
  [18 22 24]]

 [[24 19 22]
  [23 19 22]
  [22 19 22]
  ...
  [15 20 21]
  [17 21 23]
  [19 23 25]]

 [[23 18 20]
  [24 19 22]
  [22 17 19]
  ...
  [16 21 21]
  [16 20 20]
  [17 21 21]]]
True
[[[86 53 49]
  [79 49 48]
  [77 54 56]
  ...
  [56 37 48]
  [56 38 46]
  [54 37 45]]

 [[86 57 53]
  [82 57 55]
  [84 63 65]
  ...
  [50 39 44]
  [47 37 40]
  [48 38 41]]
```

```
 [[77 60 57]
  [78 62 61]
  [73 62 63]
  ...
  [48 38 42]
  [46 36 37]
  [48 39 39]]

 ...

 [[24 21 24]
  [24 21 24]
  [24 21 24]
  ...
  [19 21 22]
  [17 21 23]
  [18 22 24]]

 [[24 19 22]
  [23 19 22]
  [22 19 22]
  ...
  [15 20 21]
  [17 21 23]
  [19 23 25]]

 [[23 18 20]
  [24 19 22]
  [22 17 19]
  ...
  [16 21 21]
  [16 20 20]
  [17 21 21]]]
True
[[[37 54 41]
  [48 56 47]
  [56 54 50]
  ...
  [39 38 36]
  [37 38 34]
  [36 37 32]]

 [[53 54 42]
  [52 51 46]
  [57 55 55]
  ...
  [41 39 35]
  [42 40 36]
  [40 39 35]]

 [[65 55 51]
  [65 57 53]
  [61 58 53]
  ...
  [44 40 38]
```

```
  [42 39 40]
  [40 38 39]]

 ...

 [[21 21 21]
  [22 21 21]
  [21 18 19]
  ...
  [17 17 18]
  [17 17 19]
  [16 16 18]]

 [[23 20 23]
  [22 19 22]
  [24 21 24]
  ...
  [19 18 20]
  [19 18 20]
  [17 17 19]]

 [[22 19 22]
  [22 19 22]
  [22 19 22]
  ...
  [20 19 22]
  [20 19 22]
  [20 19 22]]]
True
[[[37 54 41]
  [48 56 47]
  [56 54 50]
  ...
  [39 38 36]
  [37 38 34]
  [36 37 32]]

 [[53 54 42]
  [52 51 46]
  [57 55 55]
  ...
  [41 39 35]
  [42 40 36]
  [40 39 35]]

 [[65 55 51]
  [65 57 53]
  [61 58 53]
  ...
  [44 40 38]
  [42 39 40]
  [40 38 39]]

 ...

 [[21 21 21]
```

```
  [22 21 21]
  [21 18 19]
  ...
  [17 17 18]
  [17 17 19]
  [16 16 18]]

 [[23 20 23]
  [22 19 22]
  [24 21 24]
  ...
  [19 18 20]
  [19 18 20]
  [17 17 19]]

 [[22 19 22]
  [22 19 22]
  [22 19 22]
  ...
  [20 19 22]
  [20 19 22]
  [20 19 22]]]
True
[[[62 60 42]
  [64 61 42]
  [62 57 39]
  ...
  [30 38 54]
  [26 40 59]
  [25 40 60]]

 [[61 58 40]
  [61 58 43]
  [62 58 47]
  ...
  [34 40 55]
  [29 39 56]
  [29 39 57]]

 [[59 57 46]
  [60 58 48]
  [60 57 51]
  ...
  [35 40 45]
  [33 39 45]
  [34 40 46]]

 ...

 [[23 23 25]
  [22 21 25]
  [23 22 27]
  ...
  [19 18 20]
  [20 19 22]
  [19 18 20]]
```

```
 [[21 20 23]
  [21 20 24]
  [21 20 25]
  ...
  [19 18 20]
  [17 17 17]
  [19 19 19]]

 [[19 18 20]
  [20 19 22]
  [20 19 22]
  ...
  [16 16 17]
  [16 16 16]
  [17 18 17]]]
True
[[[52 58 62]
  [54 56 62]
  [59 58 65]
  ...
  [34 39 47]
  [32 40 46]
  [32 40 46]]

 [[52 55 59]
  [52 56 59]
  [53 57 59]
  ...
  [33 40 45]
  [32 39 48]
  [32 39 48]]

 [[57 57 59]
  [55 57 59]
  [52 56 58]
  ...
  [31 41 44]
  [32 41 44]
  [32 41 44]]

 ...

 [[18 22 24]
  [21 23 26]
  [22 21 26]
  ...
  [17 22 30]
  [17 22 30]
  [17 22 31]]

 [[16 19 23]
  [16 20 25]
  [16 19 23]
  ...
  [16 21 30]
```

```
  [15 19 28]
  [17 22 30]]

 [[17 16 21]
  [19 18 22]
  [19 18 22]
  ...
  [16 20 31]
  [13 18 29]
  [15 19 30]]]
True
[[[52 58 62]
  [54 56 62]
  [59 58 65]
  ...
  [34 39 47]
  [32 40 46]
  [32 40 46]]

 [[52 55 59]
  [52 56 59]
  [53 57 59]
  ...
  [33 40 45]
  [32 39 48]
  [32 39 48]]

 [[57 57 59]
  [55 57 59]
  [52 56 58]
  ...
  [31 41 44]
  [32 41 44]
  [32 41 44]]

 ...

 [[18 22 24]
  [21 23 26]
  [22 21 26]
  ...
  [17 22 30]
  [17 22 30]
  [17 22 31]]

 [[16 19 23]
  [16 20 25]
  [16 19 23]
  ...
  [16 21 30]
  [15 19 28]
  [17 22 30]]

 [[17 16 21]
  [19 18 22]
  [19 18 22]
```

```
  ...
  [16 20 31]
  [13 18 29]
  [15 19 30]]]
True
[[[41 48 48]
  [43 47 47]
  [52 52 52]
  ...
  [35 38 40]
  [36 36 36]
  [38 37 37]]

 [[41 49 44]
  [42 47 42]
  [51 52 48]
  ...
  [35 38 37]
  [36 36 36]
  [38 37 37]]

 [[47 53 48]
  [49 53 48]
  [56 57 52]
  ...
  [38 37 37]
  [36 36 36]
  [37 37 37]]

 ...

 [[18 24 30]
  [18 23 31]
  [18 23 32]
  ...
  [13 19 19]
  [11 18 17]
  [12 19 19]]

 [[15 20 26]
  [17 22 29]
  [15 20 26]
  ...
  [18 20 20]
  [14 19 19]
  [15 20 20]]

 [[12 18 24]
  [13 19 25]
  [13 19 25]
  ...
  [19 19 19]
  [19 19 19]
  [17 17 17]]]
True
[[[41 48 48]
```

```
  [43 47 47]
  [52 52 52]
  ...
  [35 38 40]
  [36 36 36]
  [38 37 37]]

 [[41 49 44]
  [42 47 42]
  [51 52 48]
  ...
  [35 38 37]
  [36 36 36]
  [38 37 37]]

 [[47 53 48]
  [49 53 48]
  [56 57 52]
  ...
  [38 37 37]
  [36 36 36]
  [37 37 37]]

 ...

 [[18 24 30]
  [18 23 31]
  [18 23 32]
  ...
  [13 19 19]
  [11 18 17]
  [12 19 19]]

 [[15 20 26]
  [17 22 29]
  [15 20 26]
  ...
  [18 20 20]
  [14 19 19]
  [15 20 20]]

 [[12 18 24]
  [13 19 25]
  [13 19 25]
  ...
  [19 19 19]
  [19 19 19]
  [17 17 17]]]
True
[[[56 54 50]
  [57 55 53]
  [59 57 57]
  ...
  [44 39 45]
  [36 35 40]
  [39 39 43]]
```

```
 [[52 49 52]
  [55 52 54]
  [59 56 59]
  ...
  [38 40 43]
  [31 37 39]
  [32 39 40]]

 [[53 48 51]
  [52 47 53]
  [55 51 58]
  ...
  [28 36 38]
  [27 40 41]
  [24 38 39]]

 ...

 [[26 20 29]
  [25 20 27]
  [24 21 24]
  ...
  [20 19 23]
  [21 20 23]
  [21 20 23]]

 [[30 21 29]
  [29 22 27]
  [26 21 24]
  ...
  [20 19 22]
  [20 19 22]
  [20 19 22]]

 [[25 20 23]
  [27 20 23]
  [27 20 23]
  ...
  [19 18 20]
  [19 18 20]
  [19 18 20]]]
True
[[[56 54 50]
  [57 55 53]
  [59 57 57]
  ...
  [44 39 45]
  [36 35 40]
  [39 39 43]]

 [[52 49 52]
  [55 52 54]
  [59 56 59]
  ...
  [38 40 43]
```

```
  [31 37 39]
  [32 39 40]]

 [[53 48 51]
  [52 47 53]
  [55 51 58]
  ...
  [28 36 38]
  [27 40 41]
  [24 38 39]]

 ...

 [[26 20 29]
  [25 20 27]
  [24 21 24]
  ...
  [20 19 23]
  [21 20 23]
  [21 20 23]]

 [[30 21 29]
  [29 22 27]
  [26 21 24]
  ...
  [20 19 22]
  [20 19 22]
  [20 19 22]]

 [[25 20 23]
  [27 20 23]
  [27 20 23]
  ...
  [19 18 20]
  [19 18 20]
  [19 18 20]]]
True
[[[59 54 66]
  [61 54 66]
  [61 54 66]
  ...
  [38 36 58]
  [40 35 56]
  [39 34 55]]

 [[59 54 66]
  [61 54 67]
  [61 54 66]
  ...
  [38 38 50]
  [42 39 51]
  [41 38 50]]

 [[58 53 65]
  [60 55 65]
  [61 55 62]
```

```
  ...
  [40 37 47]
  [40 37 47]
  [41 39 48]]

 ...

 [[23 21 30]
  [23 21 31]
  [23 21 30]
  ...
  [15 17 20]
  [12 16 18]
  [12 16 18]]

 [[22 20 29]
  [23 20 29]
  [23 19 28]
  ...
  [14 16 19]
  [14 19 20]
  [12 16 18]]

 [[22 17 27]
  [22 17 27]
  [22 17 27]
  ...
  [19 18 22]
  [19 18 22]
  [20 19 23]]]
True
[[[56 59 68]
  [52 56 61]
  [58 62 64]
  ...
  [44 40 42]
  [46 41 44]
  [46 41 44]]

 [[54 60 64]
  [54 55 60]
  [62 58 63]
  ...
  [44 41 44]
  [43 40 43]
  [42 39 41]]

 [[57 55 55]
  [55 52 54]
  [59 53 58]
  ...
  [42 40 42]
  [41 41 41]
  [43 43 43]]

 ...
```

```
 [[21 19 26]
  [20 18 25]
  [21 19 26]
  ...
  [17 16 21]
  [17 16 21]
  [16 15 20]]

 [[20 19 22]
  [19 18 21]
  [22 21 26]
  ...
  [19 18 22]
  [16 15 20]
  [16 15 20]]

 [[20 19 22]
  [20 19 21]
  [20 19 22]
  ...
  [16 16 19]
  [16 16 18]
  [16 16 18]]]
True
[[[56 59 68]
  [52 56 61]
  [58 62 64]
  ...
  [44 40 42]
  [46 41 44]
  [46 41 44]]

 [[54 60 64]
  [54 55 60]
  [62 58 63]
  ...
  [44 41 44]
  [43 40 43]
  [42 39 41]]

 [[57 55 55]
  [55 52 54]
  [59 53 58]
  ...
  [42 40 42]
  [41 41 41]
  [43 43 43]]

 ...

 [[21 19 26]
  [20 18 25]
  [21 19 26]
  ...
  [17 16 21]
```

```
  [17 16 21]
  [16 15 20]]

 [[20 19 22]
  [19 18 21]
  [22 21 26]
  ...
  [19 18 22]
  [16 15 20]
  [16 15 20]]

 [[20 19 22]
  [20 19 21]
  [20 19 22]
  ...
  [16 16 19]
  [16 16 18]
  [16 16 18]]]
True
[[[68 50 59]
  [70 52 58]
  [72 55 59]
  ...
  [32 41 49]
  [29 41 40]
  [30 42 40]]

 [[72 55 59]
  [70 55 59]
  [70 58 61]
  ...
  [29 40 42]
  [29 42 36]
  [28 41 34]]

 [[74 57 61]
  [76 58 62]
  [74 57 61]
  ...
  [31 40 42]
  [29 41 40]
  [30 42 40]]

 ...

 [[20 19 22]
  [20 19 22]
  [19 18 20]
  ...
  [18 22 24]
  [13 18 17]
  [14 19 18]]

 [[19 18 22]
  [19 18 21]
  [19 18 20]
```

```
  ...
  [14 16 17]
  [12 17 16]
  [13 18 17]]

 [[17 17 17]
  [20 19 22]
  [17 16 21]
  ...
  [16 16 18]
  [17 17 19]
  [17 17 19]]]
True
[[[68 50 59]
  [70 52 58]
  [72 55 59]
  ...
  [32 41 49]
  [29 41 40]
  [30 42 40]]

 [[72 55 59]
  [70 55 59]
  [70 58 61]
  ...
  [29 40 42]
  [29 42 36]
  [28 41 34]]

 [[74 57 61]
  [76 58 62]
  [74 57 61]
  ...
  [31 40 42]
  [29 41 40]
  [30 42 40]]

 ...

 [[20 19 22]
  [20 19 22]
  [19 18 20]
  ...
  [18 22 24]
  [13 18 17]
  [14 19 18]]

 [[19 18 22]
  [19 18 21]
  [19 18 20]
  ...
  [14 16 17]
  [12 17 16]
  [13 18 17]]

 [[17 17 17]
```

```
  [20 19 22]
  [17 16 21]
  ...
  [16 16 18]
  [17 17 19]
  [17 17 19]]]
True
[[[76 54 51]
  [75 54 51]
  [76 56 53]
  ...
  [55 34 41]
  [57 35 42]
  [58 35 42]]

 [[74 51 49]
  [74 53 49]
  [74 54 49]
  ...
  [56 37 42]
  [55 36 40]
  [53 34 38]]

 [[68 51 48]
  [66 48 45]
  [67 49 47]
  ...
  [48 33 37]
  [52 35 39]
  [52 35 39]]

 ...

 [[28 24 41]
  [26 23 35]
  [24 23 30]
  ...
  [12 17 16]
  [12 17 16]
  [12 17 16]]

 [[24 20 39]
  [24 21 36]
  [22 20 29]
  ...
  [19 23 24]
  [18 23 22]
  [18 23 22]]

 [[20 17 29]
  [22 19 32]
  [21 18 30]
  ...
  [19 24 23]
  [16 20 20]
  [19 24 23]]]
```

```
Processing image...
Converting RGB image to grayscale...
Resizing image to 28x28 scale...
Resized...
Image saved!
```

Data Collection from Microphone

In [13]:
```
!pip3 install sounddevice
```

```
Collecting sounddevice
  Downloading sounddevice-0.4.6-py3-none-win_amd64.whl.metadata (1.4 kB)
Requirement already satisfied: CFFI>=1.0 in c:\users\lenovo\anaconda3\lib\site-packa
ges (from sounddevice) (1.16.0)
Requirement already satisfied: pycparser in c:\users\lenovo\anaconda3\lib\site-packa
ges (from CFFI>=1.0->sounddevice) (2.21)
Downloading sounddevice-0.4.6-py3-none-win_amd64.whl (199 kB)
   ---------------------------------------- 0.0/199.7 kB ? eta -:--:--
   ---------------------------------------- 0.0/199.7 kB ? eta -:--:--
   -- ------------------------------------- 10.2/199.7 kB ? eta -:--:--
   ----- ---------------------------------- 30.7/199.7 kB 262.6 kB/s eta 0:00:01
   ------- -------------------------------- 41.0/199.7 kB 245.8 kB/s eta 0:00:01
   ---------------------------------------  194.6/199.7 kB 985.7 kB/s eta 0:00:01
   ---------------------------------------- 199.7/199.7 kB 869.1 kB/s eta 0:00:00
Installing collected packages: sounddevice
Successfully installed sounddevice-0.4.6
```

In [14]:
```
!pip3 install wavio
```

```
Collecting wavio
  Downloading wavio-0.0.8-py3-none-any.whl.metadata (5.7 kB)
Requirement already satisfied: numpy>=1.19.0 in c:\users\lenovo\anaconda3\lib\site-p
ackages (from wavio) (1.26.4)
Downloading wavio-0.0.8-py3-none-any.whl (9.4 kB)
Installing collected packages: wavio
Successfully installed wavio-0.0.8
```

In [15]:
```
!pip3 install scipy
```

```
Requirement already satisfied: scipy in c:\users\lenovo\anaconda3\lib\site-packages
(1.11.4)
Requirement already satisfied: numpy<1.28.0,>=1.21.6 in c:\users\lenovo\anaconda3\li
b\site-packages (from scipy) (1.26.4)
```

In [2]:
```
# import required libraries
import sounddevice as sd
from scipy.io.wavfile import write
import wavio as wv

# Sampling frequency
freq = 44100

# Recording duration
duration = 5

# Start recorder with the given values
# of duration and sample frequency
```

```
recording = sd.rec(int(duration * freq),
 samplerate=freq, channels=2)

# Record audio for the given number of seconds
sd.wait()

# This will convert the NumPy array to an audio
# file with the given sampling frequency
write("recording0.wav", freq, recording)
# Convert the NumPy array to audio file
wv.write("recording1.wav", recording, freq, sampwidth=2)
```

Web Scraping

In [21]:
```
!pip install bs4
```

```
Collecting bs4
  Downloading bs4-0.0.2-py2.py3-none-any.whl.metadata (411 bytes)
Requirement already satisfied: beautifulsoup4 in c:\users\lenovo\anaconda3\lib\site-
packages (from bs4) (4.12.2)
Requirement already satisfied: soupsieve>1.2 in c:\users\lenovo\anaconda3\lib\site-p
ackages (from beautifulsoup4->bs4) (2.5)
Downloading bs4-0.0.2-py2.py3-none-any.whl (1.2 kB)
Installing collected packages: bs4
Successfully installed bs4-0.0.2
```

In [22]:
```
pip install requests
```

```
Requirement already satisfied: requests in c:\users\lenovo\anaconda3\lib\site-packag
es (2.31.0)Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\lenovo\anaconda3
\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\lenovo\anaconda3\lib\site-pa
ckages (from requests) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\lenovo\anaconda3\lib\s
ite-packages (from requests) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\lenovo\anaconda3\lib\s
ite-packages (from requests) (2024.2.2)
```

In [3]:
```
import requests
from bs4 import BeautifulSoup

def getdata(url):
    r = requests.get(url)
    return r.text

htmldata = getdata("https://www.google.com/")
soup = BeautifulSoup(htmldata, 'html.parser')
for item in soup.find_all('img'):
    print(item['src'])
```

```
/images/branding/googlelogo/1x/googlelogo_white_background_color_272x92dp.png
```

In [25]:
```
pip install selenium
```

```
Collecting selenium
  Downloading selenium-4.18.1-py3-none-any.whl.metadata (6.9 kB)
Requirement already satisfied: urllib3<3,>=1.26 in c:\users\lenovo\anaconda3\lib\sit
e-packages (from urllib3[socks]<3,>=1.26->selenium) (2.0.7)
Collecting trio~=0.17 (from selenium)
  Downloading trio-0.25.0-py3-none-any.whl.metadata (8.7 kB)
Collecting trio-websocket~=0.9 (from selenium)
  Downloading trio_websocket-0.11.1-py3-none-any.whl.metadata (4.7 kB)
Requirement already satisfied: certifi>=2021.10.8 in c:\users\lenovo\anaconda3\lib\s
ite-packages (from selenium) (2024.2.2)
Requirement already satisfied: typing_extensions>=4.9.0 in c:\users\lenovo\anaconda3
\lib\site-packages (from selenium) (4.9.0)
Collecting attrs>=23.2.0 (from trio~=0.17->selenium)
  Downloading attrs-23.2.0-py3-none-any.whl.metadata (9.5 kB)
Requirement already satisfied: sortedcontainers in c:\users\lenovo\anaconda3\lib\sit
e-packages (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: idna in c:\users\lenovo\anaconda3\lib\site-packages
(from trio~=0.17->selenium) (3.4)
Collecting outcome (from trio~=0.17->selenium)
  Downloading outcome-1.3.0.post0-py2.py3-none-any.whl.metadata (2.6 kB)
Requirement already satisfied: sniffio>=1.3.0 in c:\users\lenovo\anaconda3\lib\site-
packages (from trio~=0.17->selenium) (1.3.0)
Requirement already satisfied: cffi>=1.14 in c:\users\lenovo\anaconda3\lib\site-pack
ages (from trio~=0.17->selenium) (1.16.0)
Collecting wsproto>=0.14 (from trio-websocket~=0.9->selenium)
  Downloading wsproto-1.2.0-py3-none-any.whl.metadata (5.6 kB)
Requirement already satisfied: pysocks!=1.5.7,<2.0,>=1.5.6 in c:\users\lenovo\anacon
da3\lib\site-packages (from urllib3[socks]<3,>=1.26->selenium) (1.7.1)
Requirement already satisfied: pycparser in c:\users\lenovo\anaconda3\lib\site-packa
ges (from cffi>=1.14->trio~=0.17->selenium) (2.21)
Collecting h11<1,>=0.9.0 (from wsproto>=0.14->trio-websocket~=0.9->selenium)
  Downloading h11-0.14.0-py3-none-any.whl.metadata (8.2 kB)
Downloading selenium-4.18.1-py3-none-any.whl (10.0 MB)
   ---------------------------------------- 0.0/10.0 MB ? eta -:--:--
   ---------------------------------------- 0.1/10.0 MB 4.1 MB/s eta 0:00:03
   - -------------------------------------- 0.3/10.0 MB 3.9 MB/s eta 0:00:03
   -- ------------------------------------- 0.7/10.0 MB 6.0 MB/s eta 0:00:02
   --- ------------------------------------ 0.8/10.0 MB 5.1 MB/s eta 0:00:02
   --- ------------------------------------ 1.0/10.0 MB 4.8 MB/s eta 0:00:02
   ----- ---------------------------------- 1.4/10.0 MB 5.4 MB/s eta 0:00:02
   ------ --------------------------------- 1.6/10.0 MB 5.5 MB/s eta 0:00:02
   ------- -------------------------------- 1.9/10.0 MB 5.6 MB/s eta 0:00:02
   --------- ------------------------------ 2.3/10.0 MB 5.9 MB/s eta 0:00:02
   ---------- ----------------------------- 2.7/10.0 MB 6.2 MB/s eta 0:00:02
   ------------ --------------------------- 3.0/10.0 MB 6.2 MB/s eta 0:00:02
   ------------- -------------------------- 3.3/10.0 MB 6.3 MB/s eta 0:00:02
   -------------- ------------------------- 3.6/10.0 MB 6.3 MB/s eta 0:00:02
   --------------- ------------------------ 3.8/10.0 MB 6.2 MB/s eta 0:00:02
   ----------------- ---------------------- 4.2/10.0 MB 6.5 MB/s eta 0:00:01
   ------------------ --------------------- 4.6/10.0 MB 6.7 MB/s eta 0:00:01
   -------------------- ------------------- 5.1/10.0 MB 6.8 MB/s eta 0:00:01
   --------------------- ------------------ 5.4/10.0 MB 6.7 MB/s eta 0:00:01
   ----------------------- ---------------- 5.8/10.0 MB 6.8 MB/s eta 0:00:01
   ------------------------ --------------- 6.0/10.0 MB 6.7 MB/s eta 0:00:01
   ------------------------- -------------- 6.4/10.0 MB 6.8 MB/s eta 0:00:01
   -------------------------- ------------- 6.7/10.0 MB 6.8 MB/s eta 0:00:01
```

```
   ---------------------------- ----------- 7.1/10.0 MB 6.9 MB/s eta 0:00:01
   ---------------------------- ----------- 7.5/10.0 MB 6.9 MB/s eta 0:00:01
   ------------------------------ --------- 7.8/10.0 MB 6.9 MB/s eta 0:00:01
   -------------------------------- ------- 8.1/10.0 MB 6.9 MB/s eta 0:00:01
   -------------------------------- ------- 8.3/10.0 MB 6.9 MB/s eta 0:00:01
   -------------------------------- ------- 8.4/10.0 MB 6.7 MB/s eta 0:00:01
   ---------------------------------- ----- 9.0/10.0 MB 6.8 MB/s eta 0:00:01
   ---------------------------------- ----- 9.0/10.0 MB 6.8 MB/s eta 0:00:01
   ---------------------------------- ----- 9.0/10.0 MB 6.8 MB/s eta 0:00:01
   ------------------------------------  9.8/10.0 MB 6.7 MB/s eta 0:00:01
   ------------------------------------  10.0/10.0 MB 6.7 MB/s eta 0:00:01
   ------------------------------------- 10.0/10.0 MB 6.4 MB/s eta 0:00:00
Downloading trio-0.25.0-py3-none-any.whl (467 kB)
   ---------------------------------------- 0.0/467.2 kB ? eta -:--:--
   ------------------------------------  460.8/467.2 kB 14.5 MB/s eta 0:00:01
   ---------------------------------------- 467.2/467.2 kB 5.9 MB/s eta 0:00:00
Downloading trio_websocket-0.11.1-py3-none-any.whl (17 kB)
Downloading attrs-23.2.0-py3-none-any.whl (60 kB)
   ---------------------------------------- 0.0/60.8 kB ? eta -:--:--
   ---------------------------------------- 60.8/60.8 kB 1.6 MB/s eta 0:00:00
Downloading wsproto-1.2.0-py3-none-any.whl (24 kB)
Downloading outcome-1.3.0.post0-py2.py3-none-any.whl (10 kB)
Downloading h11-0.14.0-py3-none-any.whl (58 kB)
   ---------------------------------------- 0.0/58.3 kB ? eta -:--:--
   ---------------------------------------- 58.3/58.3 kB 1.0 MB/s eta 0:00:00
Installing collected packages: h11, attrs, wsproto, outcome, trio, trio-websocket, s
elenium
  Attempting uninstall: attrs
    Found existing installation: attrs 23.1.0
    Uninstalling attrs-23.1.0:
      Successfully uninstalled attrs-23.1.0
Successfully installed attrs-23.2.0 h11-0.14.0 outcome-1.3.0.post0 selenium-4.18.1 t
rio-0.25.0 trio-websocket-0.11.1 wsproto-1.2.0
Note: you may need to restart the kernel to use updated packages.
```

In [4]:

```
!pip install selenium
#!apt-get update # to update ubuntu to correctly run apt install
#!apt install chromium-chromedriver
#!cp /usr/lib/chromium-browser/chromedriver/ usr/bin
import sys
sys.path.insert(0,'/usr/lib/chromium-browser/chromedriver')

from selenium import webdriver
import time
import requests
import shutil
import os
import getpass
import urllib.request
import io
import time
from PIL import Image

user = getpass.getuser()
chrome_options = webdriver.ChromeOptions()
chrome_options.add_argument('--headless')
```

```
chrome_options.add_argument('--no-sandbox')
chrome_options.add_argument('--disable-dev-shm-usage')
driver = webdriver.Chrome(options=chrome_options)

search_url = "https://www.google.com/search?q={q}&tbm=isch&tbs=sur%3Afc&hl=en&ved=0
driver.get(search_url.format(q='Car'))

def scroll_to_end(driver):
    driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
    time.sleep(5)

def getImageUrls(name,totalImgs,driver):

    search_url = "https://www.google.com/search?q={q}&tbm=isch&tbs=sur%3Afc&hl=en&v
    driver.get(search_url.format(q=name))
    img_urls = set()
    img_count = 0
    results_start = 0

    while(img_count<totalImgs):

        scroll_to_end(driver)

        thumbnail_results = driver.find_elements('xpath',"//img[contains(@class,'Q4
        totalResults=len(thumbnail_results)
        print(f"Found: {totalResults} search results. Extracting links from{results

        for img in thumbnail_results[results_start:totalResults]:

            img.click()
            time.sleep(2)
            actual_images = driver.find_elements_by_css_selector('img.n3VNCb')
            for actual_image in actual_images:
                if actual_image.get_attribute('src') and 'https' in actual_image.ge
                    img_urls.add(actual_image.get_attribute('src'))

            img_count=len(img_urls)

            if img_count >= totalImgs:
                print(f"Found: {img_count} image links")
                break
            else:
                print("Found:", img_count, "looking for more image links ...")
                load_more_button = driver.find_element_by_css_selector(".mye4qd")
                driver.execute_script("document.querySelector('.mye4qd').click();")
                results_start = len(thumbnail_results)
    return img_urls


def downloadImages(folder_path,file_name,url):
    try:

        image_content = requests.get(url).content
    except Exception as e:
        print(f"ERROR - COULD NOT DOWNLOAD {url} - {e}")
    try:
```

```
        image_file = io.BytesIO(image_content)
        image = Image.open(image_file).convert('RGB')

        file_path = os.path.join(folder_path, file_name)

        with open(file_path, 'wb') as f:
            image.save(f, "JPEG", quality=85)
        print(f"SAVED - {url} - AT: {file_path}")
    except Exception as e:
        print(f"ERROR - COULD NOT SAVE {url} - {e}")

def saveInDestFolder(searchNames,destDir,totalImgs,driver):
    for name in list(searchNames):
        path=os.path.join(destDir,name)
        if not os.path.isdir(path):
            os.mkdir(path)
        print('Current Path',path)
        totalLinks=getImageUrls(name,totalImgs,driver)
        print('totalLinks',totalLinks)
    if totalLinks is None:
            print('images not found for :',name)

    else:
        for i, link in enumerate(totalLinks):
            file_name = f"{i:150}.jpg"
            downloadImages(path,file_name,link)

searchNames=['cat']
destDir=f'Downloads/DataSci'
totalImgs=5

saveInDestFolder(searchNames,destDir,totalImgs,driver)
```

```
Requirement already satisfied: selenium in c:\users\lenovo\anaconda3\lib\site-packag
es (4.18.1)
Requirement already satisfied: urllib3<3,>=1.26 in c:\users\lenovo\anaconda3\lib\sit
e-packages (from urllib3[socks]<3,>=1.26->selenium) (2.0.7)
Requirement already satisfied: trio~=0.17 in c:\users\lenovo\anaconda3\lib\site-pack
ages (from selenium) (0.25.0)
Requirement already satisfied: trio-websocket~=0.9 in c:\users\lenovo\anaconda3\lib
\site-packages (from selenium) (0.11.1)
Requirement already satisfied: certifi>=2021.10.8 in c:\users\lenovo\anaconda3\lib\s
ite-packages (from selenium) (2024.2.2)
Requirement already satisfied: typing_extensions>=4.9.0 in c:\users\lenovo\anaconda3
\lib\site-packages (from selenium) (4.9.0)
Requirement already satisfied: attrs>=23.2.0 in c:\users\lenovo\anaconda3\lib\site-p
ackages (from trio~=0.17->selenium) (23.2.0)
Requirement already satisfied: sortedcontainers in c:\users\lenovo\anaconda3\lib\sit
e-packages (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: idna in c:\users\lenovo\anaconda3\lib\site-packages
(from trio~=0.17->selenium) (3.4)
Requirement already satisfied: outcome in c:\users\lenovo\anaconda3\lib\site-package
s (from trio~=0.17->selenium) (1.3.0.post0)
Requirement already satisfied: sniffio>=1.3.0 in c:\users\lenovo\anaconda3\lib\site-
packages (from trio~=0.17->selenium) (1.3.0)
Requirement already satisfied: cffi>=1.14 in c:\users\lenovo\anaconda3\lib\site-pack
ages (from trio~=0.17->selenium) (1.16.0)
Requirement already satisfied: wsproto>=0.14 in c:\users\lenovo\anaconda3\lib\site-p
ackages (from trio-websocket~=0.9->selenium) (1.2.0)
Requirement already satisfied: pysocks!=1.5.7,<2.0,>=1.5.6 in c:\users\lenovo\anacon
da3\lib\site-packages (from urllib3[socks]<3,>=1.26->selenium) (1.7.1)
Requirement already satisfied: pycparser in c:\users\lenovo\anaconda3\lib\site-packa
ges (from cffi>=1.14->trio~=0.17->selenium) (2.21)
Requirement already satisfied: h11<1,>=0.9.0 in c:\users\lenovo\anaconda3\lib\site-p
ackages (from wsproto>=0.14->trio-websocket~=0.9->selenium) (0.14.0)
Current Path Downloads/DataSci\cat
Found: 0 search results. Extracting links from0:0
Found: 0 search results. Extracting links from0:0
Found: 0 search results. Extracting links from0:0
Found: 0 search results. Extracting links from0:0
Found: 0 search results. Extracting links from0:0
Found: 0 search results. Extracting links from0:0
Found: 0 search results. Extracting links from0:0
Found: 0 search results. Extracting links from0:0
Found: 0 search results. Extracting links from0:0
Found: 0 search results. Extracting links from0:0
Found: 0 search results. Extracting links from0:0
Found: 0 search results. Extracting links from0:0
Found: 0 search results. Extracting links from0:0
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Cell In[4], line 109
    106 destDir=f'Downloads/DataSci'
    107 totalImgs=5
--> 109 saveInDestFolder(searchNames,destDir,totalImgs,driver)

Cell In[4], line 95, in saveInDestFolder(searchNames, destDir, totalImgs, driver)
     93         os.mkdir(path)
     94     print('Current Path',path)
---> 95     totalLinks=getImageUrls(name,totalImgs,driver)
     96     print('totalLinks',totalLinks)
     97 if totalLinks is None:

Cell In[4], line 43, in getImageUrls(name, totalImgs, driver)
     39 results_start = 0
     41 while(img_count<totalImgs):
---> 43     scroll_to_end(driver)
     45     thumbnail_results = driver.find_elements('xpath',"//img[contains(@clas
s,'Q4LuWd')]")
     46     totalResults=len(thumbnail_results)

Cell In[4], line 31, in scroll_to_end(driver)
     29 def scroll_to_end(driver):
     30     driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
---> 31     time.sleep(5)

KeyboardInterrupt:
```

Web Scraping of Movies Information using BeautifulSoup

In [5]:
```
from requests import get
url = 'https://www.imdb.com/search/title?release_date=2017&sort=num_votes,desc&page
agent = {"User-Agent": 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.3
response = get(url, headers = agent)
print(response)
print(response.text[:500])
```

```
<Response [200]>
<!DOCTYPE html><html lang="en-US" xmlns:og="http://opengraphprotocol.org/schema/" xm
lns:fb="http://www.facebook.com/2008/fbml"><head><meta charSet="utf-8"/><meta name
="viewport" content="width=device-width"/><script>if(typeof uet === 'function'){ uet
('bb', 'LoadTitle', {wb: 1}); }</script><script>window.addEventListener('load', (eve
nt) => {
        if (typeof window.csa !== 'undefined' && typeof window.csa === 'function') {
            var csaLatencyPlugin = window.csa('Content', {
```

In [6]:
```
from bs4 import BeautifulSoup
soup = BeautifulSoup(response.text, 'html.parser')
headers = {'Accept-Language': 'en-US,en;q=0.8'}
type(soup)
```

Out[6]: bs4.BeautifulSoup

In [7]:
```
movie_containers = soup.find_all('li', class_ = 'ipc-metadata-list-summary-item')
print(type(movie_containers))
print(len(movie_containers))
```

```
<class 'bs4.element.ResultSet'>
50
```

In [8]:
```
first_movie = movie_containers[0]
first_movie
```

Out[8]:
```
<li class="ipc-metadata-list-summary-item"><div class="ipc-metadata-list-summary-i
tem__c"><div class="ipc-metadata-list-summary-item__tc"><span aria-disabled="fals
e" class="ipc-metadata-list-summary-item__t"></span><div class="sc-ab6fa25a-3 bVYf
LY dli-parent"><div class="sc-ab6fa25a-2 gOsifL"><div class="sc-e5a25b0f-0 jQjDIb
dli-poster-container"><div class="ipc-poster ipc-poster--base ipc-poster--dynamic-
width ipc-sub-grid-item ipc-sub-grid-item--span-2" role="group"><div aria-label="a
dd to watchlist" class="ipc-watchlist-ribbon ipc-focusable ipc-watchlist-ribbon--s
ipc-watchlist-ribbon--base ipc-watchlist-ribbon--loading ipc-watchlist-ribbon--onI
mage ipc-poster__watchlist-ribbon" role="button" tabindex="0"><svg class="ipc-watc
hlist-ribbon__bg" height="34px" role="presentation" viewbox="0 0 24 34" width="24p
x" xmlns="http://www.w3.org/2000/svg"><polygon class="ipc-watchlist-ribbon__bg-rib
bon" fill="#000000" points="24 0 0 0 0 32 12.2436611 26.2926049 24 31.7728343"></p
olygon><polygon class="ipc-watchlist-ribbon__bg-hover" points="24 0 0 0 0 32 12.24
36611 26.2926049 24 31.7728343"></polygon><polygon class="ipc-watchlist-ribbon__bg
-shadow" points="24 31.7728343 24 33.7728343 12.2436611 28.2926049 0 34 0 32 12.24
36611 26.2926049"></polygon></svg><div class="ipc-watchlist-ribbon__icon" role="pr
esentation"><svg class="ipc-loader ipc-loader--circle ipc-watchlist-ribbon__loade
r" data-testid="watchlist-ribbon-loader" height="48px" role="presentation" version
="1.1" viewbox="0 0 48 48" width="48px" xmlns="http://www.w3.org/2000/svg"><g clas
s="ipc-loader__container" fill="currentColor"><circle class="ipc-loader__circle ip
c-loader__circle--one" cx="24" cy="9" r="4"></circle><circle class="ipc-loader__ci
rcle ipc-loader__circle--two" cx="35" cy="14" r="4"></circle><circle class="ipc-lo
ader__circle ipc-loader__circle--three" cx="39" cy="24" r="4"></circle><circle cla
ss="ipc-loader__circle ipc-loader__circle--four" cx="35" cy="34" r="4"></circle><c
ircle class="ipc-loader__circle ipc-loader__circle--five" cx="24" cy="39" r="4"></
circle><circle class="ipc-loader__circle ipc-loader__circle--six" cx="13" cy="34"
r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--seven" cx="9"
cy="24" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--eigh
t" cx="13" cy="14" r="4"></circle></g></svg></div></div><div class="ipc-media ipc-
media--poster-27x40 ipc-image-media-ratio--poster-27x40 ipc-media--base ipc-media-
-poster-m ipc-poster__poster-image ipc-media__img" style="width:100%"><img alt="Hu
gh Jackman in Logan (2017)" class="ipc-image" loading="lazy" sizes="50vw, (min-wid
th: 480px) 34vw, (min-width: 600px) 26vw, (min-width: 1024px) 16vw, (min-width: 12
80px) 16vw" src="https://m.media-amazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhL
Tg3NWEtMTY4OTEyMzJhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX140_CR0,1,140,207_.
jpg" srcset="https://m.media-amazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhLTg3N
WEtMTY4OTEyMzJhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX140_CR0,1,140,207_.jpg
140w, https://m.media-amazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhLTg3NWEtMTY4
OTEyMzJhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX210_CR0,2,210,311_.jpg 210w, h
ttps://m.media-amazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhLTg3NWEtMTY4OTEyMzJ
hZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX280_CR0,3,280,414_.jpg 280w" width="1
40"/></div><a aria-label="View title page for Logan" class="ipc-lockup-overlay ipc
-focusable" href="/title/tt3315342/?ref_=sr_i_1"><div class="ipc-lockup-overlay__s
creen"></div></a></div></div><div class="sc-b0691f29-0 jbYPfh"><div class="ipc-tit
le ipc-title--base ipc-title--title ipc-title-link-no-icon ipc-title--on-textPrima
ry sc-b0691f29-9 klOwFB dli-title"><a class="ipc-title-link-wrapper" href="/title/
tt3315342/?ref_=sr_t_1" tabindex="0"><h3 class="ipc-title__text">1. Logan</h3></a>
</div><div class="sc-b0691f29-7 hrgukm dli-title-metadata"><span class="sc-b0691f2
9-8 ilsLEX dli-title-metadata-item">2017</span><span class="sc-b0691f29-8 ilsLEX d
li-title-metadata-item">2h 17m</span><span class="sc-b0691f29-8 ilsLEX dli-title-m
etadata-item">R-16</span></div><span class="sc-b0691f29-1 grHDBY"><div class="sc-e
2dbc1a3-0 ajrIH sc-b0691f29-2 bhhtyj dli-ratings-container" data-testid="ratingGro
up--container"><span aria-label="IMDb rating: 8.1" class="ipc-rating-star ipc-rati
ng-star--base ipc-rating-star--imdb ratingGroup--imdb-rating" data-testid="ratingG
roup--imdb-rating"><svg class="ipc-icon ipc-icon--star-inline" fill="currentColor"
height="24" role="presentation" viewbox="0 0 24 24" width="24" xmlns="http://www.w
```

3.org/2000/svg"><path d="M12 20.1l5.82 3.682c1.066.675 2.37-.322 2.09-1.584l-1.543
-6.926 5.146-4.667c.94-.85.435-2.465-.799-2.567l-6.773-.602L13.29.89a1.38 1.38 0 0
0-2.581 0l-2.65 6.53-6.774.602C.052 8.126-.453 9.74.486 10.59l5.147 4.666-1.542 6.
926c-.28 1.262 1.023 2.26 2.09 1.585L12 20.099z"></path></svg>8.1<span class="ipc-
rating-star--voteCount"> (<!-- -->827K<!-- -->)<button aria-label="R
ate Logan" class="ipc-rate-button sc-e2dbc1a3-1 jboOQc ratingGroup--user-rating ip
c-rate-button--unrated ipc-rate-button--base" data-testid="rate-button"><span clas
s="ipc-rating-star ipc-rating-star--base ipc-rating-star--rate"><svg class="ipc-ic
on ipc-icon--star-border-inline" fill="currentColor" height="24" role="presentatio
n" viewbox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg"><path d="M22.
724 8.217l-6.786-.587-2.65-6.22c-.477-1.133-2.103-1.133-2.58 0l-2.65 6.234-6.772.5
73c-1.234.098-1.739 1.636-.8 2.446l5.146 4.446-1.542 6.598c-.28 1.202 1.023 2.153
2.09 1.51l5.818-3.495 5.819 3.509c1.065.643 2.37-.308 2.089-1.51l-1.542-6.612 5.14
5-4.446c.94-.81.45-2.348-.785-2.446zm-10.726 8.89l-5.272 3.174 1.402-5.983-4.655-
4.026 6.141-.531 2.384-5.634 2.398 5.648 6.14.531-4.654 4.026 1.402 5.983-5.286-3.
187z"></path></svg>Rate</button>
<span class="sc-b0901df4-0 bcQdDJ metacri
tic-score-box" style="background-color:#54A72A">77<span class="metacritic-s
core-label">Metascore <div class="sc-ab6fa25a-4 ggHbBR d
li-post-element"><button aria-disabled="false" aria-label="See more information ab
out Logan" class="ipc-icon-button dli-info-icon ipc-icon-button--base ipc-icon-but
ton--onAccent2" role="button" tabindex="0" title="See more information about Loga
n"><svg class="ipc-icon ipc-icon--info" fill="currentColor" height="24" role="pres
entation" viewbox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg"><path
d="M0 0h24v24H0V0z" fill="none"></path><path d="M11 7h2v2h-2zm0 4h2v6h-2zm1-9C6.48
2 2 6.48 2 12s4.48 10 10 10 10-4.48 10-10S17.52 2 12 2zm0 18c-4.41 0-8-3.59-8-8s3.
59-8 8-8 8 3.59 8 8-3.59 8-8 8z"></path></svg></button> <div class="sc-
ab6fa25a-1 bBwFsP"><div class="ipc-html-content ipc-html-content--base sc-ab6fa25a
-0 bhexuD dli-plot-container" role="presentation"><div class="ipc-html-content-inn
er-div">In a future where mutants are nearly extinct, an elderly and weary Logan l
eads a quiet life. But when Laura, a mutant child pursued by scientists, comes to
him for help, he must get her to safety.

In [9]:
```
first_movie.div
```

Out[9]: <div aria-label="add to watchlist" class="ipc-watchlist-ribbon ipc-focusable ipc-watchlist-ribbon--s ipc-watchlist-ribbon--base ipc-watchlist-ribbon--loading ipc-watchlist-ribbon--onImage ipc-poster__watchlist-ribbon" role="button" tabindex="0"><svg class="ipc-watchlist-ribbon__bg" height="34px" role="presentation" viewbox="0 0 24 34" width="24px" xmlns="http://www.w3.org/2000/svg"><polygon class="ipc-watchlist-ribbon__bg-ribbon" fill="#000000" points="24 0 0 0 0 32 12.2436611 26.2926049 24 31.7728343"></polygon><polygon class="ipc-watchlist-ribbon__bg-hover" points="24 0 0 0 0 32 12.2436611 26.2926049 24 31.7728343"></polygon><polygon class="ipc-watchlist-ribbon__bg-shadow" points="24 31.7728343 24 33.7728343 12.2436611 28.2926049 0 34 0 32 12.2436611 26.2926049"></polygon></svg> <svg class="ipc-loader ipc-loader--circle ipc-watchlist-ribbon__loader" data-testid="watchlist-ribbon-loader" height="48px" role="presentation" version="1.1" viewbox="0 0 48 48" width="48px" xmlns="http://www.w3.org/2000/svg"><g class="ipc-loader__container" fill="currentColor"><circle class="ipc-loader__circle ipc-loader__circle--one" cx="24" cy="9" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--two" cx="35" cy="14" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--three" cx="39" cy="24" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--four" cx="35" cy="34" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--five" cx="24" cy="39" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--six" cx="13" cy="34" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--seven" cx="9" cy="24" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--eight" cx="13" cy="14" r="4"></circle></g></svg> <a aria-label="View title page for Logan" class="ipc-lockup-overlay ipc-focusable" href="/title/tt3315342/?ref_=sr_i_1"> <h3 class="ipc-title__text">1. Logan</h3> 20172h 17mR-16 <svg class="ipc-icon ipc-icon--star-inline" fill="currentColor" height="24" role="presentation" viewbox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg"><path d="M12 20.1l5.82

```
3.682c1.066.675 2.37-.322 2.09-1.584l-1.543-6.926 5.146-4.667c.94-.85.435-2.465-.7
99-2.567l-6.773-.602L13.29 9.89a1.38 1.38 0 0 0-2.581 0l-2.65 6.53-6.774.602C.052 8.
126-.453 9.74.486 10.59l5.147 4.666-1.542 6.926c-.28 1.262 1.023 2.26 2.09 1.585L1
2 20.099z"></path></svg>8.1<span class="ipc-rating-star--voteCount"> (<!-- -->827K
<!-- -->)</span></span><button aria-label="Rate Logan" class="ipc-rate-button sc-e
2dbc1a3-1 jboOQc ratingGroup--user-rating ipc-rate-button--unrated ipc-rate-button
--base" data-testid="rate-button"><span class="ipc-rating-star ipc-rating-star--ba
se ipc-rating-star--rate"><svg class="ipc-icon ipc-icon--star-border-inline" fill
="currentColor" height="24" role="presentation" viewbox="0 0 24 24" width="24" xml
ns="http://www.w3.org/2000/svg"><path d="M22.724 8.217l-6.786-.587-2.65-6.22c-.477
-1.133-2.103-1.133-2.58 0l-2.65 6.234-6.772.573c-1.234.098-1.739 1.636-.8 2.446l5.
146 4.446-1.542 6.598c-.28 1.202 1.023 2.153 2.09 1.51l5.818-3.495 5.819 3.509c1.0
65.643 2.37-.308 2.089-1.51l-1.542-6.612 5.145-4.446c.94-.81.45-2.348-.785-2.446zm
-10.726 8.89l-5.272 3.174 1.402-5.983-4.655-4.026 6.141-.531 2.384-5.634 2.398 5.6
48 6.14.531-4.654 4.026 1.402 5.983-5.286-3.187z"></path></svg><span class="ipc-ra
ting-star--rate">Rate</span></span></button></div><span class="sc-b0691f29-11 TmkK
M"><span class="sc-b0901df4-0 bcQdDJ metacritic-score-box" style="background-colo
r:#54A72A">77</span><span class="metacritic-score-label">Metascore</span></span></
span></div><div class="sc-ab6fa25a-4 ggHbBR dli-post-element"><button aria-disable
d="false" aria-label="See more information about Logan" class="ipc-icon-button dli
-info-icon ipc-icon-button--base ipc-icon-button--onAccent2" role="button" tabinde
x="0" title="See more information about Logan"><svg class="ipc-icon ipc-icon--inf
o" fill="currentColor" height="24" role="presentation" viewbox="0 0 24 24" width
="24" xmlns="http://www.w3.org/2000/svg"><path d="M0 0h24v24H0V0z" fill="none"></p
ath><path d="M11 7h2v2h-2zm0 4h2v6h-2zm1-9C6.48 2 2 6.48 2 12s4.48 10 10 10 10-4.4
8 10-10S17.52 2 12 2zm0 18c-4.41 0-8-3.59-8-8s3.59-8 8-8 8 3.59 8 8-3.59 8-8 8z">
</path></svg></button></div></div><div class="sc-ab6fa25a-1 bBwFsP"><div class="ip
c-html-content ipc-html-content--base sc-ab6fa25a-0 bhexuD dli-plot-container" rol
e="presentation"><div class="ipc-html-content-inner-div">In a future where mutants
are nearly extinct, an elderly and weary Logan leads a quiet life. But when Laura,
a mutant child pursued by scientists, comes to him for help, he must get her to sa
fety.</div></div></div></div></div></div>
```

In [10]:
```
first_movie.a
```

Out[10]:
```
<a aria-label="View title page for Logan" class="ipc-lockup-overlay ipc-focusable"
href="/title/tt3315342/?ref_=sr_i_1"><div class="ipc-lockup-overlay__screen"></div
></a>
```

In [11]:
```
first_movie.h3
```

Out[11]:
```
<h3 class="ipc-title__text">1. Logan</h3>
```

In [12]:
```
first_name = first_movie.h3.text
first_name[3:]
```

Out[12]:
```
'Logan'
```

In [13]:
```
first_year = movie_containers[0].find('span', class_ = "sc-b0691f29-8 ilsLEX dli-ti
first_year
```

Out[13]:
```
<span class="sc-b0691f29-8 ilsLEX dli-title-metadata-item">2017</span>
```

In [14]:
```
first_year = first_year.text
first_year
```

Out[14]: '2017'

In [15]:
```
first_imdb = movie_containers[0].find('span', class_ = "ipc-rating-star ipc-rating-
first_imdb.text[:3]
```

Out[15]: '8.1'

In [16]:
```
first_mscore = movie_containers[0].find('span', class_ = 'sc-b0901df4-0 bcQdDJ meta
first_mscore = first_mscore.text
print(first_mscore)
```

```
77
```

In [17]:
```
first_votes = movie_containers[0].find('span', class_ = 'ipc-rating-star--voteCount
first_votes.text[2:-1]
```

Out[17]: '827K'

In [18]:
```
# Lists to store the scraped data in
names = []
years = []
imdb_ratings = []
metascores = []
votes = []
# Extract data from individual movie container
for container in movie_containers:
  names.append(container.find('h3', class_ = "ipc-title__text").text[3:])
  # print(container.find('h3', class_ = "ipc-title__text").text[3:])
  years.append(container.find('span', class_ = "sc-b0691f29-8 ilsLEX dli-title-meta
  # print(container.find('span', class_ = "sc-b0691f29-8 ilsLEX dli-title-metadata-
  imdb_ratings.append(container.find('span', class_ = "ipc-rating-star ipc-rating-s
  # print(container.find('span', class_ = "ipc-rating-star ipc-rating-star--base ip
  rate = container.find('span', class_='sc-b0901df4-0 bcQdDJ metacritic-score-box')
  if rate:
    metascores.append(rate.text)
  else:
    metascores.append(0)
  # print(container.find('span', class_ = 'ipc-rating-star--voteCount').text[2:-1])
  votes.append(container.find('span', class_ = 'ipc-rating-star--voteCount').text[2
print(len(names))
print(len(years))
print(len(imdb_ratings))
print(len(metascores))
print(len(votes)
)
```

```
50
50
50
50
50
```

In [19]:
```
import pandas as pd
test_df = pd.DataFrame({'movie': names,
'year': years,
'imdb': imdb_ratings,
```

```
'metascore': metascores,
'votes': votes
})
print(test_df.info())
test_df
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   movie      50 non-null     object
 1   year       50 non-null     object
 2   imdb       50 non-null     object
 3   metascore  50 non-null     object
 4   votes      50 non-null     object
dtypes: object(5)
memory usage: 2.1+ KB
None
```

Out[19]:

	movie	year	imdb	metascore	votes
0	Logan	[2017]	8.1	77	827K
1	Thor: Ragnarok	[2017]	7.9	74	813K
2	Guardians of the Galaxy Vol. 2	[2017]	7.6	67	756K
3	Dunkirk	[2017]	7.8	94	736K
4	Spider-Man: Homecoming	[2017]	7.4	73	716K
5	Wonder Woman	[2017]	7.3	76	698K
6	Get Out	[2017]	7.8	85	691K
7	Star Wars: Episode VIII - The Last Jedi	[2017]	6.9	84	670K
8	Blade Runner 2049	[2017]	8.0	81	658K
9	Baby Driver	[2017]	7.5	86	605K
10	It	[2017]	7.3	69	603K
11	Coco	[2017]	8.4	81	586K
12	Three Billboards Outside Ebbing, Missouri	[2017]	8.1	88	553K
13	Money Heist	[2017–2021]	8.2	0	529K
14	John Wick: Chapter 2	[2017]	7.4	75	509K
15	Justice League	[2017]	6.1	45	477K
16	The Shape of Water	[2017]	7.3	87	446K
17	Dark	[2017–2020]	8.7	0	440K
18	Jumanji: Welcome to the Jungle	[2017]	6.9	58	436K
19	Kingsman: The Golden Circle	[2017]	6.7	44	361K
20	Kong: Skull Island	[2017]	6.7	62	345K
21	Ozark	[2017–2022]	8.5	0	344K
22	Pirates of the Caribbean: Salazar's Revenge	[2017]	6.5	39	344K
23	Beauty and the Beast	[2017]	7.1	65	333K
24	Mindhunter	[2017–2019]	8.6	0	333K
25	Lady Bird	[2017]	7.4	93	326K
26	13 Reasons Why	[2017–2020]	7.5	0	314K
27	Call Me by Your Name	[2017]	7.8	94	313K
28	The Greatest Showman	[2017]	7.5	48	310K
29	Alien: Covenant	[2017]	6.4	65	302K

	movie	year	imdb	metascore	votes
30	Murder on the Orient Express	[2017]	6.5	52	295K
31	War for the Planet of the Apes	[2017]	7.4	82	280K
32	Wind River	[2017]	7.7	73	279K
33	The Punisher	[2017–2019]	8.4	0	263K
34	The Handmaid's Tale	[2017–]	8.4	0	257K
35	Fast & Furious 8	[2017]	6.6	56	253K
36	Life	[2017]	6.6	54	252K
37	Mother!	[2017]	6.6	76	249K
38	The Hitman's Bodyguard	[2017]	6.9	47	246K
39	I, Tonya	[2017]	7.5	77	242K
40	King Arthur: Legend of the Sword	[2017]	6.7	41	232K
41	Ghost in the Shell	[2017]	6.3	52	227K
42	Big Little Lies	[2017–]	8.4	0	223K
43	Darkest Hour	[2017]	7.4	75	220K
44	The End of the F***ing World	[2017–2019]	8.0	0	218K
45	American Made	[2017]	7.1	65	207K
46	Atomic Blonde	[2017]	6.7	63	206K
47	The Mummy	[2017]	5.4	34	206K
48	Baywatch	[2017]	5.5	37	201K
49	Bright	[2017]	6.3	29	201K

In [30]:
```
test_df.to_csv('Movies.csv', index=False)
```

In [31]:
```
movies = pd.read_csv('Movies.csv')
movies.head()
```

Out[31]:

	movie	year	imdb	metascore	votes
0	Logan	<span class="sc-b0691f29-8 ilsLEX dli-title-me...	8.1	77	827K
1	Thor: Ragnarok	<span class="sc-b0691f29-8 ilsLEX dli-title-me...	7.9	74	813K
2	Guardians of the Galaxy Vol. 2	<span class="sc-b0691f29-8 ilsLEX dli-title-me...	7.6	67	756K
3	Dunkirk	<span class="sc-b0691f29-8 ilsLEX dli-title-me...	7.8	94	736K
4	Spider-Man: Homecoming	<span class="sc-b0691f29-8 ilsLEX dli-title-me...	7.4	73	716K

In [32]:

```
from time import time
from time import sleep
from requests import get
from random import randint
from IPython.core.display import clear_output
from bs4 import BeautifulSoup
from IPython.core.display import clear_output
pages = ['1','2','3','4','5']
years_url = ['2017', '2018', '2019', '2020']

# Redeclaring the lists to store data in
names = []
years = []
imdb_ratings = []
metascores = []
votes = []
# Preparing the monitoring of the loop
start_time = time()
requests = 0
# For every year in the interval 2000-2017
for year_url in years_url:

    # For every page in the interval 1-4
    for page in pages:

        # Make a get request
        url = f'https://www.imdb.com/search/title/?release_date={year_url}-01-01,{y
        agent = {"User-Agent": 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebK
        response = get(url,headers = agent)
        print(response.text[:500])
        #response = get('https://www.imdb.com/search/title?release_date=' + year_ur
        #'&sort=num_votes,desc&page=' + page, headers = headers)
```

```
  Cell In[32], line 27
    url = f'https://www.imdb.com/search/title/?release_date={year_url}-01-01,{year_u
rl}-12-31&sort=num_votes,desc
          ^
SyntaxError: unterminated string literal (detected at line 27)
```

Data Preparation

In [ ]: