In [6]:

```
# lib's
import numpy as np
import pandas as pd

import requests
from selenium import webdriver
from selenium.webdriver.common.by import By

import os
import re
import shutil
import time
import warnings
warnings.filterwarnings("ignore")

```

Problem Statement 1:

Write a python code using web scraping method for creating a list of

1. Name of Diseases,
2. URLs associated with diseases and,
3. Icon images of diseases.

Save the list as a CSV file.

Create the folder using python commands to save the icon images.

URL of webpage: https://dermnetnz.org/image-library/ (https://dermnetnz.org/image-library/)

Use selenium libraries

In [7]:

```
# constant
URL = "https://dermnetnz.org/image-library"
IMAGE_DIR = "images_of_diseases"
```

In [8]:

```
# function

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

def save_image(image_url, image_path):
    r = requests.get(image_url, stream=True)
    if r.status_code == 200:
        with open(image_path, 'wb') as f:
            r.raw.decode_content = True
            shutil.copyfileobj(r.raw, f)
```

In [11]:

```python
# initiating the webdriver.
driver = webdriver.Chrome(r'E:\chromedriver.exe')
driver.get(URL)

# this is just to ensure that the page is loaded
time.sleep(1)
print(driver.title, ' || ', driver.current_url)

# to scroll down full page
scroll_to_end(driver)

# to find the element by class name
element = driver.find_elements(By.CLASS_NAME, "imageList__group")
# print(element)

```

```
Image library | DermNet NZ  ||  https://dermnetnz.org/image-library (https://dermnetnz.org/image-library)
```

In [12]:

```

# to create image folder
os.makedirs(IMAGE_DIR, exist_ok = True)

for ele in range(len(element)):

    element_text_list = element[ele].text.split('\n')
    element_tag_a_list = element[ele].find_elements(By.TAG_NAME, "a")
    element_tag_img_list = element[ele].find_elements(By.TAG_NAME, "img")

    # print(len(element_text_list), len(element_tag_a_list), len(element_tag_i

    if len(element_tag_a_list) != 0:

        for data in range(len(element_text_list)):

            Name_of_Disease = element_text_list[data]
            Disease_Page_URL = element_tag_a_list[data].get_attribute("href")
            Image_of_Diseases = element_tag_img_list[data].get_attribute("src"

            print(Name_of_Disease, end = " | ")
            print(Disease_Page_URL, end = " | ")
            print(Image_of_Diseases, "\n")

            image_name = re.sub('\W+','', Name_of_Disease)+".jpg"
            image_path = os.path.join(IMAGE_DIR,image_name)

            # to save images
            save_image(Image_of_Diseases, image_path)

    else:
        pass

# to close
driver.close()
```

```
Lentigo maligna melanoma images | https://dermnetnz.org/topics/lentigo-maligna-
melanoma-images (https://dermnetnz.org/topics/lentigo-maligna-melanoma-images)
 | https://dermnetnz.org/assets/Uploads/lmm5-big2__FocusFillWzE1MCwxMTAsInkiLDF
d.jpg (https://dermnetnz.org/assets/Uploads/lmm5-big2__FocusFillWzE1MCwxMTAsInk
iLDFd.jpg)

Lesion, tumour and cancer images | https://dermnetnz.org/image-catalogue/lesion
-tumour-and-cancer-images (https://dermnetnz.org/image-catalogue/lesion-tumour-
and-cancer-images) | https://dermnetnz.org/assets/Uploads/ep-n5-s__FocusFillWzE
1MCwxMTAsInkiLDFd.jpg (https://dermnetnz.org/assets/Uploads/ep-n5-s__FocusFillW
zE1MCwxMTAsInkiLDFd.jpg)

Lichen aureus images | https://dermnetnz.org/topics/capillaritis-images (http
s://dermnetnz.org/topics/capillaritis-images) | https://dermnetnz.org/assets/Up
loads/2509__FocusFillWzE1MCwxMTAsIngiLDdd.jpg (https://dermnetnz.org/assets/Upl
oads/2509__FocusFillWzE1MCwxMTAsIngiLDdd.jpg)


Lichen planus images | https://dermnetnz.org/images/lichen-planus-images (http
s://dermnetnz.org/images/lichen-planus-images) | https://dermnetnz.org/assets/U
```

Problem Statement 2:

Complete the python function to get the output of below cases :

i. case 1: n = 1, v = 1

ii. case 2: n= 2, v = 23 (Note: 23 is derived as 1 + 22)

iii. case 3: n= 3, v = 356 (Note: 356 is derived as 1+22+333)

iv. case 4: n= 4, v = 4800 (Note: 356 is derived as 1+22+333+4444)

def mystery(n):

...

...

...

return v

In [13]:

```python
def mystery(n):
    v = 0
    for i in range(1, n+1):
        v1 = ''
        for j in range(i):
            v1 += str(i)
        v += int(v1)

    return v

print(mystery(1))
print(mystery(2))
print(mystery(3))
print(mystery(4))
print(mystery(5))
```

```
1
23
356
4800
60355
```

In [ ]:

```

```