```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Feb  5 15:06:57 2024

@author: Dawood Sarfraz
"""

"""
Picture is array of numbers or pixels. Each pixel has own value. Red, Green, Blue
[having values between 0 -255] and Alpha. Alpha define the tranpancy of pixel at that
location.
In type uint8 means numbers (2**8 = 256) in this image can go form (0 to 255)
This image i'm using have dimensions (256, 256, 3), Here 3 stands for Red, Green, Blue.
when you change data type information lose

other Data Types
uint8 - 0 to (2**8 -1)
uint16 - 0 to (2**16 -1)
uint32 - 0 to (2**32 -1)
float - -1 to 1 or 0 to 1
int8 - (2**7) to (2**7 - 1) = -128 to 127
int16 - -(2**15) to (2**31 - 1) = 32768 to 32767
1nt32 - -(2**31) to (2**31 - 1) = (-2147483648 to 2147483647)

Function that converts images to desired dtypes and properly rescale their values
img_as_float - convert to 64-bit floating point
img-as_ubyte - convert to 8-bit uint
img_as_uint - convert to 16-bit uint
img_as_int - convert to 16-bit int


"""
from skimage import io
from skimage import img_as_float
import numpy as np
import matplotlib.pyplot as plt


input_image = io.imread("images/dog.jpg")
print(input_image) # array of numbers
```

```
# print(input_image[0][0]) # value at 0x0
print("Minimum value in Image",input_image.min())
print("Maximum value in Image",input_image.max())
float_img = img_as_float(input_image)
print("Float value Image", float_img)
print("Minimum value in Image", float_img.min())
print("Maximum value in Image", float_img.max())
# input and floated image will be same just value changed 0-255 and 0.0 -1.0
plt.imshow(input_image)
plt.imshow(float_img)
print("\n\n Complete \n\n ")
```

```
# Let's create artificial Image and fill will random numbers
random_image = np.random.random([500, 500])
plt.imshow(random_image)
print(random_image) # as type is float64 so values will be between 0 and 1
print("Minimum value in Image",random_image.min())
print("Maximum value in Image",random_image.max())


# Let's half all image values an see results
new_image = input_image * 0.5
plt.imshow(new_image) # will show different image pixels
print(new_image) # all values will be half
print("Minimum value in Image",new_image.min())
print("Maximum value in Image",new_image.max())

# Create box in image
#              [x,     y,   cannels] = [R, G, B]
input_image[100:150, 100:150, :]     = [250,0,150]
plt.imshow(input_image)
```