STEGANOGRAPHY

TEAM PRESENTATION

👦	👦	👦	👦
ทศพล กนกพิพัฒน์วงศ์ 65056040	**ณัฐวุฒิ ทองศรีนุ่น** 65056036	**อานิก เวพาสยนันท์** 65056099	**ภูริวัฒน์ แสงระวี** 65056071

Agenda :

- Definition , History and Difference between Cryptography & Steganography
- Type of Steganography & Basic Model
- How LSB technique works?
- Python Code Example

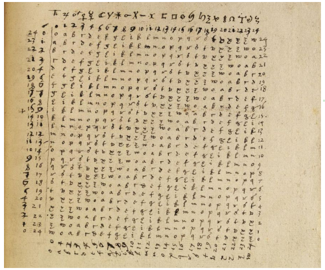
What is Steganography?

What is Steganography?

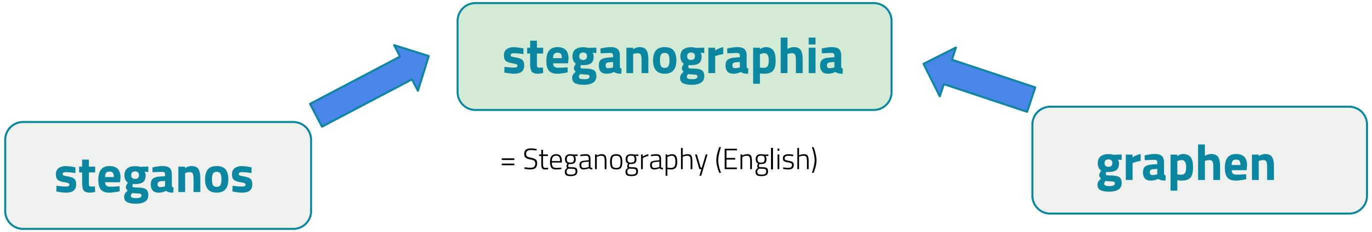
Steganography is the process of hiding a secret message within a larger one in such a way that someone can not know the presence or contents of the hidden message. **The purpose of Steganography is to maintain secret communication between two parties. Unlike cryptography, which conceals the contents of a secret message, steganography conceals the very fact that a message is communicated.** Although steganography differs from cryptography, there are many analogies between the two, and some authors classify steganography as a form of cryptography since hidden communication is a type of secret message.

Some History

- – **The First historical record** can be traced **back to 440 BC in Greece** from ***Herodotus*** mentioned two example in his ***Histories -*** One of those example :Demaratus sent a warning about forthcoming attack to Greece by writing it directly on the wooden backing of **wax tablet before applying its beeswax surface**



- The First recorded use in term was in 1499 by ***Johannes Trithemius*** in his ***Steganographia*** which some messages or characters are hidden in something else (image , other text etc.) which this part written by invisible ink (or Security ink)



steganographia

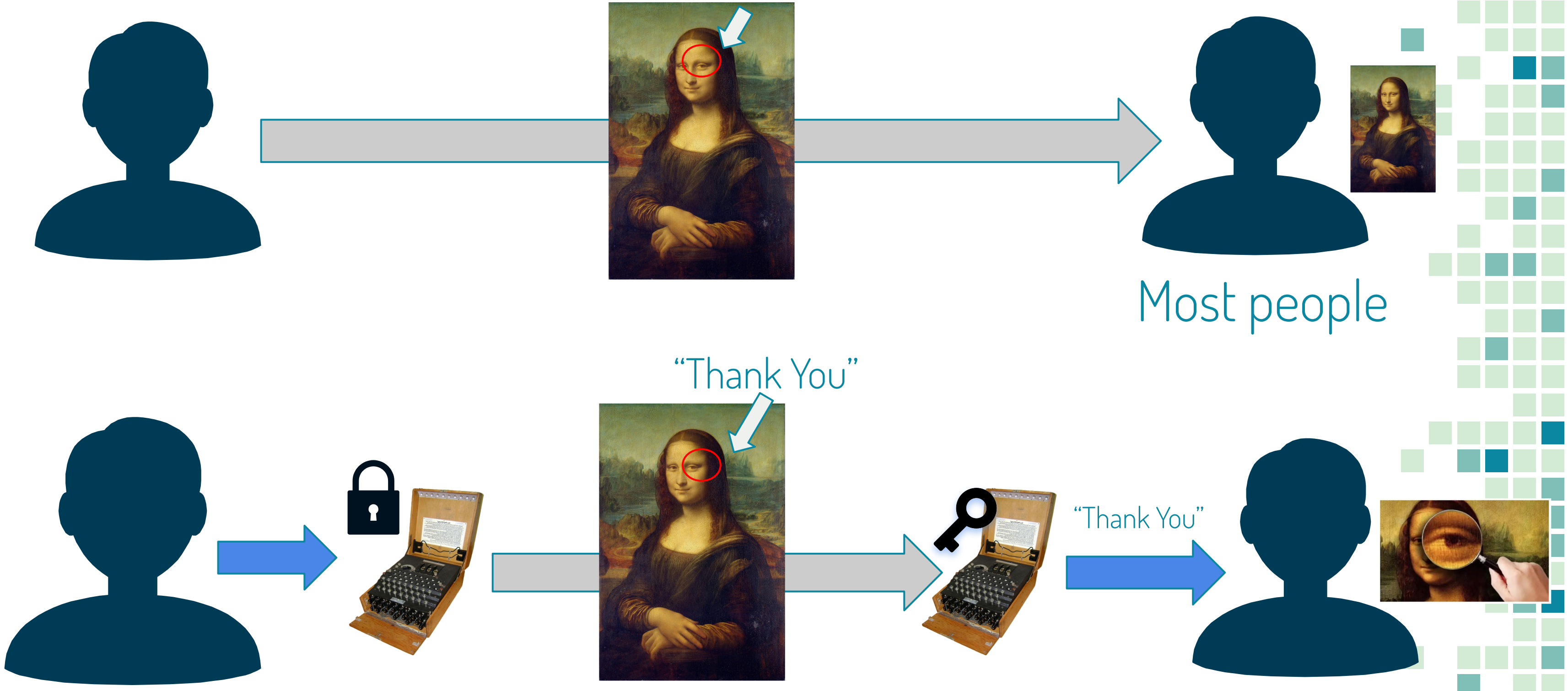
= Steganography (English)

steganos

Concealed or Hidden

graphen

Drawing or Writing



Most people

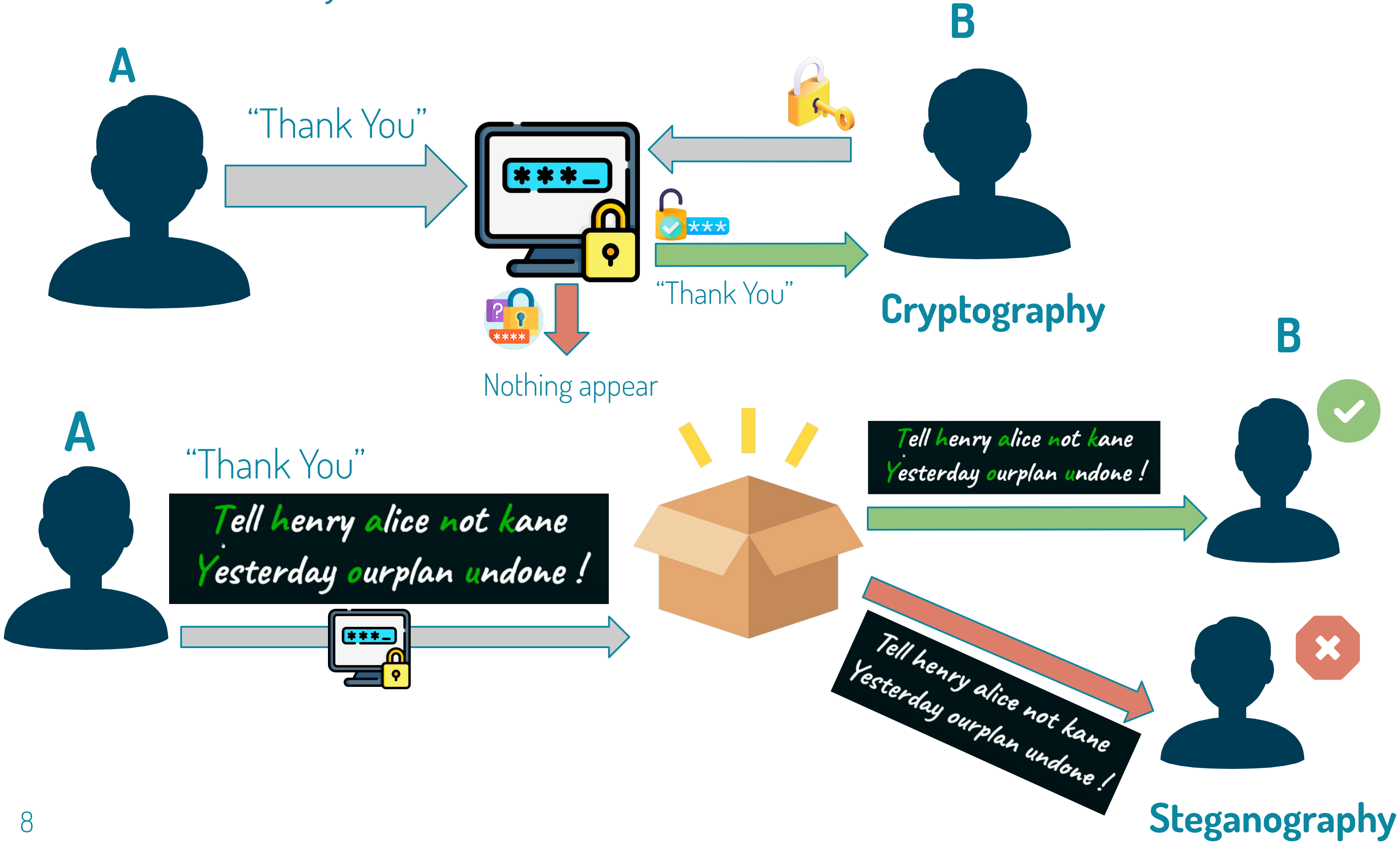
“Thank You”

“Thank You”

“Thank You”

Target

A need to say “**Thank You**” to B



Tell henry alice not kane
Yesterday ourplan undone !

Tell henry alice not kane
Yesterday ourplan undone !

Tell henry alice not kane
Yesterday ourplan undone !

Difference between

Cryptography vs Steganography

Criteria	Cryptography	Steganography
Definition	To **Convert** secret message into other form	To **Hide** the existence of the communication
Purpose	**Secure** communication	**Hidden** Communication
Structure of Data	**Alter** structure data of secret message	**Does not** alter structure data of secret message
Result	**Cipher Text**	**Stego Media**
Discovery	No one can easily get secret data	Anyone can get secret data

Advantages & Drawbacks

Advantages

- has the added benefit of **hiding communications** so well that they **receive no attention**. However, in countries where encryption is illegal, sending an encrypted message that you can easily decipher will raise suspicion and may be risky.
- **protects the information within a message and the connections** between sender and receiver.
- **security, capacity, and robustness**—make it worthwhile to convert information transfer via text files and develop covert communication channels.
- **You can store an encrypted copy** of a file containing sensitive information on the server **without fear of unauthorized parties gaining access** to the data.
- Government and law enforcement agencies can communicate secretly with the help of steganography corporations.

Drawbacks

- **large number of information , Huge file size** (someone may suspected about it)
- If this approach is gone in the wrong hands such as hackers , terrorist then this can be very much critical
- **Ones the system is discovered , It becomes virtually worthless.**
- High overhead is needed

Note :

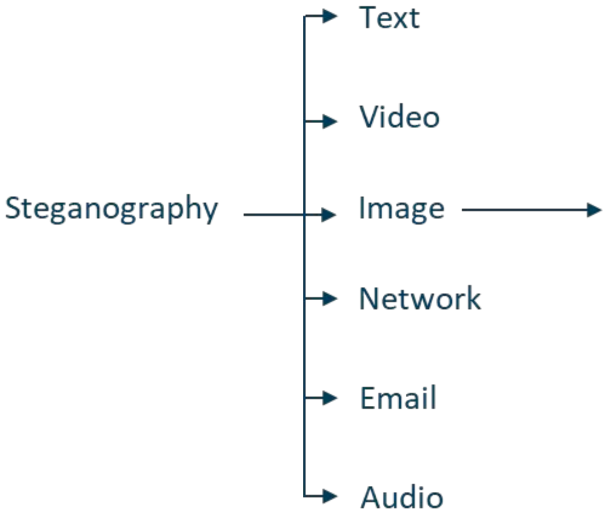
Alternatively , a message can be first encrypted and then hidden using steganography

> *Advantage of using Steganography over Cryptography?*

Advantage of using Steganography over Cryptography?

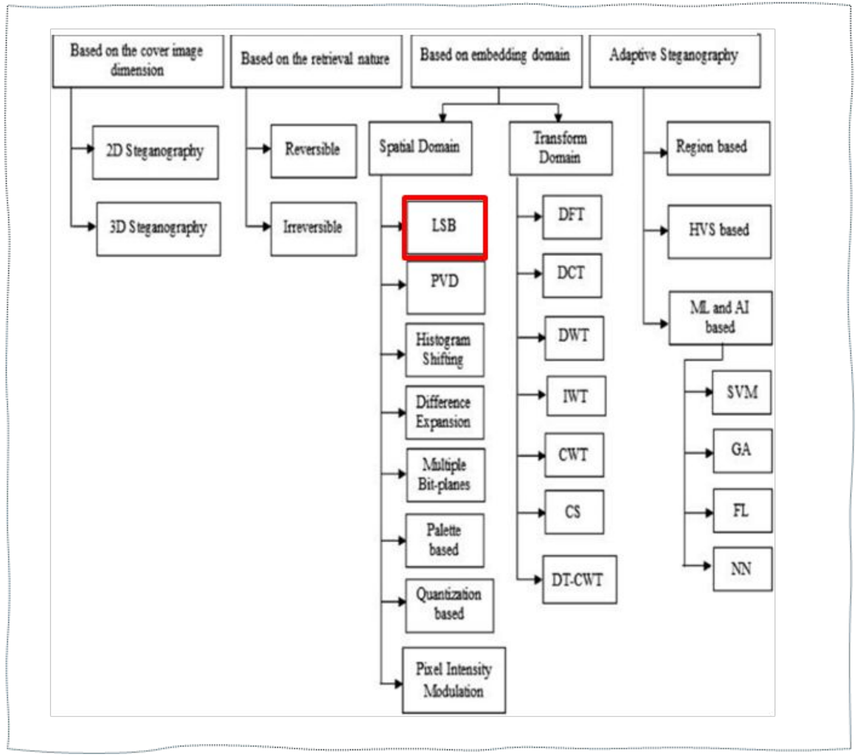
Up to now, cryptography has always had its ultimate role in protecting the secrecy between the sender and the intended receiver. However, nowadays steganography techniques are used increasingly besides cryptography to add more protective layers to the hidden data. **The advantage of using steganography over cryptography alone is that the intended secret message does not attract attention to itself as an object of scrutiny**. Plainly visible encrypted messages, no matter how unbreakable they are, arouse interest and may in themselves be incriminating in countries in which encryption is illegal.

Type of Steganography

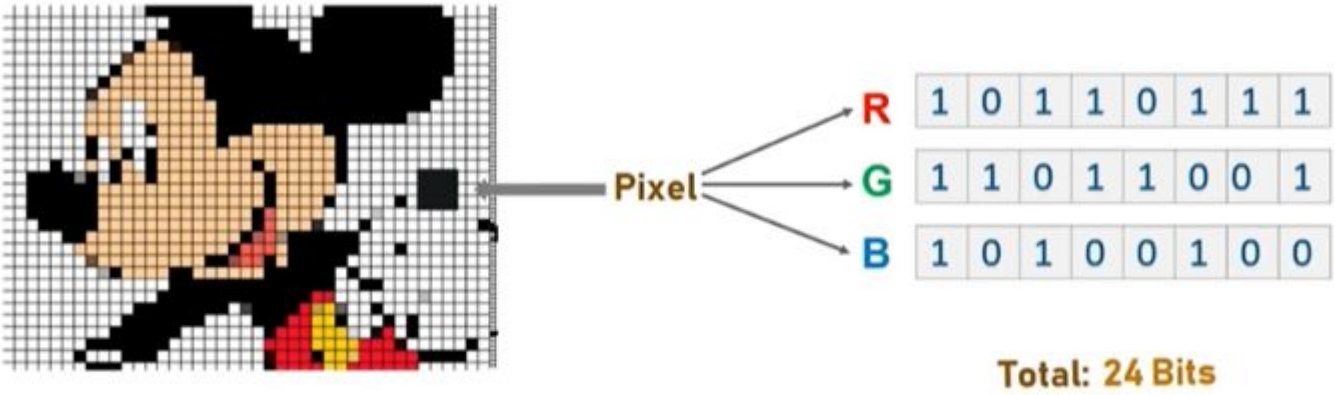


Steganography

- Text
- Video
- Image
- Network
- Email
- Audio



Least Significant Bit Steganography

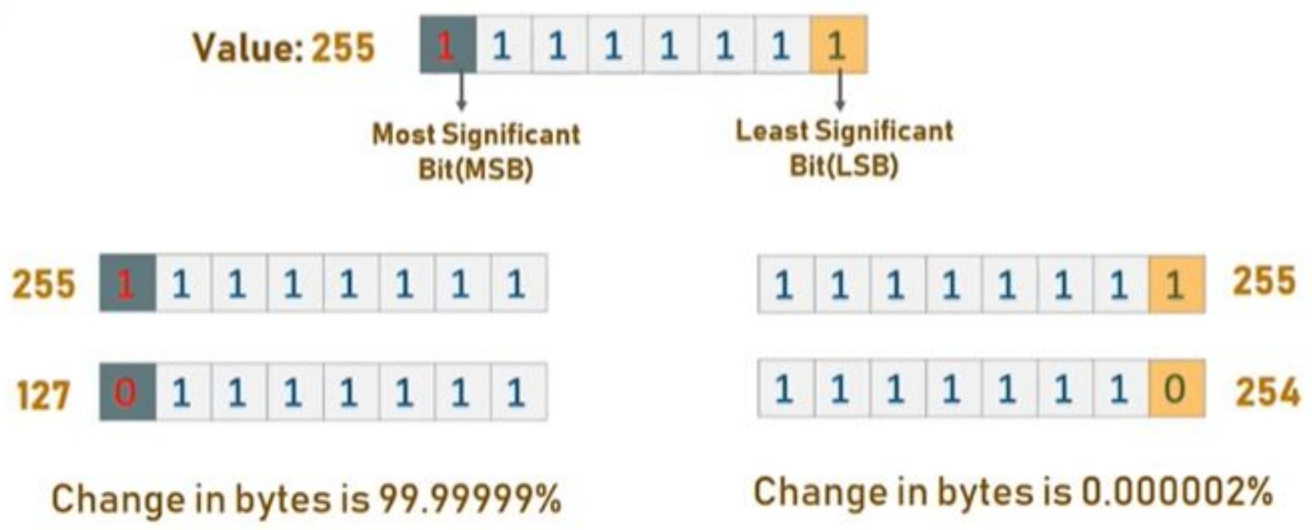


R	1	0	1	1	0	1	1	1
G	1	1	0	1	1	0	0	1
B	1	0	1	0	0	1	0	0

Total: 24 Bits

Photo credits to Edureka Steganography tutorial

Least Significant Bit Steganography



Value: 255 1 1 1 1 1 1 1 1

Most Significant Bit(MSB)

Least Significant Bit(LSB)

255 1 1 1 1 1 1 1 1

127 0 1 1 1 1 1 1 1

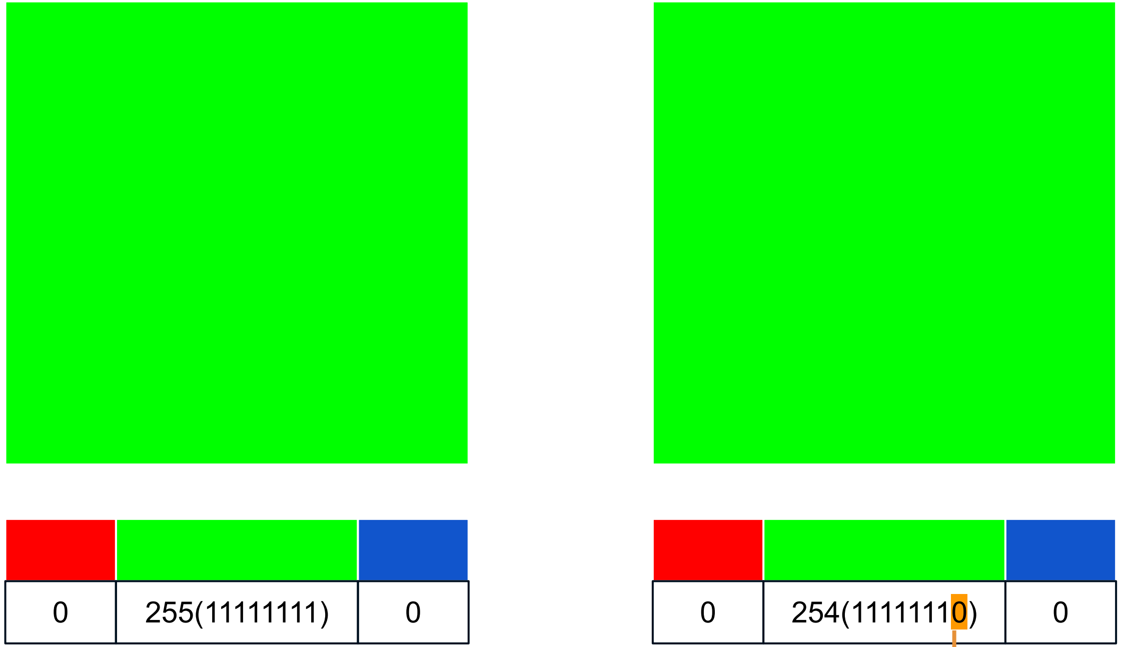
1 1 1 1 1 1 1 1 255

1 1 1 1 1 1 1 0 254

Change in bytes is 99.99999%

Change in bytes is 0.000002%

Photo credits to Edureka Steganography tutorial



🟥	🟩	🟦
0	255(11111111)	0

🟥	🟩	🟦
0	254(11111110)	0

These LSBs can be used for storing the information

How LSB technique of Encode works?

Current Pixel values :

[(225, 12, 99), (155, 2, 50),
(99, 51, 15), (15, 55, 22),
(155, 61, 87), (63, 30, 17),
(1, 55, 19), (99, 81, 66),
(219, 77, 91), (69, 39, 50),
(18, 200, 33), (25, 54, 190)]

How LSB technique of Encode works?

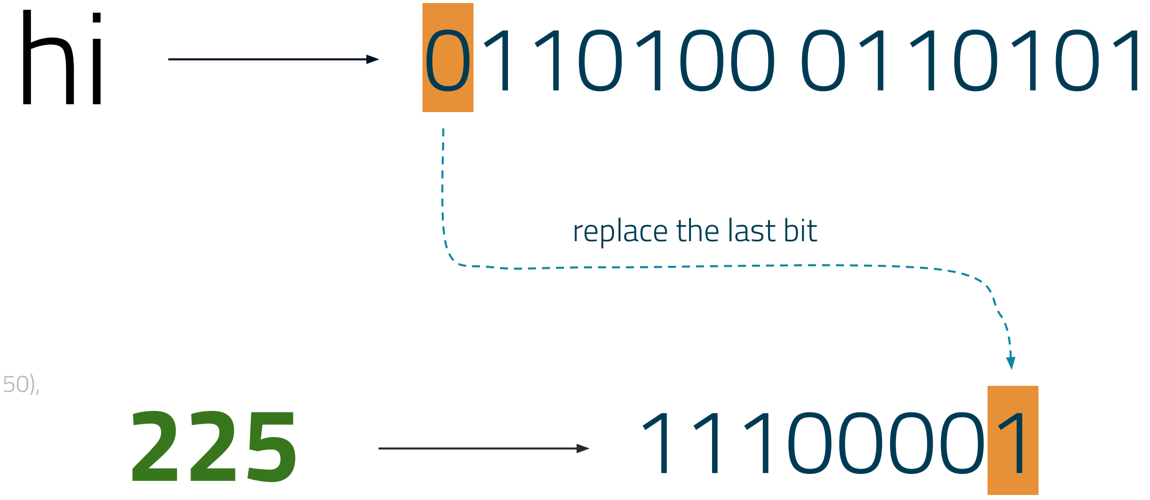
hi → 0110100 0110101

Pixel values:

[(**225**, 12, 99), (155, 2, 50),
(99, 51, 15), (15, 55, 22),
(155, 61, 87), (63, 30, 17),
(1, 55, 19), (99, 81, 66),
(219, 77, 91), (69, 39, 50),
(18, 200, 33), (25, 54, 190)]

225 → 11100001

How LSB technique of Encode works?



hi → 0110100 0110101

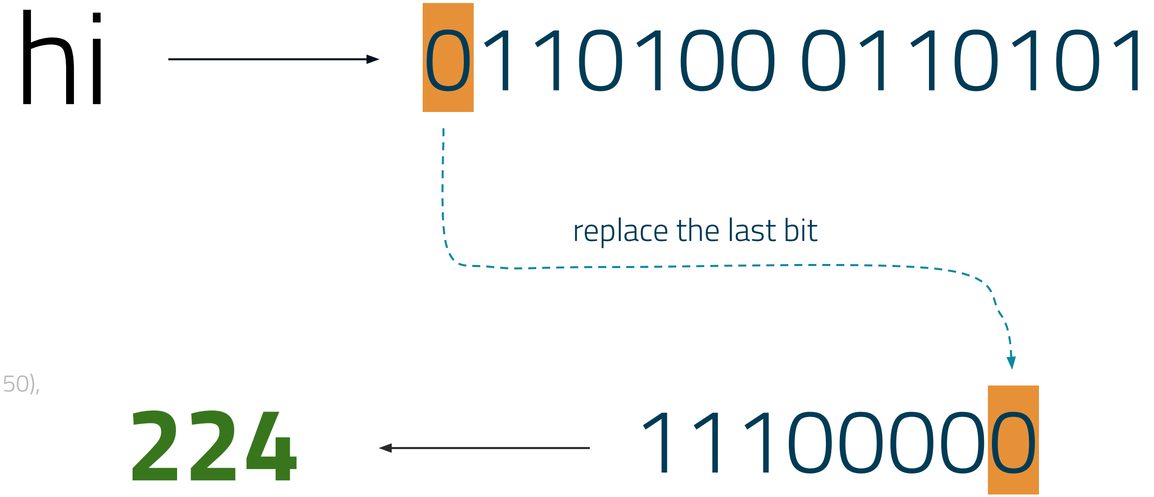
replace the last bit

225 → 11100001

Pixel values:

[(**225**, 12, 99), (155, 2, 50),
(99, 51, 15), (15, 55, 22),
(155, 61, 87), (63, 30, 17),
(1, 55, 19), (99, 81, 66),
(219, 77, 91), (69, 39, 50),
(18, 200, 33), (25, 54, 190)]

How LSB technique of Encode works?



hi → 0110100 0110101

replace the last bit

224 ← 11100000

Pixel values:

[(**224**, 12, 99), (155, 2, 50),
(99, 51, 15), (15, 55, 22),
(155, 61, 87), (63, 30, 17),
(1, 55, 19), (99, 81, 66),
(219, 77, 91), (69, 39, 50),
(18, 200, 33), (25, 54, 190)]

How LSB technique of Encode works?

Current Pixel values :

[(**225**, **12**, 99), (**155**, **2**, 50),
(**99**, **51**, 15), (15, **55**, **22**),
(**155**, 61, 87), (63, 30, 17),
(1, 55, 19), (99, 81, 66),
(219, 77, 91), (69, 39, 50),
(18, 200, 33), (25, 54, 190)]

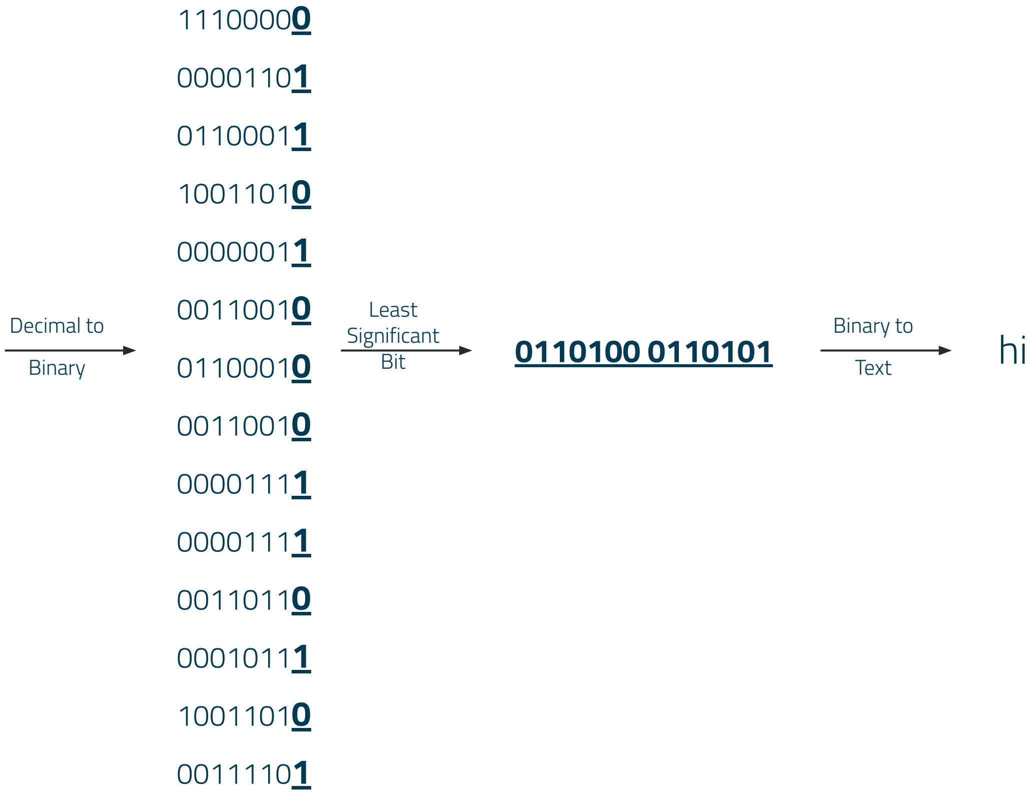
New Pixel values :

[(**224**, **13**, 99),(**154**, **3**, 50),
(**98**, **50**, 15),(15, **54**, **23**),
(**154**, 61, 87),(63, 30, 17),
(1, 55, 19),(99, 81, 66),
(219, 77, 91),(69, 39, 50),
(18, 200, 33),(25, 54, 190)]

How LSB technique of Decode works?

New Pixel values :

[(**224**, **13**, 99),(**154**, **3**, 50),
(**98**, **50**, 15),(15, **54**, **23**),
(**154**, 61, 87),(63, 30, 17),
(1, 55, 19),(99, 81, 66),
(219, 77, 91),(69, 39, 50),
(18, 200, 33),(25, 54, 190)]



Decicmal to Binary

1110000**0**
0000110**1**
0110001**1**
1001101**0**
0000001**1**
0011001**0**
0110001**0**
0011001**0**
0000111**1**
0000111**1**
0011011**0**
0001011**1**
1001101**0**
0011110**1**

Least Significant Bit

0110100 0110101

Binary to Text

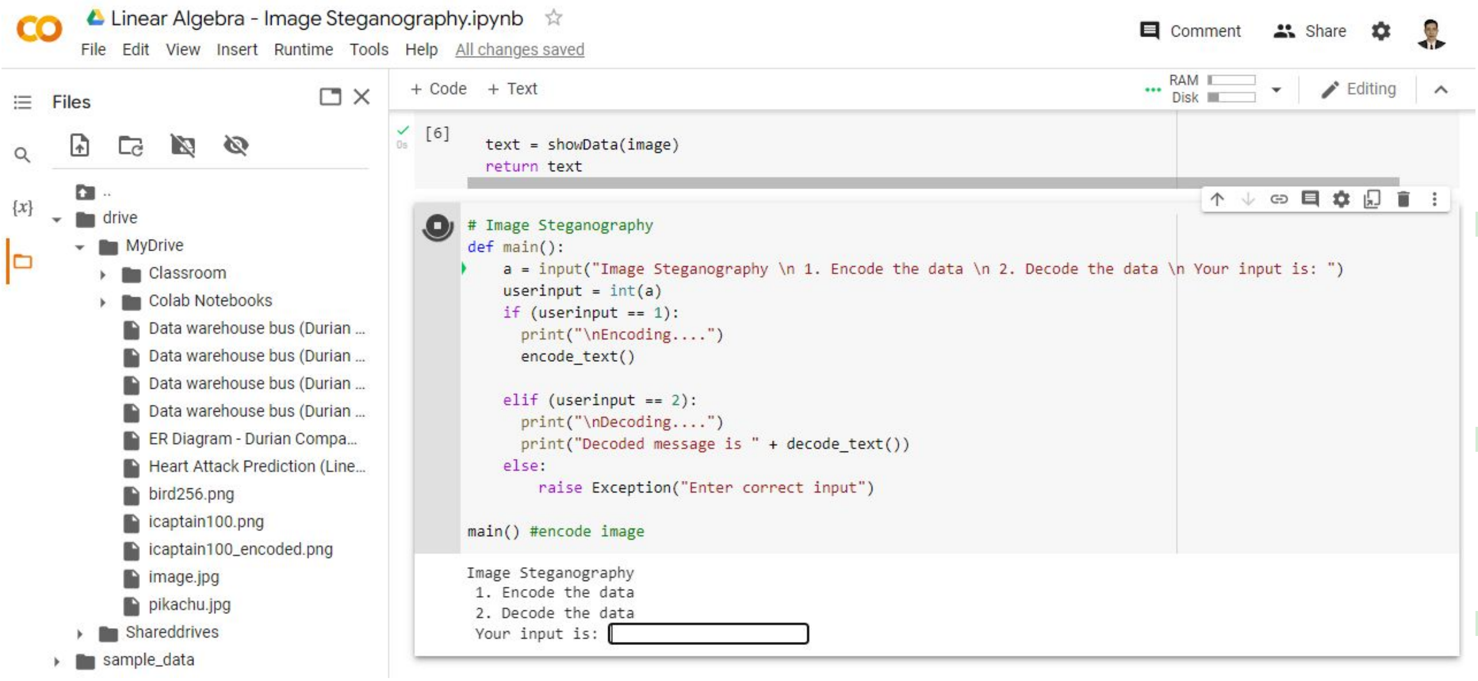
hi

Hiding text inside an image using Python

In this section, we can find a step-by-step of the hide and reveal process using Python code. Open a google collab notebook and follow the steps below:

Before beginning with the code, you can upload the image(png) that you would like to use for steganography using the upload option that appears on the left hand side menu bar.

Hiding text inside an image using Python

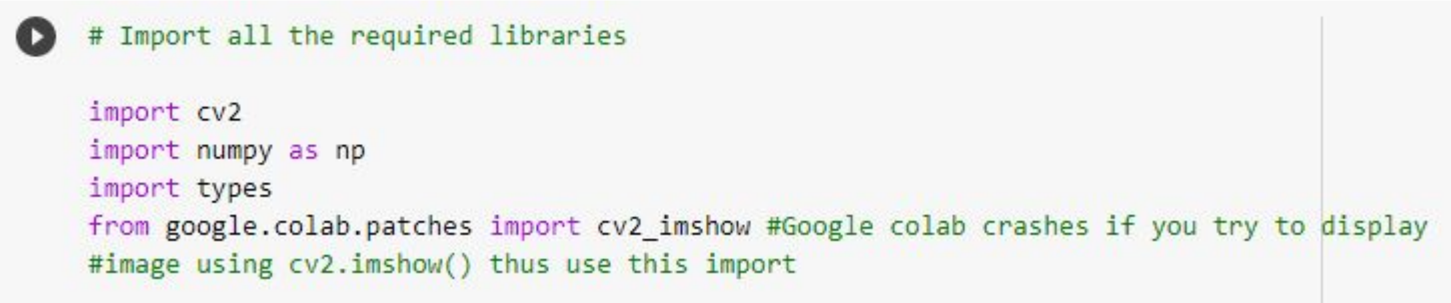

Linear Algebra - Image Steganography.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
Editing
..
drive
MyDrive
Classroom
Colab Notebooks
Data warehouse bus (Durian ...
Data warehouse bus (Durian ...
Data warehouse bus (Durian ...
Data warehouse bus (Durian ...
ER Diagram - Durian Compa...
Heart Attack Prediction (Line...
bird256.png
icaptain100.png
icaptain100_encoded.png
image.jpg
pikachu.jpg
Shareddrives
sample_data
[6]
0s

```python
  text = showData(image)
  return text
```

```python
# Image Steganography
def main():
    a = input("Image Steganography \n 1. Encode the data \n 2. Decode the data \n Your input is: ")
    userinput = int(a)
    if (userinput == 1):
      print("\nEncoding....")
      encode_text()

    elif (userinput == 2):
      print("\nDecoding....")
      print("Decoded message is " + decode_text())
    else:
        raise Exception("Enter correct input")

main() #encode image
```

```
Image Steganography
 1. Encode the data
 2. Decode the data
 Your input is: 
```

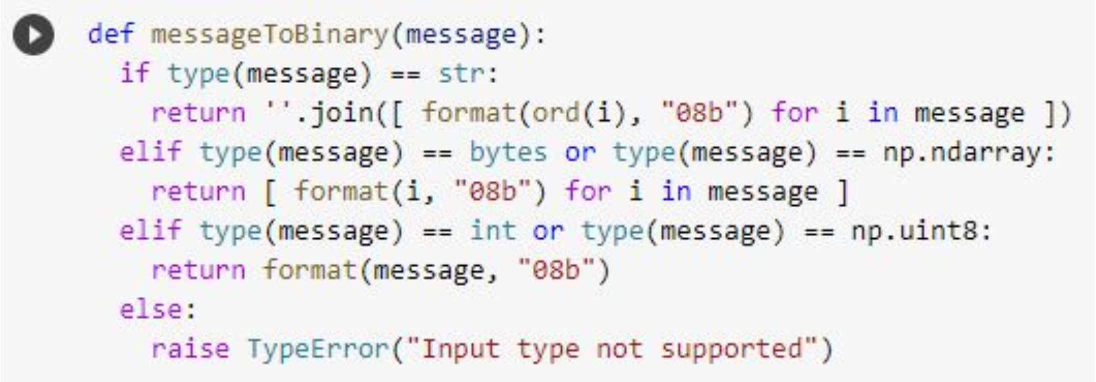
Hiding text inside an image using Python



```
# Import all the required libraries

import cv2
import numpy as np
import types
from google.colab.patches import cv2_imshow #Google colab crashes if you try to display
#image using cv2.imshow() thus use this import
```

Step 1: Import all the required python libraries

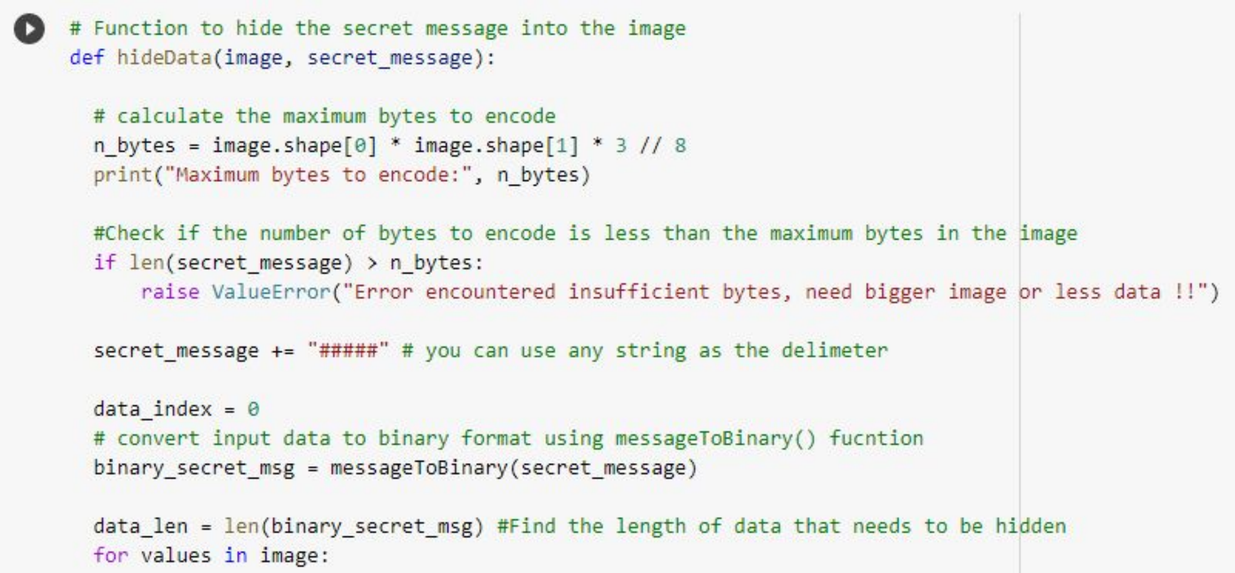
Hiding text inside an image using Python



```python
def messageToBinary(message):
  if type(message) == str:
    return ''.join([ format(ord(i), "08b") for i in message ])
  elif type(message) == bytes or type(message) == np.ndarray:
    return [ format(i, "08b") for i in message ]
  elif type(message) == int or type(message) == np.uint8:
    return format(message, "08b")
  else:
    raise TypeError("Input type not supported")
```

Step 2: Define a function to convert any type of data into binary, we will use this to convert the secret data and pixel values to binary in the encoding and decoding phase.

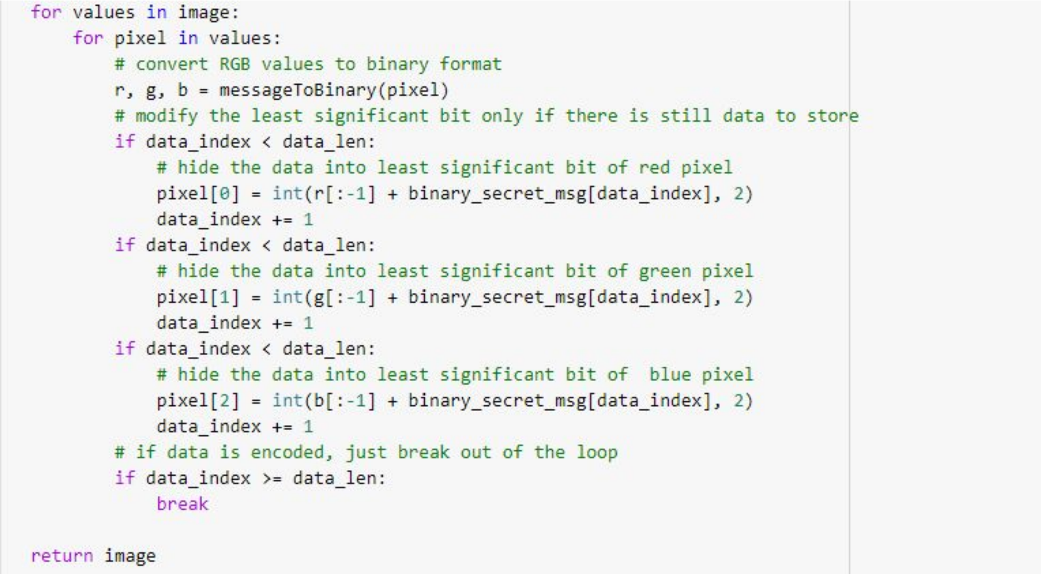
Hiding text inside an image using Python



```
# Function to hide the secret message into the image
def hideData(image, secret_message):

  # calculate the maximum bytes to encode
  n_bytes = image.shape[0] * image.shape[1] * 3 // 8
  print("Maximum bytes to encode:", n_bytes)

  #Check if the number of bytes to encode is less than the maximum bytes in the image
  if len(secret_message) > n_bytes:
      raise ValueError("Error encountered insufficient bytes, need bigger image or less data !!")

  secret_message += "#####" # you can use any string as the delimeter

  data_index = 0
  # convert input data to binary format using messageToBinary() fucntion
  binary_secret_msg = messageToBinary(secret_message)

  data_len = len(binary_secret_msg) #Find the length of data that needs to be hidden
  for values in image:
```

Step 3: Write a function to hide secret message into the image by altering the LSB

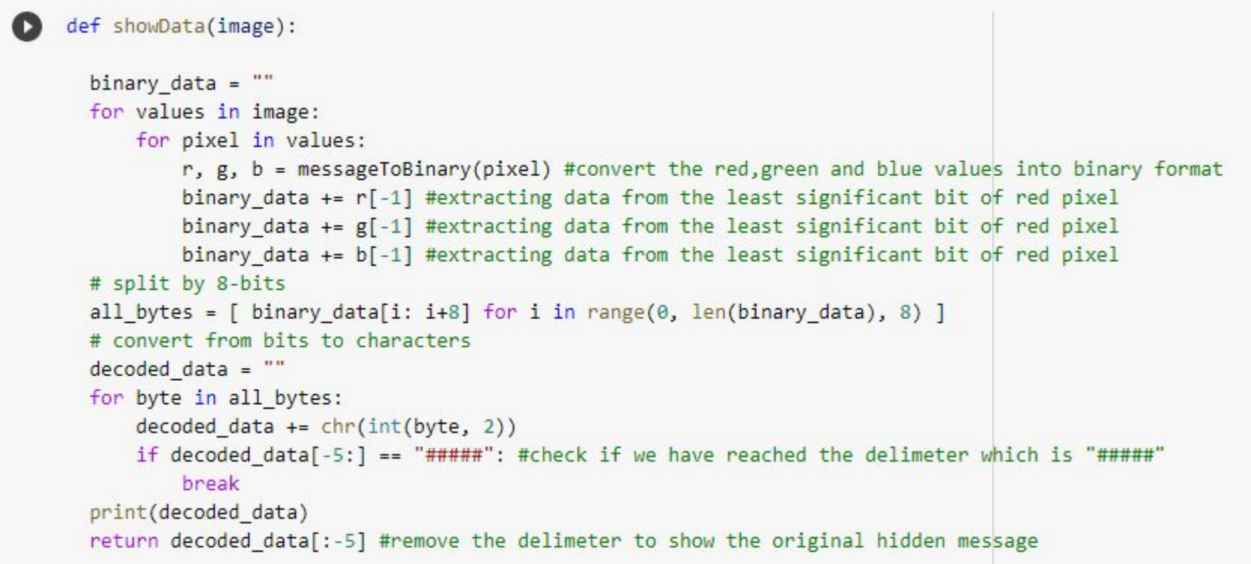
Hiding text inside an image using Python



```
for values in image:
    for pixel in values:
        # convert RGB values to binary format
        r, g, b = messageToBinary(pixel)
        # modify the least significant bit only if there is still data to store
        if data_index < data_len:
            # hide the data into least significant bit of red pixel
            pixel[0] = int(r[:-1] + binary_secret_msg[data_index], 2)
            data_index += 1
        if data_index < data_len:
            # hide the data into least significant bit of green pixel
            pixel[1] = int(g[:-1] + binary_secret_msg[data_index], 2)
            data_index += 1
        if data_index < data_len:
            # hide the data into least significant bit of  blue pixel
            pixel[2] = int(b[:-1] + binary_secret_msg[data_index], 2)
            data_index += 1
        # if data is encoded, just break out of the loop
        if data_index >= data_len:
            break

return image
```

Step 3: Write a function to hide secret message into the image by altering the LSB

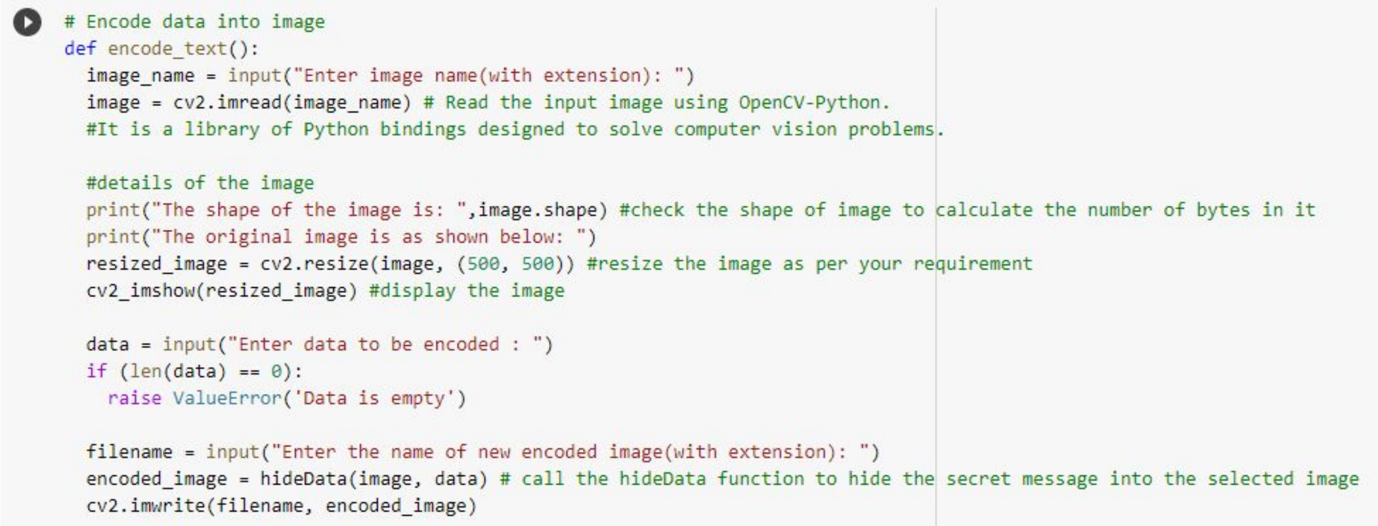
Hiding text inside an image using Python



```
def showData(image):

    binary_data = ""
    for values in image:
        for pixel in values:
            r, g, b = messageToBinary(pixel) #convert the red,green and blue values into binary format
            binary_data += r[-1] #extracting data from the least significant bit of red pixel
            binary_data += g[-1] #extracting data from the least significant bit of red pixel
            binary_data += b[-1] #extracting data from the least significant bit of red pixel
    # split by 8-bits
    all_bytes = [ binary_data[i: i+8] for i in range(0, len(binary_data), 8) ]
    # convert from bits to characters
    decoded_data = ""
    for byte in all_bytes:
        decoded_data += chr(int(byte, 2))
        if decoded_data[-5:] == "#####": #check if we have reached the delimeter which is "#####"
            break
    print(decoded_data)
    return decoded_data[:-5] #remove the delimeter to show the original hidden message
```

Step 4: Define a function to decode the hidden message from the stego image

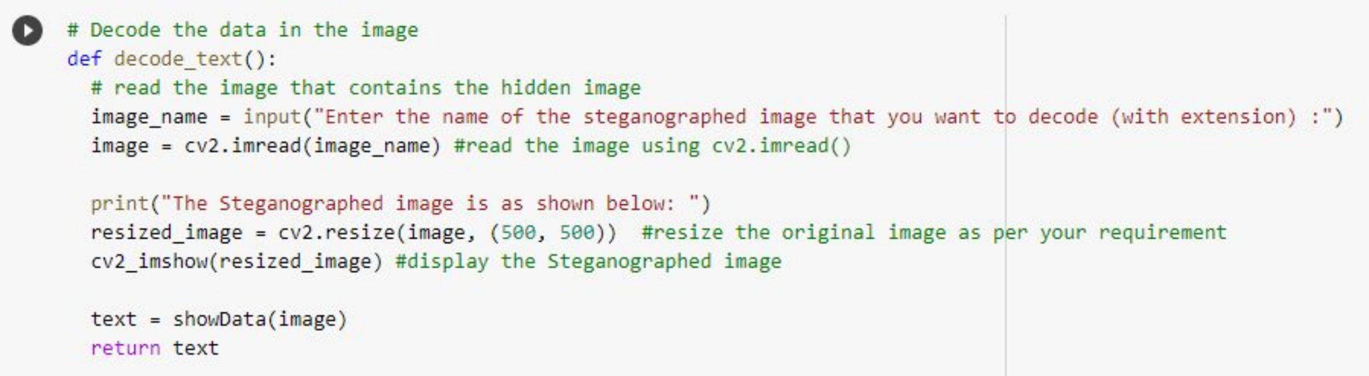
Hiding text inside an image using Python



```
# Encode data into image
def encode_text():
  image_name = input("Enter image name(with extension): ")
  image = cv2.imread(image_name) # Read the input image using OpenCV-Python.
  #It is a library of Python bindings designed to solve computer vision problems.

  #details of the image
  print("The shape of the image is: ",image.shape) #check the shape of image to calculate the number of bytes in it
  print("The original image is as shown below: ")
  resized_image = cv2.resize(image, (500, 500)) #resize the image as per your requirement
  cv2_imshow(resized_image) #display the image

  data = input("Enter data to be encoded : ")
  if (len(data) == 0):
    raise ValueError('Data is empty')

  filename = input("Enter the name of new encoded image(with extension): ")
  encoded_image = hideData(image, data) # call the hideData function to hide the secret message into the selected image
  cv2.imwrite(filename, encoded_image)
```

Step 5: Function that takes the input image name and secret message as input from user and calls hideData() to encode the message

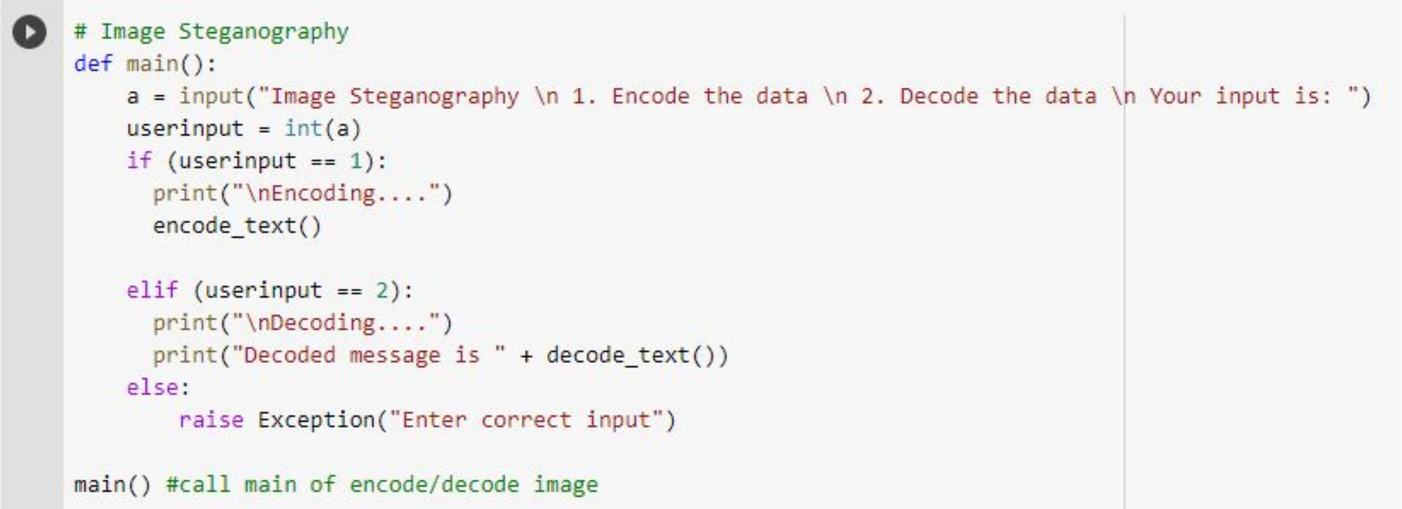
Hiding text inside an image using Python



```
# Decode the data in the image
def decode_text():
  # read the image that contains the hidden image
  image_name = input("Enter the name of the steganographed image that you want to decode (with extension) :")
  image = cv2.imread(image_name) #read the image using cv2.imread()

  print("The Steganographed image is as shown below: ")
  resized_image = cv2.resize(image, (500, 500))  #resize the original image as per your requirement
  cv2_imshow(resized_image) #display the Steganographed image

  text = showData(image)
  return text
```

Step 6: Create a function to ask user to enter the name of the image that needs to be decoded and call the showData() function to return the decoded message.

Hiding text inside an image using Python

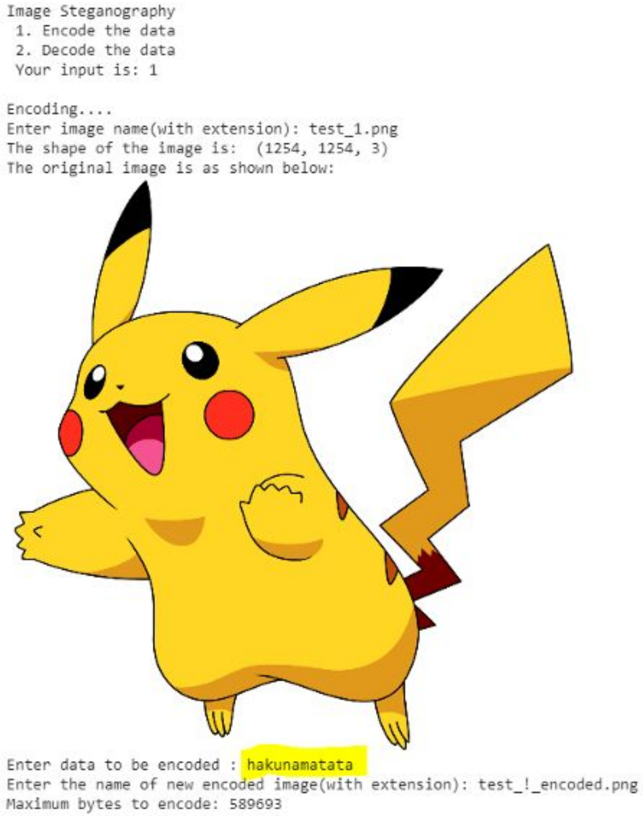


```
# Image Steganography
def main():
    a = input("Image Steganography \n 1. Encode the data \n 2. Decode the data \n Your input is: ")
    userinput = int(a)
    if (userinput == 1):
      print("\nEncoding....")
      encode_text()

    elif (userinput == 2):
      print("\nDecoding....")
      print("Decoded message is " + decode_text())
    else:
        raise Exception("Enter correct input")

main() #call main of encode/decode image
```

Step 7: Main Function()

Output/Results:

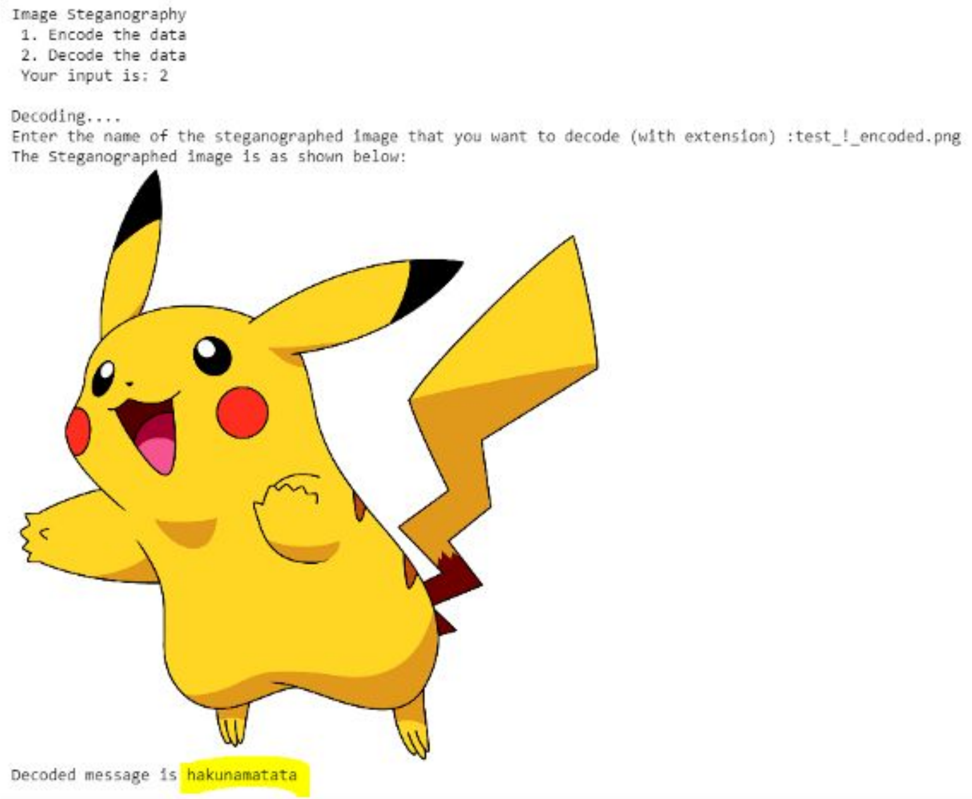
Encoding the message:



```
Image Steganography
 1. Encode the data
 2. Decode the data
 Your input is: 1

Encoding....
Enter image name(with extension): test_1.png
The shape of the image is:  (1254, 1254, 3)
The original image is as shown below:

Enter data to be encoded : hakunamatata
Enter the name of new encoded image(with extension): test_!_encoded.png
Maximum bytes to encode: 589693
```

Output/Results:

Decoding the message:

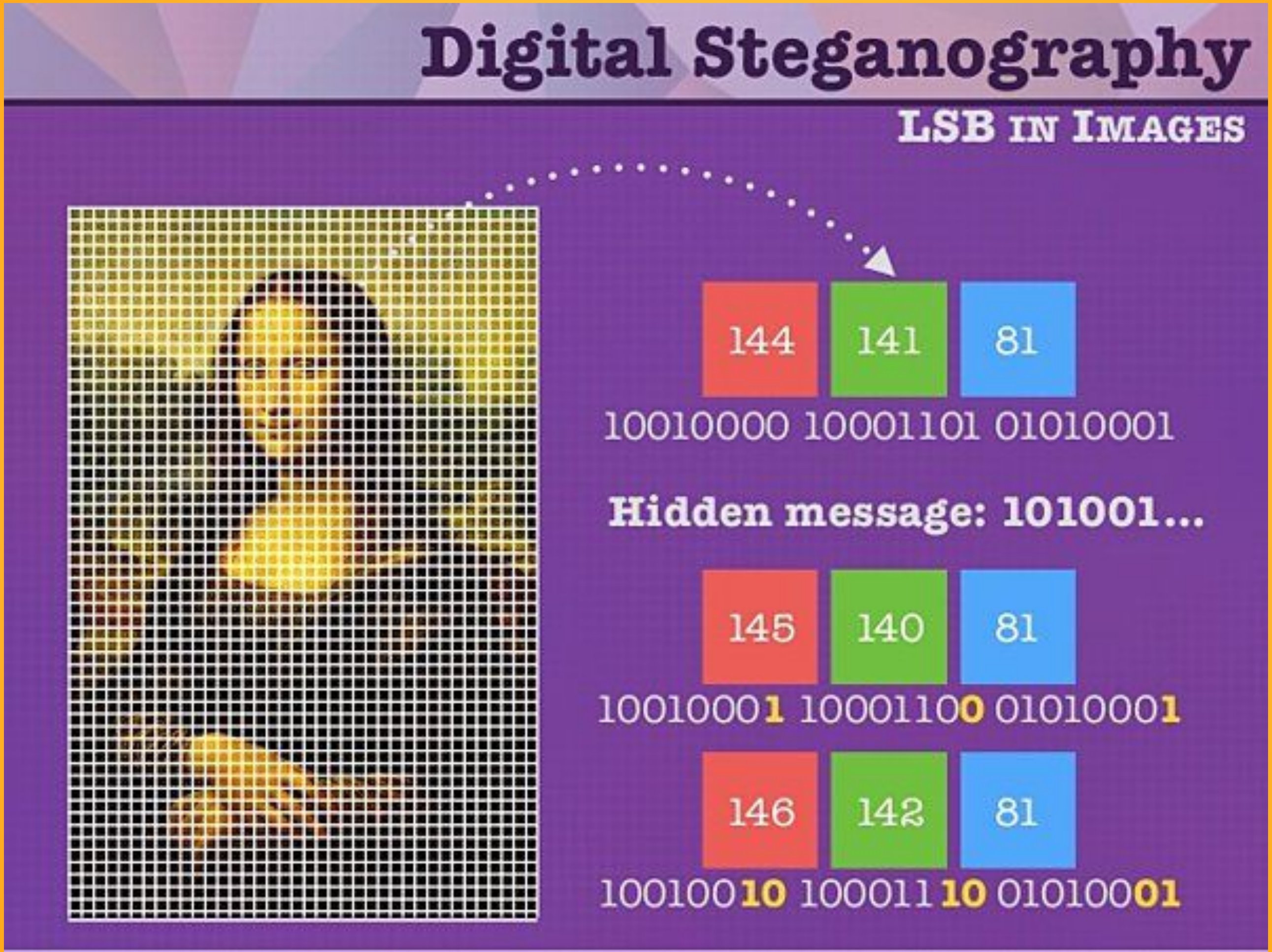


```
Image Steganography
 1. Encode the data
 2. Decode the data
 Your input is: 2

Decoding....
Enter the name of the steganographed image that you want to decode (with extension) :test_!_encoded.png
The Steganographed image is as shown below:

Decoded message is hakunamatata
```

References

“ References:

- https://towardsdatascience.com/steganography-hiding-an-image-inside-another-77ca66b2acb1
- https://www.edureka.co/blog/steganography-tutorial
- https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#191d0b0160ba
- https://www.ukessays.com/essays/computer-science/steganography-uses-methods-tools-3250.php
- https://www.thepythoncode.com/article/hide-secret-data-in-images-using-steganography-python
- https://www.youtube.com/watch?v=xepNoHgNj0w&t=1922s



Digital Steganography

LSB in Images

144 141 81

10010000 10001101 01010001

Hidden message: 101001...

145 140 81

1001000**1** 1000110**0** 0101000**1**

146 142 81

100100**10** 100011**10** 010100**01**