

PIXELS TO PICTURES

A PROGRAMMING COURSE ON IMAGES WITH MATLAB

Instructor Guide

Module 12: Encipher and Decipher Text

Prerequisite Domain Knowledge: *Text on Images*,

Expected Completion Time: 50 minutes

Table of Contents

Types of Ciphers.....	1
Learning Objectives.....	1
Materials.....	1
Steps.....	1
Part I: Swap Cipher.....	2
Part II: Atbash Cipher.....	3
Part III: Caesar Cipher.....	4
Cipher Messages.....	5
Learning Objectives.....	5
Materials.....	5
Steps.....	5

Types of Ciphers

Expected Duration: 30 minutes

Learning Objectives

- Understand what a cipher is
- Understand three different ways to cipher and decipher text in MATLAB

Materials

- MATLAB®
- Demo PDF "CiphersDemo"

Steps

Tell students that before they start enciphering and deciphering text let's make a list of what we know at this point.

Ask:

- *What we have learned about writing text on images so far?*
- You can write on any image.
- You can change font size.
- You can change the font color.

Now we will learn how we can write secret messages using ciphers!

Ask:

- *What is a cipher?*
- A cipher is a form of *secret communication* where the letters in the message are rearranged or substituted to disguise the original message. This process is called **enciphering**.
- *How might enciphering a message be useful?*
- In ancient times when messages were carried by foot for miles, kings and rulers would encipher the letters they would send to allies. This helped to protect the secrecy of the message in case they were stolen.

Before we can start writing secret messages in images, lets learn about a few ciphers and how to write the secret messages.

Ask the students to open the following script:

```
open 'CipherMessages.mlx'
```

Part I: Swap Cipher

Lets start with the Swap Cipher. Have the students run the first section of "CipherMessages.mlx" and observe the result:

Cipher Messages

Clean up

```
clc  
clear all
```

Swap Cipher

```
secret = SwapCipher('hello there')
```

```
secret = 'olleh ereht'
```

```
% secret2 = SwapCipher
```

Ask:

- *What did the SwapCipher function do?*
- Students should notice that the cipher swapped the letters in each word:
- hello → olleh
- there → ereht

It's a simple trick but makes the secret message a lot harder to read. Ask the students to work in pairs and one of them should use the SwapCipher function to create a longer secret message while the other student is looking away.

The other student can now look and try to decipher the message.

The enciphered message should be pretty challenging to read. Tell them that because all the function is doing is reversing the order of the letters, they can use the SwapCipher function again to decipher the secret message and verify they read the secret correctly!

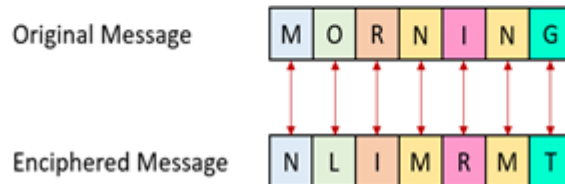
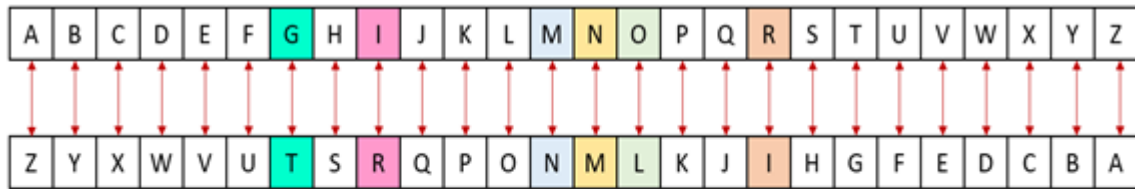
Students can take turns enciphering and deciphering the messages.

Open the **CiphersDemo** Demo PDF and show it to the students:

```
web('demos/CiphersDemo.pdf', '-browser');
```

Part II: Atbash Cipher

The Atbash cipher is another simple but clever cipher. Here the letters of the alphabet are reversed – A gets replaced with Z, B gets replaced with Y, C gets replaced with X, and so on.



Turn attention to the "Atbash Cipher" section in "CipherMessages.mlx". Have the students run the section and observe the result:

```

10
11
12
13
14
15
16
Atbash Cipher

x = AtbashCipher('ABCDEF')

% secret3 = '';
%
% y = AtbashCipher(secret3)
%
% z = AtbashCipher(secret3)

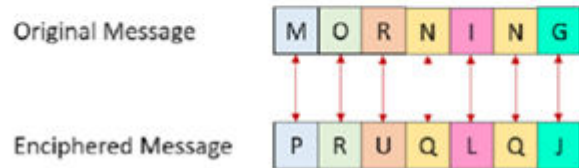
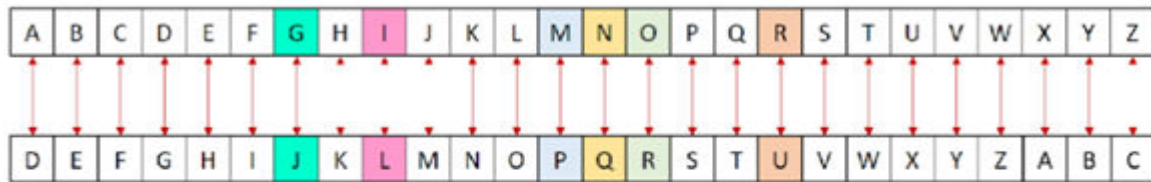
```

Ask the students to try using the Atbash Cipher on a few words and check it is giving the expected answer.

Similar to the Swap Cipher, students can use the AtbashCipher again to decode the message back to its original form.

Part III: Caesar Cipher

For the Caesar Cipher, letters of the alphabet are shifted by a certain number. In this course we are shifting the letters by 3. So A becomes D, B becomes E, and so on:



Repeat the exercise in "CipherMessages.mlx" by asking the students to try out the CaesarCipher and CaesarDecipher functions to create secret messages.

```

17  a = CaesarCipher('this is an encrypted message')

18
19  a = 'wklv lv dq hqfubswhg phvvdjh'
20
21  % secret4 = '';
22  %
23  % b = CaesarCipher(secret4)
24  %
25  % b = CaesarDecipher(secret4)

```

Cipher Messages

Expected Duration: 20 minutes

Learning Objectives

-

Materials

- MATLAB®
- Worksheet "Cipher Messages"

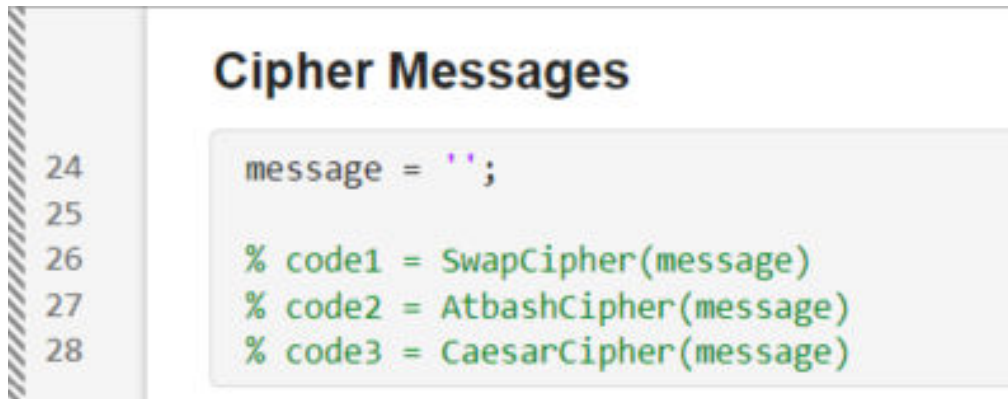
Steps

Distribute the **Cipher Messages** Worksheet among the students.

```
web('worksheets_and_handouts/CipherMessages.pdf', '-browser');
```

Explain to the students that they need to work in pairs for this activity.

The first student should think of a question to ask his/her friend in class and can use any one of the three ciphers in MATLAB to encipher their message. Students can use the last section of "CipherMessages.mlx" to achieve this:



```
>> secret = CaesarCipher('Who is your favorite singer?')  
secret =  
Zkr lv brxu idyrulwh vlqjhu?
```

They can then copy the enciphered message onto the worksheet and pass it to their friend.

1. Think of a question to ask a friend in class
2. Encode your question in MATLAB using one of the ciphers
3. Write down your encoded question here

Zkr lv brxu idyrulwh vlqjhu?

4. Pass the sheet to your friend

The second student then needs to use the deciphering functions to decipher and read the question.

```
>> message = CaesarDecipher('Zkr lv brxu idyrulwh vlqjhu?')  
message =  
Who is your favorite singer?
```

Then they can reply to the question by using any one of the ciphers (same or different one) to encipher their response and return it the first student.

```
>> secret = AtbashCipher('Taylor Swift')  
secret =  
Gzboli Hdrug
```

1. Use one of the ciphers in MATLAB to decode the question your friend has sent you
2. Encode your answer to the question using one of the ciphers
3. Write down your encoded response here

Gzboli Hdrug

4. Pass the sheet back to your friend!

The first student then needs to decipher the response using the Live Script.

This activity should give both of them a chance to encipher and decipher secret messages.

When time has expired, gather students.

Ask:

1. *What did you enjoy most about this activity?*
2. Accept all responses from students.
3. *What did you find challenging about this activity?*
4. Accept all responses from students.