Computer Science Topics

University of Alberta

Cryptography

Caesar Cipher

Created by Julius Caesar in 100 B.C., this cipher was created to relay military information in secret

With the caesar cipher, you pick a number (between 1 and 25) to shift each letter by.

This makes the message very difficult to read to those who don't know that the message has been encrypted, but very easy to decode and read for those who do.

Example Caesar Cipher

1. Come up with your message
   Attack at dawn
2. Decide how many letters to shift by
   1
3. Shift each letter by the amount you chose in step 3
   A becomes **B**
   T becomes **U**
   T becomes **U**
   A becomes **B**
   C becomes **D**
   K becomes **L**
   : **Attack**, shifted by 1, becomes **Buubdl**
   : Repeat this process for each word in the message
4. Done!
   Attack at dawn becomes **Buubdl bu ebxo**

   This encrypted message is difficult to understand if you don’t know that the message was encrypted and the method used to encrypt it

Example code

1. Get the message from the user and convert it into lowercase (for simplicity)
2. Initialize a variable for the encrypted text
3. Start a for loop that checks every character in the message
4. If the character is a letter, shift it and add it to the encrypted text variable

```
plaintext = input(“Message: ”).lower()
ciphertext = “”
val = 5

for char in plaintext:
     ciphertext += shift(char, val)

print(ciphertext)
```

Substitution Cipher

Background info

With a substitution cipher, you replace each letter with a different letter of the alphabet. These letters can be in order, but don’t have to be.

Example Substitution

```
Alphabet: ABCDEFGHIJKLMNOPQUSTUVWXYZ
Key:      QWERTYUIOPASDFGHJKLZXCVBNM
```

So if we were to use this key for our substitution cipher:

- A becomes Q
- B becomes W
- C becomes E
- D becomes R
- E becomes T

  . . .

Example Substitution

```
Alphabet: ABCDEFGHIJKLMNOPQUSTUVWXYZ
Key:      NOPQUSTUVWXYZABCDEFGHIJKLM
```

So if we were to use this key for our substitution cipher:

- A becomes N
- B becomes O
- C becomes P
- D becomes Q
- E becomes R

  . . .

Pseudocode

1. Set up your alphabet and key
2. Initialize a variable for the encrypted text
3. Start a for loop that checks every character in the message
4. If the character is a letter, make the right substitution according to the key and add that to the encrypted text variable

```
alphabet = “abcdefghijklmnopqrstuvwxyz”
key = “qwertyuiopasdfghjklzxcvbnm”
ciphertext = “”

for char in plaintext:
     if char.isalpha():
          char += substitution(alpha, key, char)

print(ciphertext)
```

Cryptography Project

https://bit.ly/2KOv3za

Your tasks for this period are to

1. Create a caesar cipher
2. Create a substitution cipher

The specifics to each task are located in the Github link to the left of this text box

File IO

File Types

Although there are many types of files in computer science, we'll be working with one type of file in this section

.csv (comma separated values)

CSV files are commonly used in spreadsheets, because they are a very simple and universally-accepted way of storing (non-sensitive) data.

Each value is separated by a comma, which allows for information to be stored in a very organized manner which makes using the data very simple

Files in Python

How do we use these files while programming in Python?

In Python, we can import files. When imported, Python treats those files as <file objects> and those include methods that allow us to turn them into strings.

What can we do with files in Python?

File objects have many important functions and methods, but these 3 are what you will use the most often

```
file = open(“data.txt”)
file.readlines()
file.close()
```

open() is a function that takes a filename as a parameter and returns a file object (think of it as turning on a light).

.readlines() is a method that reads each line in a file and returns a list of every line in the file

.close() closes the file (think of it as turning off the light). You need to make sure that you always close the file after you’re done with it.

File Project

https://bit.ly/2m6eGz9

Your tasks for this period are to

1. Download both the data.txt and the students.py file in the repository
2. Complete all of the functions defined in the students.py file

The specifics to each task are located in the Github link to the left of this text box

Imaging

Images

How do computers recognize Images?

There are a few properties that go into a computer image. Here are a few of those properties:

1. File type
   .jpg .png .gif .raw
2. Resolution
   1280x720p 1920x1080p 2160x1440p
3. Pixel properties
   RGB CMYK Hexadecimal

File types

JPEG (.jpg) files shrink the size of your image by aiming to remove details of the image that you won't notice. This method saves storage space, but results in lower quality images.

PNG (.png) files stand for "portable network graphics". These files are not only compressed without loss (the quality of the image doesn't get worse), but they also support transparency in images, which is very important to people such as designers. However, they usually take up a little more storage space.

GIF (.gif) files can show both regular photos and videos without audio, but PNGs have replaced their use in photos. GIFs are very lossy (meaning the quality will be very bad), but they use very low amounts of storage space and are pretty common today.

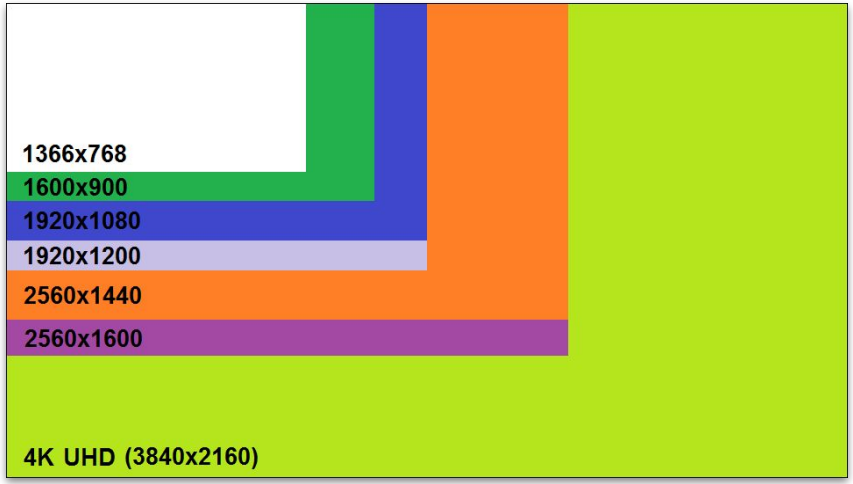
RAW (.raw) files are processed directly from the sensors of digital cameras. This makes them extremely large files, but they retain very high quality photos and any changes to them only alter the metadata and not the actual raw file, meaning the original image is never altered.

Resolution

Image resolutions determine how many pixels are contained in an image, and how tall or wide the image will be. This is the biggest determining factor in an image's overall quality.

Common resolutions used in today's world are:

720p (1280x720p)
1080p (1920x1080p)
1440p (2560x1440p)
4K (3840x2160p)



Color Modes

RGB, CMYK, and hexadecimal

Each pixel is coded as a color of a specific value. This value can either be an RGB, CMYK, or hexadecimal value.

RGB = Red(0-255) Green(0-255) Blue(0-255)
Each number in an RGB value represents the intensity of that color. For example (255, 255, 0) would be maximum intensity red and green, with no blue

Hexadecimal = #000000 to #FFFFFF
In hexadecimal, the first, second, and last pair characters represent the three numbers that go into an RGB value (don't worry about understanding this part if you can't figure it out, just know that they work the same as RGB but look a little differently to write out)

CMYK = (0-100%, 0-100%, 0-100%, 0-100%)
This model is generally used for printing. We won't be using this color model, but included it to show that there are other, less common (but still useful) models

cImage

Using the cImage library in Python

Python has a library available to it called **cImage**. We will be using this library to manipulate images in Python.

A quick guide to using cImage will be included in the Github link in the following slide.

Imaging Project

https://github.com/UofAScienceCamps2018/CS-Topics/tree/master/imaging

Your tasks for this period are to

1. Create a function that turns a pixel blue
2. Create a function that turns a pixel into a negative pixel
3. Create a function that turns a pixel into a grayscale pixel

The specifics to each task are located in the Github link to the left of this text box