

# Homework Assignment

**Andreas Vikke & Martin Frederiksen**



10. februar 2021  
Professionsbachelor i softwareudvikling  
Cphbusiness Lyngby  
Denmark

# 1. Description

In order to look at self reflection and to judge your assessment of information, you should solve the programming exercise below.

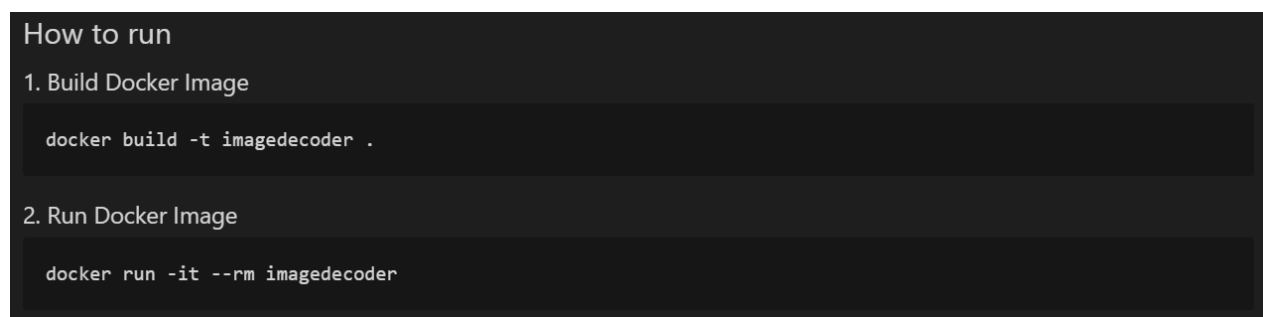
However - the important thing in this exercise is how you solved it, not the end result.

At the end of the programming exercise you should have:

- A list of **all** search queries you made to solve it, and timestamps (just copy it from the browser history)
- A list all pages you visited to solve it (just copy it from the browser history)
- A list of the 3 biggest stumbling blocks you came across and your reflection on why they were problematic (did you misunderstand something, was some of the info you found wrong, did you miss a detail, ...)
- A brief "every 30 min" diary as explained in the slides (this is more frequent than one would normally do, and is just meant as part of the exercise)

## 2. Løsning

Vi valgte at bruge Python som programmingssprog og vi har bygget det som en docker container. Nedenfor ses en række billeder af hvordan projektet kan køres, samt vores løsnings forslag med tilhørende resultat.



```
How to run

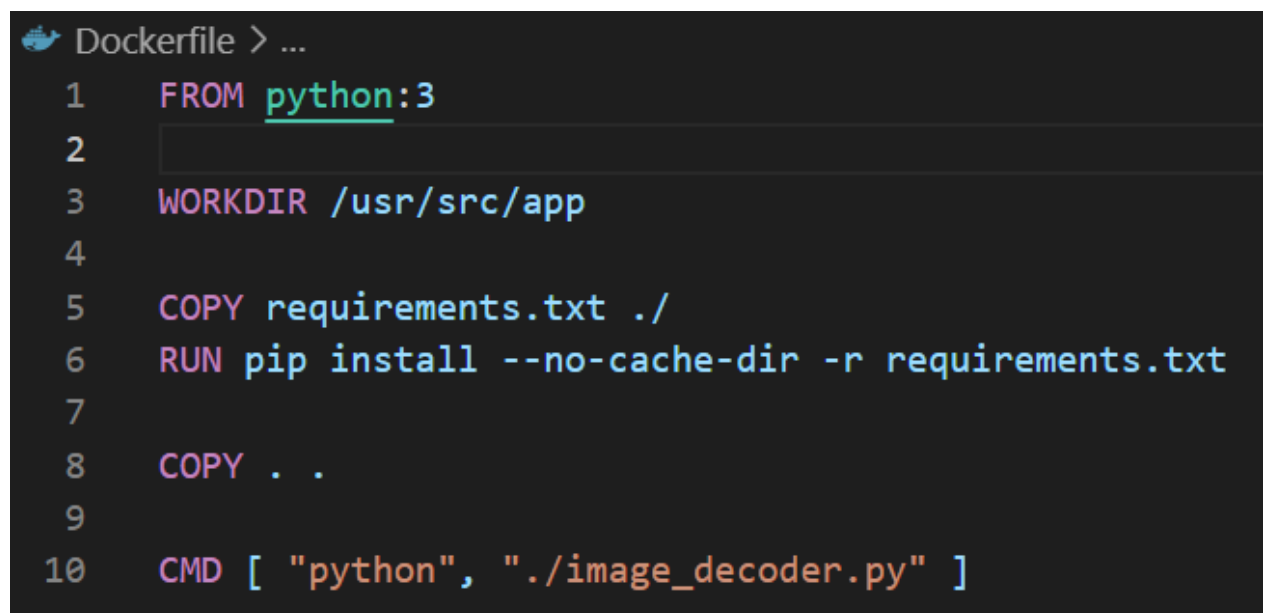
1. Build Docker Image

docker build -t imagedecoder .

2. Run Docker Image

docker run -it --rm imagedecoder
```

Figur 1: Billede af kommandoer der kan bruges til at køre projektet.



```
Dockerfile > ...

1  FROM python:3
2
3  WORKDIR /usr/src/app
4
5  COPY requirements.txt ./
6  RUN pip install --no-cache-dir -r requirements.txt
7
8  COPY . .
9
10 CMD [ "python", "./image_decoder.py" ]
```

Figur 2: Billede af vores dockerfil.

```
image_decoder.py
1  from PIL import Image
2  import binascii
3
4  img = Image.open('7fe3c3f6-Stego.png')
5  width, height = img.size
6
7  extracted_bin = [str(img.getpixel((y, x))[2]&1) for x in range(0, height) for y in range(0, width)]
8  extracted_bytes = ["".join(extracted_bin[i:i+8][::-1]) for i in range(0, len(extracted_bin), 8)]
9
10 result = ''
11 for byte in extracted_bytes:
12     if byte != '00000000':
13         result += chr(int(byte, 2))
14     else:
15         break
16
17 print ("Secret Message: ", result)
```

Figur 3: Billede af vores pythonkode.

```
PS C:\Users\Andreas Vikke\OneDrive\Documents\Skolet\SoftwareUdvikling\CPH-Business-UFO\Week1> docker build -t imagedecode
r .
[+] Building 0.9s (10/10) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 32B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/python:3 0.8s
=> [internal] load build context 0.0s
=> => transferring context: 237B 0.0s
=> [1/5] FROM docker.io/library/python:3@sha256:9dc8877a34fd1614f7d4240c7fc45853f3605eb37bbfa9ccef714a1a071529be 0.0s
=> CACHED [2/5] WORKDIR /usr/src/app 0.0s
=> CACHED [3/5] COPY requirements.txt ./ 0.0s
=> CACHED [4/5] RUN pip install --no-cache-dir -r requirements.txt 0.0s
=> CACHED [5/5] COPY . . 0.0s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:c81ea72cc36ae58ac0c652da4f1810fcde2683d49713c81dc31b1087dcbf5a79 0.0s
=> => naming to docker.io/library/imagedecoder 0.0s
PS C:\Users\Andreas Vikke\OneDrive\Documents\Skolet\SoftwareUdvikling\CPH-Business-UFO\Week1> docker run -it --rm imagede
coder
Secret Message: Congratulations, this is the secret message of the UFO class! (no, not 42)
PS C:\Users\Andreas Vikke\OneDrive\Documents\Skolet\SoftwareUdvikling\CPH-Business-UFO\Week1>
```

Figur 4: Billede af resultatet.

### 3. Største udfordringer

- **Forståelse af opgaven:** det største problem for vores gruppe var at forstå opgaven. Vi prøvede hver især først at decode billedet uden noget meningsfuldt resultat. Vi brugte dernæst lidt tid på lige at få snakket opgaven ordenligt igennem sådan at der ikke var nogen tvivl om hvad der præsisst skulle ske. Altså den klassiske programfejl med at starte på opgaven før den overhovedet er forstået.
- **Least significant bit of the blue values:** dette brugte vi lidt tid på at forstå ordenligt i forhold til at vi skal tage det sidste bit i den blå del af pixlen. Hvis vi printede en pixel ud fik vi RGBA værdien ud som (R, G, B, A) hvor A er alpha og beskriver transparens i den givne pixel. Derudover stod det ikke 100% klart at *Least significant* betød den sidste bit i farve værdien.
- **Little-endian:** her var vi langsomme om at opfatte at hver byte var spejlvendt.

## 4. Dagbog

### 05/01-21 kl. 14.07:

Da vi hver især bare kastede os ud i opgaven uden held brugte vi tiden på det første møde til at gennemgå opgaven samlet. På den måde kunne alle parter blive enige om hvad der præcis skulle ske. Vores gruppe havde hver især også problemer med at forstå enkelte ord/begreber i opgaven og under mødet kunne vi svare hinanden på hvad ordene/begreberne betød.

### 05/01-21 kl. 14.45:

Efter vi havde fået en forståelse af opgaven kunne vi begynde at arbejde rigtig på en løsning. Vi løb så ind i det næste problem før dette møde, hvilket var at forstå det output ordenligt som vi fik da vi så på en bestemt pixel. Vi var lidt i tvivl om hvad den sidst bit betød i forhold til pxilen da vi så på 4 værdier og ikke kun 3. Vi undersøgte så dette under mødet og kunne derved konstatere at den sidste værdi i pixlen er den værdi der beskriver hvor transparent pixlen er. Derudover var vi nødt til igen at undersøge hvad *Least significant* betød eftersom vi ikke havde den rette forståelse da det kom til stykket.

### 05/01-21 kl. 15.20:

Under dette møde havde vi en følelse af at vi var ved at være i mål. Vi havde nu alle sluttende bits fra den blå farve i hele billedet men outputtet gav ingen mening. Efter endnu en gennemlæsning af opgaven fandt vi vores fejl som lå i at vi skulle spejlvende alle vores bytes og derved fik vi et resultat der gav mening. Til sidst kunne vi også fjerne resten af outputtet efter vi fandt byten `00000000`.

## 5. Søgning og links

5/2-21, 13.45 - <https://en.wikipedia.org/wiki/Steganography>  
5/2-21, 13.45 - <https://amiradata.com/how-to-do-steganography-in-python/>  
5/2-21, 14.07 - <https://www.geeksforgeeks.org/image-based-steganography-using-python/>  
5/2-21, 14.08 - <https://pythonexamples.org/python-opencv-extract-blue-channel-from-color-image/>  
5/2-21, 14.08 - <https://pypi.org/project/opencv-python/>  
5/2-21, 14.12 - <https://py.processing.org/tutorials/pixels/>  
5/2-21, 14.14 - [https://bogotobogo.com/python/OpenCV\\_Python/python\\_opencv3\\_basic\\_image\\_operations\\_pixel\\_access\\_image\\_load.php](https://bogotobogo.com/python/OpenCV_Python/python_opencv3_basic_image_operations_pixel_access_image_load.php)  
5/2-21, 14.08 - <http://2017.compciv.org/guide/topics/python-nonstandard-libraries/pillow.html>  
5/2-21, 14.08 - <https://en.wikipedia.org/wiki/BPCS-steganography>  
5/2-21, 14.45 - [https://www.researchgate.net/publication/328097227\\_Hiding\\_Data\\_in\\_Color\\_Image\\_Using\\_Least\\_Significant\\_Bits\\_of\\_Blue\\_Sector](https://www.researchgate.net/publication/328097227_Hiding_Data_in_Color_Image_Using_Least_Significant_Bits_of_Blue_Sector)  
5/2-21, 14.45 - [https://www.researchgate.net/figure/A-pixel-in-RGB-color-space-Least-Significant-Bits-LSB-insertion-is-a-simple-approach\\_fig1\\_266403634](https://www.researchgate.net/figure/A-pixel-in-RGB-color-space-Least-Significant-Bits-LSB-insertion-is-a-simple-approach_fig1_266403634)  
5/2-21, 14.46 - <https://medium.com/swlh/lsb-image-steganography-using-python-2bbbee2c69a2>  
5/2-21, 14.50 - [https://www.tutorialspoint.com/python/list\\_reverse.htm](https://www.tutorialspoint.com/python/list_reverse.htm)  
5/2-21, 14.55 - <https://www.binaryhexconverter.com/binary-to-ascii-text-converter>  
5/2-21, 14.55 - <https://www.includehelp.com/python/count-total-number-of-bits-in-a-number.aspx>  
5/2-21, 14.59 - <https://itnext.io/steganography-101-lsb-introduction-with-python-4c4803e08041>  
5/2-21, 14.59 - <https://www.geeksforgeeks.org/python-pil-getpixel-method/>  
5/2-21, 15.05 - <http://2017.compciv.org/guide/topics/python-nonstandard-libraries/pillow.html>  
5/2-21, 15.06 - [https://www.researchgate.net/figure/A-pixel-in-RGB-color-space-Least-Significant-Bits-LSB-insertion-is-a-simple-approach\\_fig1\\_266403634](https://www.researchgate.net/figure/A-pixel-in-RGB-color-space-Least-Significant-Bits-LSB-insertion-is-a-simple-approach_fig1_266403634)  
5/2-21, 15.08 - <https://stackoverflow.com/questions/7396849/convert-binary-to-ascii-and-vice-versa>  
5/2-21, 15.10 - <https://py.processing.org/tutorials/pixels/>  
5/2-21, 15.20 - <https://learnmeabitcoin.com/technical/little-endian>  
5/2-21, 15.21 - [https://en.wikipedia.org/wiki/Null\\_character](https://en.wikipedia.org/wiki/Null_character)