

# Part 6A - Image File Basics, Metadata, Resizing, and Format Conversion Using ChatGPT Scripts

Yahya Nazer

2026.06.18

## Table of contents

<b>Part 6A - Image File Basics, Metadata, Resizing, and Format Conversion Using ChatGPT Scripts</b>	<b>2</b>
Purpose of This Part . . . . .	2
Learning Objectives . . . . .	3
<b>Course Coding Standard</b>	<b>3</b>
<b>Standard Folder Structure</b>	<b>3</b>
<b>Required Python Package</b>	<b>4</b>
<b>Chapter 1 - Understanding Image File Types</b>	<b>4</b>
Objective . . . . .	4
Important Terms . . . . .	4
Pixel . . . . .	4
Aspect Ratio . . . . .	4
Image Format . . . . .	4
Image Mode . . . . .	4
ChatGPT Script . . . . .	4
Exercise . . . . .	5
What You Learned . . . . .	5
<b>Chapter 2 - Reading Image Information</b>	<b>5</b>
Objective . . . . .	5
Folder Setup . . . . .	5
Python File . . . . .	6
ChatGPT Script . . . . .	6
ChatGPT Generated Python Code . . . . .	6
Run the Program . . . . .	8
Improve the Script . . . . .	8
Exercise . . . . .	8
What You Learned . . . . .	8
<b>Chapter 3 - Scanning an Image Folder</b>	<b>8</b>
Objective . . . . .	8
Folder Setup . . . . .	8
Python File . . . . .	8
ChatGPT Script . . . . .	9
ChatGPT Generated Python Code . . . . .	9

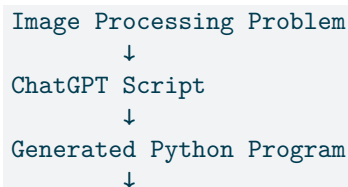
Improve the Script . . . . .	11
Exercise . . . . .	11
What You Learned . . . . .	11
<b>Chapter 4 - Resizing One Image</b>	<b>12</b>
Objective . . . . .	12
Python File . . . . .	12
ChatGPT Script . . . . .	12
ChatGPT Generated Python Code . . . . .	13
Improve the Script . . . . .	14
Exercise . . . . .	14
What You Learned . . . . .	14
<b>Chapter 5 - Batch Resizing Images</b>	<b>14</b>
Objective . . . . .	14
Python File . . . . .	15
ChatGPT Script . . . . .	15
ChatGPT Generated Python Code . . . . .	16
Improve the Script . . . . .	17
Exercise . . . . .	17
What You Learned . . . . .	17
<b>Chapter 6 - Converting Image Formats</b>	<b>17</b>
Objective . . . . .	17
Python File . . . . .	17
ChatGPT Script . . . . .	17
ChatGPT Generated Python Code . . . . .	18
Improve the Script . . . . .	19
Exercise . . . . .	19
What You Learned . . . . .	20
<b>Part 6A Review</b>	<b>20</b>
What You Built . . . . .	20
Skills Checklist . . . . .	20
Important ChatGPT Script Pattern . . . . .	20
Summary . . . . .	20
Looking Ahead . . . . .	21

## Part 6A - Image File Basics, Metadata, Resizing, and Format Conversion Using ChatGPT Scripts

### Purpose of This Part

In Part 6A, you will begin working with image files. You will learn how to ask ChatGPT to create Python programs that can understand image file types, read image information, scan image folders, resize images, preserve aspect ratio, convert formats, and save processed images to **C-Results**.

The main workflow remains:



```
Run in VS Code
  ↓
Check Output
  ↓
Improve Script
  ↓
Improve Program
```

## Learning Objectives

After completing Part 6A, you will be able to:

- Explain common image file types
- Write ChatGPT scripts for image programs
- Use Pillow in Python
- Open image files
- Read image width and height
- Read image mode and format
- Scan image folders
- Resize images
- Preserve image aspect ratio
- Convert image formats
- Save processed images to C-Results

## Course Coding Standard

Every Python program should include:

- Program Name
- Purpose
- User ChatGPT Script
- Expected Output
- Version
- Clearly labeled steps
- Beginner-friendly comments

Example:

```
# =====
# STEP 1 - Import Libraries
# =====
```

## Standard Folder Structure

Use this folder structure:

```
ChatGPT-Python-Course
  A-Data
    Images
  B-Engine
  C-Results
  D-Documentation
```

Input images go in A-Data/Images. Python programs go in B-Engine. Processed images and reports go in C-Results.

## Required Python Package

For image processing, install Pillow.

```
python -m pip install pillow
```

On some Mac computers:

```
python3 -m pip install pillow
```

## Chapter 1 - Understanding Image File Types

### Objective

Understand common image file types and when they are used.

Common image extensions include:

```
.jpg  
.jpeg  
.png  
.tif  
.tiff  
.heic  
.webp
```

### Important Terms

#### Pixel

A pixel is a small dot in an image. An image size may be 4000 x 3000 pixels.

#### Aspect Ratio

Aspect ratio means the relationship between width and height. Examples: 4:3, 3:2, 16:9, 5:7.

#### Image Format

Image format means how the image is saved. Examples: JPEG, PNG, TIFF.

#### Image Mode

Image mode describes color information. Examples: RGB, RGBA, L, CMYK.

### ChatGPT Script

Copy this into ChatGPT:

```
Explain common image file types for a beginner.
```

```
Include:
```

1. JPG
2. PNG
3. TIFF
4. HEIC
5. WEBP

```
For each file type explain:
```

1. What it is used for.
2. Advantages.
3. Disadvantages.
4. When to use it.

Also explain:

1. Pixels
2. Image size
3. Aspect ratio
4. Image mode
5. DPI

## Exercise

Ask ChatGPT:

```
I have photos from a camera and iPhone.
```

```
Which formats are best for:
```

1. Editing
2. Printing
3. Website use
4. Long-term archive

```
Explain in simple language.
```

## What You Learned

You learned the basic vocabulary needed for image processing programs.

## Chapter 2 - Reading Image Information

### Objective

Create a Python program that reads one image and displays basic information.

The program should show:

- File name
- Format
- Width
- Height
- Mode

### Folder Setup

Place one image in:

```
A-Data/Images
```

Example:

```
A-Data/Images/sample.jpg
```

## Python File

Create:

```
B-Engine/Read-Image-Info.py
```

## ChatGPT Script

Copy this into ChatGPT:

```
Write a Python program.
```

```
Project folder structure:
```

```
Top Folder
```

```
+-- A-Data
```

```
    +- Images
```

```
+-- B-Engine
```

```
+-- C-Results
```

```
+-- D-Documentation
```

```
Program Name:
```

```
Read-Image-Info.py
```

```
Program purpose:
```

```
Read one image from A-Data/Images/sample.jpg and display image information.
```

```
Requirements:
```

1. Use pathlib.
2. Use Pillow.
3. Read A-Data/Images/sample.jpg.
4. Display file name, image format, width, height, and image mode.
5. If the image file is missing, show a friendly error message.
6. Do not modify the image.

```
Coding Standards:
```

1. Add a detailed program header.
2. Include Program Name, Purpose, User ChatGPT Script, Expected Output, and Version.
3. Divide code into clearly labeled steps.
4. Add comments explaining every important line.
5. Make code beginner friendly.

```
Expected output:
```

```
The program displays basic image information.
```

## ChatGPT Generated Python Code

```
# =====  
# Program Name:  
# Read-Image-Info.py  
#  
# Purpose:  
# Read one image and display basic image information.  
#  
# User ChatGPT Script:  
# Write a Python program that reads A-Data/Images/sample.jpg
```

```

# using Pillow and displays file name, format, width, height,
# and image mode.
#
# Expected Output:
# The program displays basic image information.
#
# Version:
# 1.0
# =====

# =====
# STEP 1 - Import Libraries
# =====

from pathlib import Path
from PIL import Image

# =====
# STEP 2 - Define Input Image
# =====

image_file = Path("A-Data") / "Images" / "sample.jpg"

# =====
# STEP 3 - Check if Image Exists
# =====

if not image_file.exists():
    print("ERROR: Image file was not found.")
    print("Expected file:", image_file)
else:
    # =====
    # STEP 4 - Open Image
    # =====
    image = Image.open(image_file)

    # =====
    # STEP 5 - Display Image Information
    # =====
    print("Image Information")
    print("-----")
    print("File Name:", image_file.name)
    print("Format   :", image.format)
    print("Width    :", image.width)
    print("Height   :", image.height)
    print("Mode     :", image.mode)

# =====
# STEP 6 - End Program
# =====

print("Program finished.")

```

## Run the Program

```
python Read-Image-Info.py
```

On some Mac computers:

```
python3 Read-Image-Info.py
```

## Improve the Script

Ask ChatGPT:

```
Please improve the program.
```

```
New requirements:
```

1. Ask the user to type the image filename.
2. Look for the image in A-Data/Images.
3. Display image size in megapixels.
4. Save the image information to C-Results/Image-Info.txt.
5. Keep the program header and step comments.

## Exercise

Write a ChatGPT script for a program that reads a PNG file and reports whether it has transparency.

## What You Learned

You learned how to ask ChatGPT to generate a program that opens one image and reads image information.

## Chapter 3 - Scanning an Image Folder

### Objective

Create a Python program that scans a folder of images and creates a text report.

The report should include:

- File name
- Extension
- Width
- Height
- Format
- Mode

### Folder Setup

Place several images in A-Data/Images.

### Python File

Create:

```
B-Engine/Scan-Image-Folder.py
```

## ChatGPT Script

Copy this into ChatGPT:

Write a Python program.

Project folder structure:

Top Folder

+ A-Data  
  + Images

+ B-Engine

+ C-Results

+ D-Documentation

Program Name:

Scan-Image-Folder.py

Program purpose:

Scan A-Data/Images and create a report of image information.

Requirements:

1. Use pathlib.
2. Use Pillow.
3. Scan A-Data/Images and subfolders.
4. Include image extensions: .jpg, .jpeg, .png, .tif, .tiff, .webp.
5. For each image report file name, extension, folder, width, height, format, and mode.
6. Save the report to C-Results/Image-Folder-Report.txt.
7. Print how many images were found.
8. If an image cannot be opened, log an error and continue.

Coding Standards:

1. Add a detailed program header.
2. Include Program Name, Purpose, User ChatGPT Script, Expected Output, and Version.
3. Divide code into clearly labeled steps.
4. Add comments explaining every important line.
5. Make code beginner friendly.
6. Use UTF-8 when writing text files.

Expected output:

A text report is saved to C-Results.

## ChatGPT Generated Python Code

```
# =====  
# Program Name:  
# Scan-Image-Folder.py  
#  
# Purpose:  
# Scan A-Data/Images and create a report of image information.  
#  
# User ChatGPT Script:  
# Write a Python program that scans A-Data/Images and subfolders,  
# reads image information using Pillow, and saves a text report.  
#
```

```

# Expected Output:
# A text report is saved to C-Results.
#
# Version:
# 1.0
# =====

# =====
# STEP 1 - Import Libraries
# =====

from pathlib import Path
from PIL import Image

# =====
# STEP 2 - Define Folders
# =====

image_folder = Path("A-Data") / "Images"
results_folder = Path("C-Results")
results_folder.mkdir(exist_ok=True)
report_file = results_folder / "Image-Folder-Report.txt"

# =====
# STEP 3 - Define Image Extensions
# =====

image_extensions = {".jpg", ".jpeg", ".png", ".tif", ".tiff", ".webp"}

# =====
# STEP 4 - Scan Images
# =====

report_lines = []
image_count = 0
error_count = 0

report_lines.append("Image Folder Report")
report_lines.append("=====")
report_lines.append("")

for file_path in image_folder.rglob("*"):
    if not file_path.is_file():
        continue

    if file_path.suffix.lower() not in image_extensions:
        continue

    try:
        image = Image.open(file_path)
        report_lines.append(f"File Name: {file_path.name}")
        report_lines.append(f"Extension: {file_path.suffix.lower()}")
        report_lines.append(f"Folder    : {file_path.parent}")

```

```

report_lines.append(f"Width      : {image.width}")
report_lines.append(f"Height     : {image.height}")
report_lines.append(f"Format    : {image.format}")
report_lines.append(f"Mode      : {image.mode}")
report_lines.append("")
image_count += 1
except Exception as e:
report_lines.append(f"ERROR opening {file_path}: {e}")
report_lines.append("")
error_count += 1

# =====
# STEP 5 - Save Report
# =====

report_lines.append(f"Images found: {image_count}")
report_lines.append(f"Errors      : {error_count}")

report_file.write_text("\n".join(report_lines), encoding="utf-8")

# =====
# STEP 6 - Display Summary
# =====

print("Images found:", image_count)
print("Errors      :", error_count)
print("Report saved to:", report_file)

# =====
# STEP 7 - End Program
# =====

print("Program finished.")

```

## Improve the Script

Ask ChatGPT:

Please improve the program.

New requirements:

1. Save the image folder report as Excel instead of TXT.
2. Include size in MB.
3. Include megapixels.
4. Include orientation: Landscape, Portrait, or Square.
5. Keep the program header and step comments.

## Exercise

Write a ChatGPT script for a program that scans only JPG files and saves an Excel report.

## What You Learned

You learned how to ask ChatGPT to generate a folder-based image report program.

## Chapter 4 - Resizing One Image

### Objective

Create a Python program that resizes one image and saves the result to C-Results.

### Python File

Create:

```
B-Engine/Resize-One-Image.py
```

### ChatGPT Script

Copy this into ChatGPT:

```
Write a Python program.
```

```
Project folder structure:
```

```
Top Folder
```

```
+-- A-Data  
    +- Images
```

```
+-- B-Engine
```

```
+-- C-Results
```

```
+-- D-Documentation
```

```
Program Name:
```

```
Resize-One-Image.py
```

```
Program purpose:
```

```
Resize A-Data/Images/sample.jpg to a smaller width and save it to C-Results.
```

```
Requirements:
```

1. Use pathlib.
2. Use Pillow.
3. Read A-Data/Images/sample.jpg.
4. Resize the image to width 1200 pixels.
5. Preserve the original aspect ratio.
6. Save the resized image to C-Results/sample-resized.jpg.
7. Do not modify the original image.
8. Print original size and new size.
9. If the input image is missing, show a friendly error message.

```
Coding Standards:
```

1. Add a detailed program header.
2. Include Program Name, Purpose, User ChatGPT Script, Expected Output, and Version.
3. Divide code into clearly labeled steps.
4. Add comments explaining every important line.
5. Make code beginner friendly.

```
Expected output:
```

```
A resized image is saved to C-Results.
```

## ChatGPT Generated Python Code

```
# =====
# Program Name:
# Resize-One-Image.py
#
# Purpose:
# Resize one image to width 1200 pixels while preserving aspect ratio.
#
# User ChatGPT Script:
# Write a Python program that reads A-Data/Images/sample.jpg,
# resizes it to width 1200 pixels, preserves aspect ratio,
# and saves the resized image to C-Results.
#
# Expected Output:
# A resized image is saved to C-Results.
#
# Version:
# 1.0
# =====

# =====
# STEP 1 - Import Libraries
# =====

from pathlib import Path
from PIL import Image

# =====
# STEP 2 - Define Files and Folders
# =====

input_file = Path("A-Data") / "Images" / "sample.jpg"
results_folder = Path("C-Results")
results_folder.mkdir(exist_ok=True)
output_file = results_folder / "sample-resized.jpg"

# =====
# STEP 3 - Check Input File
# =====

if not input_file.exists():
    print("ERROR: Input image was not found.")
    print("Expected file:", input_file)
else:
    image = Image.open(input_file)
    original_width = image.width
    original_height = image.height

    # =====
    # STEP 4 - Calculate New Size
    # =====

    new_width = 1200
```

```

scale_ratio = new_width / original_width
new_height = int(original_height * scale_ratio)

# =====
# STEP 5 - Resize Image
# =====

resized_image = image.resize((new_width, new_height))

# =====
# STEP 6 - Save Resized Image
# =====

resized_image.save(output_file)

print("Original size:", original_width, "x", original_height)
print("New size      :", new_width, "x", new_height)
print("Saved to     :", output_file)

# =====
# STEP 7 - End Program
# =====

print("Program finished.")

```

## Improve the Script

Ask ChatGPT:

Please improve the program.

New requirements:

1. Ask the user for target width.
2. Validate that target width is a number.
3. Add timestamp to the output filename.
4. Save output with JPEG quality 95.
5. Keep the program header and step comments.

## Exercise

Write a ChatGPT script that resizes one image to height 1000 pixels while preserving aspect ratio.

## What You Learned

You learned how to ask ChatGPT to generate a program that resizes one image.

## Chapter 5 - Batch Resizing Images

### Objective

Create a Python program that resizes all images in a folder.

The program should:

- Read images from A-Data/Images

- Resize each image to width 1200 pixels
- Preserve aspect ratio
- Save output to C-Results/Resized-Images
- Keep original files unchanged

## Python File

Create:

B-Engine/Batch-Resize-Images.py

## ChatGPT Script

Copy this into ChatGPT:

Write a Python program.

Project folder structure:

Top Folder

```
+-- A-Data
    +- Images
+- B-Engine
+- C-Results
+- D-Documentation
```

Program Name:

Batch-Resize-Images.py

Program purpose:

Resize all image files in A-Data/Images and save resized copies.

Requirements:

1. Use pathlib.
2. Use Pillow.
3. Read images from A-Data/Images and subfolders.
4. Process extensions: .jpg, .jpeg, .png, .tif, .tiff, .webp.
5. Resize each image to width 1200 pixels.
6. Preserve aspect ratio.
7. Save resized images to C-Results/Resized-Images.
8. Do not modify original files.
9. Keep the same base filename.
10. Add -resized before the file extension.
11. Print each processed file.
12. Count processed images and errors.

Coding Standards:

1. Add a detailed program header.
2. Include Program Name, Purpose, User ChatGPT Script, Expected Output, and Version.
3. Divide code into clearly labeled steps.
4. Add comments explaining every important line.
5. Make code beginner friendly.

Expected output:

Resized images are saved to C-Results/Resized-Images.

## ChatGPT Generated Python Code

```
# =====
# Program Name:
# Batch-Resize-Images.py
#
# Purpose:
# Resize all image files in A-Data/Images and save resized copies.
#
# User ChatGPT Script:
# Write a Python program that scans A-Data/Images, resizes images
# to width 1200 pixels, preserves aspect ratio, and saves copies
# to C-Results/Resized-Images.
#
# Expected Output:
# Resized images are saved to C-Results/Resized-Images.
#
# Version:
# 1.0
# =====

from pathlib import Path
from PIL import Image

input_folder = Path("A-Data") / "Images"
output_folder = Path("C-Results") / "Resized-Images"
output_folder.mkdir(parents=True, exist_ok=True)

target_width = 1200
image_extensions = {".jpg", ".jpeg", ".png", ".tif", ".tiff", ".webp"}

processed_count = 0
error_count = 0

for input_file in input_folder.rglob("*"):
    if not input_file.is_file():
        continue
    if input_file.suffix.lower() not in image_extensions:
        continue

    try:
        image = Image.open(input_file)
        original_width = image.width
        original_height = image.height
        scale_ratio = target_width / original_width
        new_height = int(original_height * scale_ratio)
        resized_image = image.resize((target_width, new_height))
        output_name = input_file.stem + "-resized" + input_file.suffix.lower()
        output_file = output_folder / output_name
        resized_image.save(output_file)
        print("Resized:", input_file.name, "->", output_file.name)
        processed_count += 1
    except Exception as e:
        print("ERROR processing:", input_file, e)
```

```
        error_count += 1

print()
print("Images processed:", processed_count)
print("Errors          :", error_count)
print("Output folder   :", output_folder)
print("Program finished.")
```

## Improve the Script

Ask ChatGPT:

Please improve the batch resize program.

New requirements:

1. Ask the user for target width.
2. Preserve folder structure in the output folder.
3. Save a processing log to C-Results.
4. Skip images that are already smaller than the target width.
5. Keep the program header and step comments.

## Exercise

Write a ChatGPT script for a program that creates small web images at width 800 pixels.

## What You Learned

You learned how to ask ChatGPT to create a batch image resizing program.

# Chapter 6 - Converting Image Formats

## Objective

Create a Python program that converts images from one format to another.

Example:

```
PNG to JPG
JPG to PNG
TIFF to JPG
```

## Python File

Create:

```
B-Engine/Convert-Images-to-JPG.py
```

## ChatGPT Script

Copy this into ChatGPT:

Write a Python program.

Project folder structure:

```
Top Folder
+- A-Data
```

```
+ - Images
+ - B-Engine
+ - C-Results
+ - D-Documentation
```

Program Name:  
Convert-Images-to-JPG.py

Program purpose:  
Convert image files from A-Data/Images to JPG format.

Requirements:

1. Use pathlib.
2. Use Pillow.
3. Read image files from A-Data/Images and subfolders.
4. Convert .png, .tif, .tiff, and .webp files to .jpg.
5. Save converted files to C-Results/JPG-Images.
6. Do not modify original files.
7. Convert images to RGB before saving as JPG.
8. Use JPEG quality 95.
9. Print each converted file.
10. Count converted files and errors.

Coding Standards:

1. Add a detailed program header.
2. Include Program Name, Purpose, User ChatGPT Script, Expected Output, and Version.
3. Divide code into clearly labeled steps.
4. Add comments explaining every important line.
5. Make code beginner friendly.

Expected output:  
Converted JPG images are saved to C-Results/JPG-Images.

## ChatGPT Generated Python Code

```
# =====
# Program Name:
# Convert-Images-to-JPG.py
#
# Purpose:
# Convert image files from A-Data/Images to JPG format.
#
# User ChatGPT Script:
# Write a Python program that converts PNG, TIFF, and WEBP images
# from A-Data/Images to JPG and saves them to C-Results/JPG-Images.
#
# Expected Output:
# Converted JPG images are saved to C-Results/JPG-Images.
#
# Version:
# 1.0
# =====
```

```

from pathlib import Path
from PIL import Image

input_folder = Path("A-Data") / "Images"
output_folder = Path("C-Results") / "JPG-Images"
output_folder.mkdir(parents=True, exist_ok=True)

convert_extensions = {".png", ".tif", ".tiff", ".webp"}

converted_count = 0
error_count = 0

for input_file in input_folder.rglob("*"):
    if not input_file.is_file():
        continue
    if input_file.suffix.lower() not in convert_extensions:
        continue

    try:
        image = Image.open(input_file)
        rgb_image = image.convert("RGB")
        output_file = output_folder / (input_file.stem + ".jpg")
        rgb_image.save(output_file, quality=95)
        print("Converted:", input_file.name, "->", output_file.name)
        converted_count += 1
    except Exception as e:
        print("ERROR converting:", input_file, e)
        error_count += 1

print()
print("Images converted:", converted_count)
print("Errors          :", error_count)
print("Output folder   :", output_folder)
print("Program finished.")

```

## Improve the Script

Ask ChatGPT:

Please improve the conversion program.

New requirements:

1. Add GUI folder selection.
2. Allow the user to choose output format: JPG or PNG.
3. Save converted images in timestamped folder.
4. Save a log file.
5. Keep the program header and step comments.

## Exercise

Write a ChatGPT script that converts JPG files to PNG files.

## What You Learned

You learned how to ask ChatGPT to generate image format conversion programs.

## Part 6A Review

### What You Built

In Part 6A, you created ChatGPT scripts and Python programs for:

- Understanding image file types
- Reading image information
- Scanning image folders
- Resizing one image
- Batch resizing images
- Converting image formats

### Skills Checklist

By the end of Part 6A, you should be able to:

- Explain image file types
- Use Pillow
- Read image width and height
- Read image mode and format
- Scan image folders
- Resize one image
- Batch resize images
- Preserve aspect ratio
- Convert image formats
- Save processed images to C-Results
- Ask ChatGPT to improve image programs

### Important ChatGPT Script Pattern

For image programs, always tell ChatGPT:

1. Where the input images are located.
2. Which image extensions to process.
3. Whether original files must remain unchanged.
4. Where to save output files.
5. Whether aspect ratio should be preserved.
6. Whether logs are required.
7. What coding standard to use.

### Summary

In Part 6A, you learned how to use ChatGPT scripts to generate Python image processing programs.

A safe image workflow should:

```
Read Original Image
  ↓
Create Copy
  ↓
Process Copy
  ↓
```

Save to C-Results



Keep Original Unchanged

## Looking Ahead

In Part 6B, you will learn how to create print-ready images, preserve aspect ratio on a fixed canvas, add page numbers, and create contact sheets.