# SOP: Creating an HTML-Based Report from Canva PNGs on Ubuntu (WSL)

## Objective

To generate a navigable set of standalone HTML pages from PNG exports of a Canva design, using lossless compression and Python automation on an Ubuntu-based system (e.g., WSL).

## Pre-requisites

- Ubuntu shell (tested on WSL / Ubuntu 20.04+)

- Basic command-line proficiency

- Tools installed:

- pngcrush

- python3

## Step 1: Export PNG Pages from Canva

1. Open your Canva project.

2. Click Share → Download.

3. Choose File type: PNG, and select All Pages.

4. Download the zipped file and extract it into a local folder.

5. Ensure the PNGs are named sequentially (e.g., 1.png, 2.png, ..., 76.png).

## Step 2: Compress PNG Images (Lossless)

Use pngcrush to optimize images that are larger than 1 MB.

Install pngcrush (if not already):

sudo apt update

sudo apt install pngcrush

Run the following script in your shell:

mkdir -p crushed

find . -type f -size +1M -iname "\*.png" -exec bash -c '

for img; do

base=$(basename "$img")

pngcrush -brute -reduce "$img" "crushed/$base"

done

' bash {} +

This script finds all .png files over 1MB and compresses them using pngcrush. Output is saved to the crushed/ folder.

Note: Compression is lossless, so image quality remains unchanged.

## Step 3: Generate HTML Pages Using Python Script

Move to the crushed/ folder:

cd crushed

Save the following script as generate\_html\_pages.py and run it with python3 generate\_html\_pages.py.

(Full script included at the end of this document.)

## Step 4: Create the Home Page

Duplicate 1.html as the landing page:

cp 1.html index.html

You may edit index.html later to customize its appearance as the entry point.

## Final Output

- index.html — Home page

- 1.html, 2.html, ..., N.html — All individual pages

- All PNGs are served directly, not base64 embedded

- All files in crushed/ folder ready for static web hosting or ZIP distribution

## Optional Next Steps

- Upload the folder to GitHub Pages or Netlify for online access

- Bundle the folder into a .zip for sharing

- Add a search or table of contents (via another script)

## Appendix: Python Script for HTML Generation

import os

def extract\_number(filename):

return int(os.path.splitext(filename)[0]) # Assumes filenames like '1.png'

image\_files = sorted(

[f for f in os.listdir() if f.lower().endswith('.png')],

key=extract\_number

)

for i, img\_file in enumerate(image\_files):

page\_num = i + 1

prev\_page = f"{page\_num - 1}.html" if i > 0 else None

next\_page = f"{page\_num + 1}.html" if i < len(image\_files) - 1 else None

image\_src = f"{img\_file}"

home\_page = "index.html"

html = f'''<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Page {page\_num}</title>

<style>

body {{

text-align: center;

font-family: sans-serif;

margin: 0;

padding: 20px;

}}

.nav {{

margin: 20px auto;

font-size: 20px;

}}

.nav a {{

background-color: #007BFF;

color: white;

text-decoration: none;

padding: 10px 20px;

margin: 0 10px;

border-radius: 5px;

font-weight: bold;

display: inline-block;

}}

.nav a:hover {{

background-color: #0056b3;

}}

.home-button {{

font-size: 16px;

padding: 6px 14px;

margin-top: 10px;

}}

img {{

max-width: 100%;

height: auto;

margin-top: 20px;

box-shadow: 0 4px 12px rgba(0,0,0,0.1);

}}

.side-nav {{

position: fixed;

top: 50%;

transform: translateY(-50%);

font-size: 18px;

}}

.side-nav.left {{

left: 10px;

}}

.side-nav.right {{

right: 10px;

}}

.side-nav a {{

background-color: #007BFF;

color: white;

text-decoration: none;

padding: 10px 16px;

border-radius: 5px;

display: inline-block;

}}

.side-nav a:hover {{

background-color: #0056b3;

}}

</style>

</head>

<body>

<div class="nav">

{{'<a href="' + prev\_page + '">← Prev</a>' if prev\_page else ''}}

{{'<a href="' + next\_page + '">Next →</a>' if next\_page else ''}}

</div>

<h2>Page {{page\_num}}</h2>

<div>

<a href="{{home\_page}}" class="nav home-button">Home</a>

</div>

<img src="{{image\_src}}" alt="Page {{page\_num}}">

<div class="nav">

{{'<a href="' + prev\_page + '">← Prev</a>' if prev\_page else ''}}

{{'<a href="' + next\_page + '">Next →</a>' if next\_page else ''}}

</div>

{{'<div class="side-nav left"><a href="' + prev\_page + '">←</a></div>' if prev\_page else ''}}

{{'<div class="side-nav right"><a href="' + next\_page + '">→</a></div>' if next\_page else ''}}

</body>

</html>

'''

with open(f"{page\_num}.html", "w", encoding="utf-8") as f:

f.write(html)