```
# 1) ZIP
import os, re, zipfile, json, hashlib
from pathlib import Path
import pandas as pd
from datetime import datetime
# | | | | | | | |
outdir = Path("/mnt/data/KABUKI_INV_2025-06-01_outputs")
outdir.mkdir(exist ok=True)
def sha256 file(path):
h = hashlib.sha256()
with open(path, "rb") as f:
for chunk in iter(lambda: f.read(8192), b""):
h.update(chunk)
return h.hexdigest()
def extract_zip_to_dir(zip_path, extract_to):
with zipfile.ZipFile(zip_path, "r") as z:
z.extractall(extract_to)
return list(Path(extract_to).rglob("*"))
def scan file for tamper(file path, width list):
"""
with open(file_path, "rb") as f:
data = f.read()
text = data.decode("utf-8","ignore")
results = []
for width in width_list:
windows = [text[i:i+width] for i in range(0, len(text), width)]
for w in windows:
if re.search(r"\\u[0-9a-fA-F]{4}", w):
results.append({"width": width, "window": w[:80]})
return results
def normalize_time(t):
try:
return datetime.fromisoformat(t).isoformat()
return t
# 70
widths = [
222,555,888,2222,5555,8888,12222,15555,18888,
22222,25555,28888,32222,35555,38888,42222,45555,
4888,52222,55555,58888,62222,65555,68888,72222,
75555,78888,82222,85555,88888,92222,95555,98888,
102222,105555,108888,112222,115555,118888,122222,
125555,128888,132222,135555,138888,142222,145555,
148888,152222,155555,158888,162222,165555,168888,
172222,175555,178888,182222,185555,188888,192222,
195555,198888,202222,205555,208888,212222,215555,
218888,222222
# 2) PDF
def make_pdf(path, text):
from reportlab.platypus import SimpleDocTemplate, Paragraph
from reportlab.lib.styles import getSampleStyleSheet
from reportlab.lib.pagesizes import A4
```

```
styles = getSampleStyleSheet()
doc = SimpleDocTemplate(str(path), pagesize=A4)
story = [Paragraph(t, styles["Normal"]) for t in text.split("\n")]
doc.build(story)
# 3) INTERPORT OF THE PROPERTY OF THE PROPERT
if not date map df.empty:
for col in date_map_df.select_dtypes(include=[object]).columns:
date map df[col] =
date map df[col].astype(str).str.encode("utf-8", "replace").str.decode("utf-8", "replace")
date map df.to csv(outdir/"DATE MAP.csv", index=False, encoding="utf-8")
if not mixed_df.empty:
for col in mixed_df.select_dtypes(include=[object]).columns:
mixed_df[col] = mixed_df[col].astype(str).str.encode("utf-8","replace").str.decode("utf-8","replace")
mixed_df.to_csv(outdir/"MIXED_DATE_MAP.csv", index=False, encoding="utf-8")
# 4)
import pandas as pd
import ison
from reportlab.platypus import SimpleDocTemplate, Paragraph
from reportlab.lib.styles import getSampleStyleSheet
from reportlab.lib.pagesizes import A4
outdir3 = Path("/mnt/data/KABUKI_INV_2025-06-01_outputs_template3")
outdir3.mkdir(exist ok=True)
# 5)
outdir4 = Path("/mnt/data/KABUKI_INV_2025-06-01_outputs_template4")
outdir4.mkdir(exist_ok=True)
```

... (■■: summary■CSV/JSON/TXT/PDF■■■■)