

## File: .pdfignore.yml

```
directories:
  - .git
  - log
  - public/system
  - tmp
  - vendor
  - build
  - dist
files:
  - .DS_Store
  - database.yml
  - favicon.ico
  - requirements.txt
patterns:
  - '*.pdf'
  - '*.tmp'
  - '*.md'
# - '*.py'
```

## File: code-to-pdf.py

```
import fnmatch
import os
import yaml
from reportlab.lib.pagesizes import letter
from reportlab.lib.styles import ParagraphStyle, getSampleStyleSheet
from reportlab.pdfbase import pdfmetrics
from reportlab.pdfbase.cidfonts import UnicodeCIDFont
from reportlab.platypus import Paragraph, SimpleDocTemplate, Spacer
code_extensions = [
    ".py",
    ".js",
    ".html",
    ".css",
    ".php",
    ".java",
    ".cpp",
    ".c",
    ".h",
    ".hpp",
    ".cs",
    ".rb",
    ".pl",
    ".sh",
    ".sql",
    ".json",
    ".xml",
    ".yaml",
    ".yml",
    ".md",
```

```

        ".rst",
        ".txt",
    ]
def load_exclusions(config_path):
    with open(config_path, "r", encoding="utf-8") as file:
        config = yaml.safe_load(file)
        excluded_dirs = set(config.get("directories", []))
        excluded_files = set(config.get("files", []))
        excluded_patterns = config.get("patterns", [])
        return excluded_dirs, excluded_files, excluded_patterns
def generate_pdf(source_folder, output_pdf, config_path):
    excluded_dirs, excluded_files, excluded_patterns = load_exclusions(config_path)
    doc = SimpleDocTemplate(output_pdf, pagesize=letter)
    story = []
    styles = getSampleStyleSheet()
    pdfmetrics.registerFont(UnicodeCIDFont("STSong-Light"))
    code_style = ParagraphStyle(
        "Code",
        parent=styles["Normal"],
        fontName="STSong-Light",
        fontSize=10,
        leading=12,
    )
    for root, dirs, files in os.walk(source_folder, topdown=True):
        dirs[:] = [d for d in dirs if d not in excluded_dirs]
        for filename in files:
            if any(filename.endswith(ext) for ext in code_extensions):
                if filename not in excluded_files and not any(
                    fnmatch.fnmatch(filename, pattern) for pattern in exclu
ded_patterns
                ):
                    if not any(
                        root.startswith(os.path.join(source_folder, d))
                        for d in excluded_dirs
                    ):
                        filepath = os.path.join(root, filename)
                        with open(filepath, "r", encoding="utf-8") as
file:
                            code = file.read()
                            story.append(
                                Paragraph(f"File: {filename}", s
tyles["Title"])
                            )
                            for line in code.splitlines():
                                story.append(
                                    Paragraph(line.replace("
", " "), code_style)
                                )
                            story.append(Spacer(1, 12))
    doc.build(story)
# 使用该函数
generate_pdf("./", "Code.pdf", ".pdfignore.yml")

```