

Sistema di Esportazione PDF

Questo sistema permette di generare automaticamente file PDF da tutte le pagine markdown del sito, con supporto completo per i diagrammi Mermaid.

Come funziona

Architettura

1. Durante il Build (GitHub Actions):

- MkDocs genera il sito statico nella cartella `site/`
- Lo script `code/generate_pdfs.py` crea i PDF per ogni file markdown
- I PDF vengono salvati in `site/pdf/`
- Il deploy include automaticamente i PDF

2. Sul Sito Live:

- Ogni pagina ha un bottone "Scarica PDF" in alto a destra
- Cliccando il bottone si scarica il PDF della pagina corrente
- I PDF includono tutti i diagrammi Mermaid correttamente renderizzati

Componenti

- `code/generate_pdfs.py` : Script Python che:
 - Scansiona tutti i file `.md` in `docs/`
 - Converte markdown → HTML con supporto Mermaid
 - Usa Chrome headless per generare PDF di alta qualità
 - Salva i PDF in `site/pdf/`
- `hooks/pdf_button.py` : Hook MkDocs che:
 - Aggiunge un bottone download PDF in ogni pagina
 - Inserisce il link corretto al PDF
- `.github/workflows/ci.yml` : GitHub Actions workflow che:
 - Installa Chrome per generazione PDF

- Esegue il build MkDocs
- Genera i PDF
- Fa il deploy su GitHub Pages

Test in locale

Prerequisiti

- Python 3.x
- Google Chrome o Chromium installato
- MkDocs e dipendenze (vedi `requirements.txt`)

Comandi

```
# 1. Build del sito MkDocs
source infoenv/bin/activate
mkdocs build

# 2. Genera i PDF
python code/generate_pdfs.py

# 3. Verifica i PDF generati
ls -lh site/pdf/

# 4. Testa il sito in locale con i PDF
mkdocs serve

# Vai su http://localhost:8000 e clicca sul bottone "Scarica PDF"
```

Struttura file generati

I PDF vengono nominati convertendo il percorso del markdown:

- `docs/index.md` → `site/pdf/index.pdf`
- `docs/algocoding/01_flowgorithm.md` → `site/pdf/algocoding_01_flowgorithm.pdf`
- `docs/hsw/10_hw.md` → `site/pdf/hsw_10_hw.pdf`

Modifiche necessarie per aggiungere nuove pagine

Nessuna! Il sistema è completamente automatico:

1. Aggiungi un nuovo file `.md` in `docs/`
2. Fai commit con tag `[ci]`
3. GitHub Actions genera automaticamente il PDF
4. Il bottone appare automaticamente nella pagina

Troubleshooting

I PDF non vengono generati

Problema: Lo script `generate_pdfs.py` fallisce

Soluzione:

1. Verifica che Chrome sia installato:

```
which google-chrome
# oppure
which chromium
```

2. Su macOS: Verifica che Chrome sia in `/Applications/Google Chrome.app`
3. Su Linux: Installa Chrome:

```
wget -q -O - https://dl-ssl.google.com/linux/linux_signing_key.pub |
sudo sh -c 'echo "deb [arch=amd64] http://dl.google.com/linux/chrome
sudo apt-get update
sudo apt-get install -y google-chrome-stable
```

I diagrammi Mermaid non appaiono nei PDF

Problema: I diagrammi sono vuoti o mostrano solo codice

Soluzione:

- Lo script usa un timeout di 15 secondi per il rendering

- Se i tuoi diagrammi sono molto complessi, aumenta il valore di `--virtual-time-budget` in `generate_pdfs.py` :

```
"--virtual-time-budget=30000", # Aumenta a 30 secondi
```

Il bottone PDF non appare

Problema: La pagina non mostra il bottone download

Soluzione:

1. Verifica che `hooks/pdf_button.py` sia configurato in `mkdocs.yml` :

```
hooks:  
- hooks/pdf_button.py
```

2. Ricostruisci il sito: `mkdocs build --clean`

Personalizzazione

Modificare lo stile dei PDF

Edita `code/generate_pdfs.py` , sezione CSS in `html_template` :

```
# Cambia font, colori, margini, etc.  
h1 {{  
    color: #2c3e50; # Cambia colore titoli  
    ...  
}}
```

Modificare il bottone PDF

Edita `hooks/pdf_button.py` :

```
# Cambia colore, testo, icona del bottone
style="
    background-color: #2196F3; # Cambia colore
    ...
"
```

Performance

Generare PDF per 27 file markdown richiede circa:

- **Locale (Mac):** ~3-5 minuti
- **GitHub Actions:** ~5-8 minuti

Per migliorare le performance:

- Riduci `--virtual-time-budget` se i tuoi file non hanno diagrammi complessi
- Esegui solo sui commit con tag `[ci]` (già configurato)

FAQ

Q: Posso disabilitare la generazione PDF per alcune pagine?

A: Sì, modifica `code/generate_pdfs.py` e aggiungi un filtro:

```
# Escludi file specifici
if 'draft' in str(md_file) or 'private' in str(md_file):
    continue
```

Q: Posso personalizzare il nome del PDF?





A: Sì, modifica la logica in `generate_pdfs.py`:

```
# Invece di:
pdf_name = str(rel_path).replace('/', '_').replace('.md', '.pdf')

# Usa:
pdf_name = f"MyPrefix_{str(rel_path).replace('/', '_').replace('.md', '.pdf')}
```

Q: Funziona con plugin MkDocs custom?

A: Dipende. I PDF sono generati dal markdown grezzo, quindi:

-  Mermaid: Sì
-  Admonitions: Sì (vengono convertiti in HTML)
-  Tabelle/Liste: Sì
-  Plugin che generano contenuto dinamico: Potrebbero non funzionare

Crediti

Sistema sviluppato per il progetto **infohandouts** utilizzando:

- MkDocs Material
- Chrome/Chromium Headless
- Mermaid.js
- Marked.js

Per problemi o suggerimenti, apri una issue su GitHub.