

Didattica deproprietarizzata

Breve guida a quarto

Ospitare un libro interattivo su Codeberg

Luca Baldini

<https://unipico.codeberg.page>

Indice

Prefazione	3
Licenza	4
Tu sei libero di:	4
Alle seguenti condizioni:	4
Note:	5
1 Iniziare un nuovo progetto	6
1.1 Creazione del repository	6
1.2 Modifiche iniziali	7
1.2.1 Scelta della lingua	7
1.2.2 Scelta della licenza	7
1.3 C'è altro?	7
2 Struttura del repository	8
2.1 Descrizione dei file	8
2.1.1 Testo e figure	8
2.2 Controllo delle versioni	9
3 Composizione	10
4 Rilascio	11
4.1 Rilascio manuale	11
4.1.1 Configurazione iniziale	11
4.1.2 Workflow di rilascio	11
Appendici	13
A Cronologia delle versioni	13
1.1.0 (2026-10-02)	13
1.0.1 (2026-10-01)	13
1.0.0 (2026-10-01)	13

Prefazione

Versione 1.1.0 ([cronologia](#))

Questa è una breve guida per pubblicare su Codeberg Pages libri interattivi realizzati con [quarto](#) a partire dal template [quarto_template](#).

Buon divertimento!

Licenza

Tutto il materiale dell'opera è distribuito con licenza *Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)*.



© 2026 Luca Baldini

Tu sei libero di:

Condividere — riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare questo materiale con qualsiasi mezzo e formato per qualsiasi fine, anche commerciale.

Modificare — remixare, trasformare il materiale e basarti su di esso per le tue opere per qualsiasi fine, anche commerciale.

Il licenziante non può revocare questi diritti fintanto che tu rispetti i termini della licenza.

Alle seguenti condizioni:

Attribuzione — Devi riconoscere una menzione di paternità adeguata, fornire un link alla licenza e indicare se sono state effettuate delle modifiche. Puoi fare ciò in qualsiasi maniera ragionevole possibile, ma non con modalità tali da suggerire che il licenziante avalli te o il tuo utilizzo del materiale.

StessaLicenza — Se remixi, trasformi il materiale o ti basi su di esso, devi distribuire i tuoi contributi con la stessa licenza del materiale originario.

Divieto di restrizioni aggiuntive — Non puoi applicare termini legali o misure tecnologiche che impongano ad altri soggetti dei vincoli giuridici su quanto la licenza consente loro di fare.

Note:

Non sei tenuto a rispettare i termini della licenza per quelle componenti del materiale che siano in pubblico dominio o nei casi in cui il tuo utilizzo sia consentito da una eccezione o limitazione prevista dalla legge.

Non sono fornite garanzie. La licenza può non conferirti tutte le autorizzazioni necessarie per l'utilizzo che ti prefiggi. Ad esempio, diritti di terzi come i diritti all'immagine, alla riservatezza e i diritti morali potrebbero restringere gli usi che ti prefiggi sul materiale.

1 Iniziare un nuovo progetto

Questo capitolo è una breve guida a come creare da zero un nuovo testo interattivo da ospitare su [Codeberg](#) utilizzando [quarto](#).

1.1 Creazione del repository

Quando crei un nuovo repository nel tuo spazio utente di Codeberg, hai la scelta di partire da un template: seleziona `unipico/quarto_template`, spuntando *Git content (default branch)* tra i *Template items*.

A questo punto puoi clonare il repository localmente ed iniziare a lavorare nella tua *working copy*, e.g., per questa guida

```
git clone ssh://git@codeberg.org/unipico/quarto_guide.git
cd quarto_guide
git status
```

(Ovviamente devi modificare opportunamente l'url in modo da puntare al tuo nuovo repository appena creato.)

Il repository dovrebbe già essere completamente funzionale, e puoi visualizzare un'anteprima sul tuo browser facendo semplicemente

```
quarto preview
```

oppure utilizzando l'apposito target del Makefile

```
quarto preview
```

i Nota

Va da sé: è necessario avere quarto installato sul tuo computer. Puoi partire da [qui](#).

1.2 Modifiche iniziali

Il template su Codeberg è costruito in modo da minimizzare il numero di cambiamenti necessari per cominciare a lavorare sul materiale, ma non fa magie. (Non ancora, per lo meno.)

Ci sono alcune modifiche da fare prima di cominciare, ed in particolare:

- nel file `_quarto.yml`: devi aggiornare opportunamente i campi `${TITLE}`, `${SUBTITLE}` e `${AUTHORS}`—quest'ultimo due volte;
- nel file `chapters/cc-by-sa-it.qmd` (o `chapters/cc-by-sa-en.qmd`, a seconda della lingua) devi aggiornare il campo `${AUTHORS}`.

1.2.1 Scelta della lingua

Se il progetto è in italiano non c'è molto altro da fare, a parte rimuovere dal repository il file di licenza in inglese (che in questa configurazione non è usato) se proprio tieni all'economia concettuale.

Se, viceversa, il progetto è in inglese, allora devi modificare ulteriormente il file `_quarto.yml`, cambiando in `en` tutte le istanze di `it`. Dovrebbe essere semplice. (Ovviamente dovrai tradurre anche il poco testo già presente dall'italiano all'inglese.)

1.2.2 Scelta della licenza

Se la licenza di default ([CC BY-SA 4.0](#)) va bene non devi fare niente.

In caso contrario dovrai modificare opportunamente il file `LICENSE` e creare un nuovo file `.qmd` per rimpiazzare il vecchio `chapters/cc-by-sa-it.qmd`. (Il riferimento al vecchio file in `_quarto.yml` andrà modificato opportunamente.)

1.3 C'è altro?

Non direi. Siamo pronti per cominciare a scrivere il materiale.

2 Struttura del repository

Questo capitolo vuole essere una breve guida alla struttura del repository ed alle funzionalità che esso offre.

2.1 Descrizione dei file

Da un punto di vista del documento propriamente detto, i file principali sono:

- `_quarto.yml`: contiene la struttura di alto livello del documento, inclusi i metadati rilevanti, le impostazioni di quarto, e l'indice dei file;
- `index.qmd`: è il sorgente per la prima pagina del documento—quella a cui punta il link all'html statico;
- `meta.yml`: è un file ausiliario che contiene la versione corrente del documento e la data di rilascio associata.
- `references.bib`: è il file di bibliografia, che nella versione iniziale contiene una sola entry di esempio.

Ci sono poi alcuni file ancillari:

- `README.md`: è il README di più alto livello del repository—quello che appare nella pagina principale del repository su Codeberg;
- `LICENSE`: è banalmente il file della licenza, che dovrebbe essere coerente con la sezione corrispondente del libro;

Ultimo, ma non per importanza, il flusso di lavoro:

- `Makefile`: contiene i comandi della shell per le operazioni principali—anteprima, compilazione e rilascio.

2.1.1 Testo e figure

La struttura del repository è pensata in modo che i capitoli, e eventualmente le appendici, siano file `.qmd` separati nella cartella `chapters`, mentre le figure siano fisicamente nella cartella `figures`.

La versione iniziale del repository ha una manciata di figure nella sotto-cartella `figures/static` con l'idea che eventuali figure generate programmaticamente potrebbero essere collocate in una nuova cartella `figures/generated`, mentre gli script per generarle potrebbero andare in `figures/scripts`.

Ovviamente questa struttura di base può essere modificata a piacimento.

2.2 Controllo delle versioni

Per comodità il numero di versione e la data di pubblicazione sono contenute nel file separato `meta.yml`.

Questo file può essere editato manualmente per tracciare la versione del documento ad ogni rilascio.

Per chi usasse versioning semantico, il piccolo script python `tools/release.py` permette di automatizzare il controllo della versione. Più precisamente

```
[lbaldini@pcpi0188 quarto_guide]$ python tools/release.py --help
usage: release.py [-h] {major,minor,micro}

Release a new version of the package.

positional arguments:
  {major,minor,micro}  The version bump mode.

options:
  -h, --help            show this help message and exit
```

In pratica lo script è pensato per automatizzare le seguenti operazioni:

- incrementare il numero di versione;
- aggiornare il file `meta.yml`;
- aggiungere una riga con il numero di versione all'inizio del file `history.qmd`;
- fare il commit ed il push dei nuovi file nel repository;
- fare un tag del repository stesso con la versione corrente.

3 Composizione

La [guida](#) di quarto è il posto giusto da cui partire per tutte le informazioni sulle funzionalità offerte dal sistema e sulla sintassi di markup.

Il flusso di lavoro tipico consiste nell'aggiungere via via i file dei nuovi capitoli nella cartella **chapters**, aggiornare l'indice nel file `_quarto.yml` e dare libero sfogo alla propria creatività.

Il Makefile del repository dovrebbe illustrare in modo auto-esplicativo le operazioni di base rilevanti per il lavoro giornaliero:

- **preview**: anteprima dell'html statico nel proprio browser;
- **render**: compilazione completa di html e pdf;
- **clean**: rimozione degli artefatti;
- **publish**: rilascio su Codeberg Pages.

Più precisamente:

```
preview:
    quarto preview

render:
    quarto render

clean:
    rm -rf _book .quarto

publish: render
    ghp-import -o -f -p -b pages _book
```

4 Rilascio

Ci sono due possibilità per sincronizzare il repository con gli artefatti (html statico e pdf) visibili pubblicamente da web: automaticamente, attraverso una *action* Forgejo, oppure manualmente.

Nello spirito di massima semplicità, consigliamo la prima opzione.

4.1 Rilascio manuale

Il modo più semplice per rendere visibili gli artefatti su Codeberg Pages è pubblicare il contenuto della cartella `_book` in un branch di git chiamato `pages`, ed aggiungere un apposito *webhook* al repository che punti a questo branch.

4.1.1 Configurazione iniziale

Sulla pagina web del repository:

- vai su “Settings” (la piccola icona di una chiave inglese in alto a destra);
- selezione “Webhooks” nella barra verticale di navigazione a sinistra;
- premi il bottone “Add webhook” e selezione “Forgejo”;
- imposta lo url di destinazione come `https://${USER_NAME}.codeberg.page/${REPO_NAME}/` (sostituisci il nome utente ed il nome del repository opportuni);
- imposta il campo “Branch filter” a `pages`
- salva premendo il bottone “Add webhook”.

Tutto questo va fatto solo la prima volta.

4.1.2 Workflow di rilascio

Il modo più comodo per effettuare il rilascio è utilizzare il tool [ghp-import](#). Questa piccola utility è pensata essenzialmente per copiare il contenuto di una cartella—in questo caso l’html statico ed il pdf in output da quarto—in un branch di git. Esattamente quello che ci serve.

Nota

Se non hai già `ghp-import` sulla tua macchina dovrai installarlo la prima volta. Come tutti i pacchetti Python, `ghp-import` si fa banalmente con `pip`:

```
python -m pip install ghp-import
```

Il flusso di lavoro tipico per un rilascio manuale è

```
quarto render  
ghp-import -o -f -p -b pages _book
```

ovverosia:

1. rigeneriamo completamente l'html statico ed il pdf in uscita;
2. copiamo il contenuto sul branch `pages` e facciamo il push dei cambiamenti sul repository remoto.

Per la precisione, il significato delle opzioni di `ghp-import` è più o meno

- `-o`: forza un nuovo commit senza la history precedente;
- `-f`: forza il push al repository;
- `-p`: fai il push sul repository remoto dopo il commit;
- `-b pages`: specifica il nome del branch di destinazione.

Avviso

`ghp-import` rigenera completamente ogni volta il branch di destinazione, sostituendo l'intero contenuto e riscrivendo la history. Per questo particolare caso d'uso è esattamente quello che vogliamo, ma ricordate che di non editare manualmente il branch `pages`, perché qualsiasi cambiamento andrà perduto al prossimo rilascio.

Tutti i dettagli nella documentazione di [ghp-import](#).

Se questo sembra complicato, la buona notizia è che il `Makefile` nel repository template ha un target `publish` che fa esattamente questo, per cui il flusso tipico di rilascio potrebbe essere semplicemente

```
make publish
```

Il nuovo contenuto dovrebbe apparire negli url opportuni, che sono quelli pre-compilato nel file `README.md` del repository.

A Cronologia delle versioni

1.1.0 (2026-10-02)

- Aggiunta la copertina, con un design semplice e riadattabile.
- Makefile arricchito con alcuni nuovi target, incluso quello per creare la cover in formato png dal pdf compilato.
- Pull requests: [PR 1](#)

1.0.1 (2026-10-01)

- Piccoli fix per un link errato ed un riferimento duplicato.

1.0.0 (2026-10-01)

- Versione iniziale della guida.