Contributor guidelines for translations [DRAFT]

This document provides step-by-step guidance for contributors translating tutorials and documentation hosted in our GitHub repository. This guide is suitable for users with little or no experience using GitHub, explaining how to access your assigned translation branch, edit Markdown files, maintain the correct structure and formatting, and submit your work for review. By following these guidelines, you'll help ensure that all translated tutorials remain consistent, accurate, and easy to integrate into the main collection.

Purpose

We use this GitHub repository to host tutorials and translations.
Each language translation lives in its own **branch** (e.g., `es, fr`, etc.).
You'll be assigned a branch for your language before you begin.

Workflow overview

The steps within this workflow will guide you through the whole process for contributing. The workflow is set up in a way that it consists of three simple stages, 1.- grant you access to the branch that you will work on; 2.- for you the contributor edit all the documents inside the branch following the described details; 3.- the moderator to review and give you feedback or final approval.

Click on this link for a graphical view of the workflow.

- **Get access**
    - You will receive write access to your assigned translation branch (e.g., `translation-french`). If you do not have the write access, please contact the AlphaFold community space moderator (contact email).
    - In GitHub, switch to that branch using the dropdown near the top-left of the repository view.
    - You can use GitHub's web interface – no command line is required.
- **Locate the tutorial files**
    - Each tutorial page is a Markdown (`.md`) file.
    - Pages are stored in the `root` folder.
- **Translating the tutorial(s)**
    - Translate each markdown page from English to the target language
    - Translating images (you can translate figure legends but not images)
    - Don't change the file names or folders
    - Keep the numbering of files and folders as given
    - Translate all text including:

        - Section headings
        - Paragraphs and bullet points
        - Captions or image alt text
    - Images
        - If a new image needs to be uploaded then upload the images in the /images folder and link them in your content.
        - For linking the uploaded image, use the following markdown syntax (assuming the AlphaFold.png has been uploaded to the images folder)
            - `![AlphaFold logo](/images/AlphaFold.png)`
    - Interactive content created with H5P app (e.g. quizzes)
        - Keep the interactivities as they are in the English main branch. However, you may add a short paragraph, below the interactivity content, in your language to explain what the interactivity is about.
    - Translate metadata
        - Translate the metadata inside 'language'.yml file inside the _data folder
            - Inside the 'language'.yml there are labels that need to be filled out to acknowledge your contributions. These are:
            - Translated_ny_name, name, profile_url and orcid
            - Translate all the labels in this file (everything inside quotation marks)
    - Saving the files
        - Save the file with .md extension
        - Don not change its name, numbering or extension
        - When saving a file for the first time the system will ask you to save your changes to a separate branch, accept it and save all your changes in one separate branch
            - Note: from now on save all further changes to the new branch that was created on the first time.
        - You can do the translations in multiple rounds. To do this please use the same branch that you created in the first round of the translation process.

- **Check formatting**

    - Add the following markdown on the top of each markdown file

```
---

layout: default

title: PAGE_TITLE (note: use actual page title here)

---
```

    - Follow the formatting from the English version of the tutorial

    - Preserve Markdown formatting (e.g., `## Headings`, `* bullet lists`, `[links](url)`).
    - Maintain paragraphs and line breaks

- **Raise a Pull Request (PR)**
    - Once the translation is complete, open a **Pull Request** (PR) from your branch to the `language branch`.
    - In the PR description, include:
        - Language of translation
        - List of translated tutorials
        - Any notes for reviewers
- **Review process**
    - A moderator will review your PR for completeness and accuracy.
    - You may receive comments or suggestions — please update accordingly.
    - Once approved, your translation will be merged and published.

Important Rules

- **Raise the Pull Request only for the target language branch and not for the main or other protected branches**
  You will work only in your assigned translation branch.
- **While working on all the files that need to be translated may require multiple rounds, only do the Pull Request when all the files have been finished.**
  Partial translations may cause confusion to the moderator.
- **Do not rename or move files or other folders** unless explicitly instructed.

Quality Checklist Before Submitting

- The entire tutorial is translated (not partial).
- All links and formatting are preserved.
- Code blocks and file paths are unchanged.

- PR targets the correct translation branch.
- Translation is consistent with terminology used in previous tutorials.

Helpful Tips

- You can preview Markdown rendering directly in GitHub's editor.
- Use a translation memory or glossary to maintain consistent phrasing.
- If unsure about a technical term, leave it in English.

Need Help?

If you encounter any issues (branch access, merge conflicts, or file confusion), please contact the repository maintainer via GitHub Issues or email.

A reference table for terms used in GitHub:

Term	Meaning
Repository (Repo)	The main project space on GitHub — like a shared folder that holds all files, tutorials, and translations.
Branch	A separate version of the repository. Each translation has its own branch (e.g., `translation-french`) so that work can happen safely without changing the main version.
Main Branch (`main`)	The official version of the project. Only reviewed and approved work is added (merged) here.
Directory (Folder)	A folder inside the repository that holds related files. Example: `Practicals/` might contain several Markdown pages.
File	A single document in the repository, often a `.md` (Markdown) file such as `1.Introduction.md`.
Markdown (.md)	A lightweight text format used for writing web pages on GitHub. Uses simple symbols like `#` for headings and `*` for lists.
Commit	Saving your work in GitHub. Each commit keeps a record of what changed and who changed it.
Commit Message	A short note describing your change. Example: “Translated Introduction page to Spanish.”

Push	Sending your saved (committed) work to GitHub so it appears online. (If you use the web editor, this happens automatically when you click **Commit changes**.)
Pull Request (PR)	A request to merge your work into another branch — usually asking for review before adding your translation to the main version.
Merge	Combining your branch's changes into another branch (for example, adding your translated files into `main`).
Issue	A message or discussion post inside the repository. Use it to report problems or ask for help.
Maintainer / Moderator	The person or team responsible for managing the repository, reviewing submissions, and approving merges.
Preview	A tab in GitHub's editor that shows how your Markdown page will look when published.
Practicals/	An example of a *directory* (folder). The `/` at the end means it's a folder, not a file.