

Github: for adding/structuring content

About Github

Git is a version control management software that is used on Github.

- Version control software allows you to have “versions” of a project, which show the changes that were made to the code over time, and allows you to backtrack if necessary and undo those changes.¹
- This means that any changes that are made (see below) can be “undone” by reverting to a previous version. So it’s pretty hard to break things!
- [Here’s](#) a short video to learn more about version control.

Helpful links

- [An overview of your Github repository](#) - high-level view of what different parts of the page mean
- [Glossary](#) of Github-related terms

Making changes

Overview

[General overview: using Github in the browser](#) (tl;dr for the below)

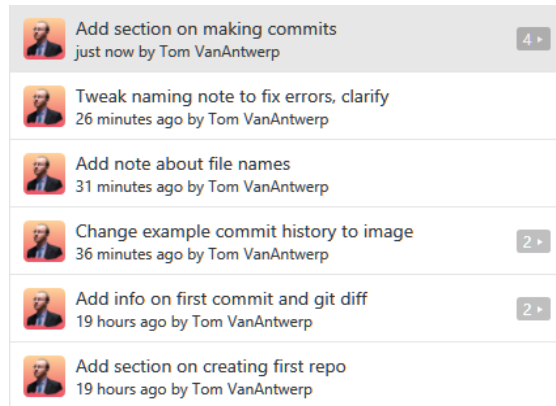
- If you’re wondering “how do I do X thing?”, this probably has your answer.

Changes are made in [branches](#) of a master file.

- The master branch is what shows up when viewing the tool online. **Making changes directly to the master branch will impact the tool as it shows online for users. Generally this is avoided to ensure there are no errors.**
- Creating a branch allows you to make specific changes that can then be reviewed for addition to the master branch. (Learn more [here](#)).
- Changes to the branch are called [commits](#)
 - These can be adding lines of text, a file, etc.

¹<https://www.makeuseof.com/tag/git-version-control-youre-developer/>

- Generally, they're frequent and named simply (see below for list of sample commits)



- Once you've committed your changes to the branch, you can create a [pull request](#).
 - A pull request notifies the admin that your branch (your changes) are ready for review. The admin can merge your branch (your changes) into the master branch, and the changes will appear on the live tool. (More about pull requests [here](#)). Admins can approve their own pull requests.
 - [How to review a pull request](#)
 - [How to approve a pull request](#)
- You can view any additions, deletions, or other changes to each file in a pull request. You can also view these by looking at the commit history of the project if you're trying to target a specific problem or revert back to an earlier stage.
 - When you choose a commit from the list, you can see what files have been changed on the right. The colored squares to the far right of a file's name indicate how much has been added versus how much has been removed from that file. Green equals additions, and red equals subtractions. If you click the arrow to the immediate left of a file's name, it will show you the exact lines of the file that have changed. (Note: this doesn't work for files that are not just lines of text, such as an image file.)²

Adding text content

Follow [these steps](#) to add or remove text content, making sure to create a pull request (Step 7). Use the below guidelines for working with YAML.

Editing is easiest using [Sublime Text](#) (free!) — it controls indentations and is visually easier to manage than Github.

1. Download `en.yml` file from Github
2. Edit in Sublime Text
3. Re-upload `en.yml` file to Github

YAML

- [Overview of keys, values, and hierarchy \(i.e., why indentations matter and what they mean\)](#)
- **YAML is case sensitive**
- Don't use the **tab** key - indent using spaces

² <https://github.com/tvanantwerp/github-for-non-programmers/blob/master/03-commits/03-01-new-repo-first-commit.md>

- Keys (the thing before the colon) are written in “camel case” with no spaces in between words
 - sampleKey: “value”
 - secondSampleKey: “value”
- Values (the thing after the colon) should usually be surrounded by single “ or double “” quotes.
 - sampleKey: “This is a value, too!”
 - sampleKeyIcon: “icon-url”
 - *Note - when adding images (icons, logos), you don’t need to add the extension afterward in the value (.jpg, .svg)*
- Whenever you see the # character anywhere in the file that isn't first enclosed in " (quotes) or "" (double quotes), it marks the beginning of a comment. This means all the text after it, up until the end of the line, is completely ignored and has no effect. You can use this to write notes on the file or temporarily disable one or more options.³
- To span multiple lines, use a pipe (|) symbol to insert new lines between values. Use a greater than (>) symbol to combine all of the text onto a single line.⁴

```
new_lines: |
    line 1
    line 2
    line 3

my_last_words: >
    I hope my wife doesn't mind
    I'm exposing her
    Instagram addiction
```

- Items marked as **# Do Not Translate** are file paths and URLs that have the potential to be changed if they go through a translator. You can prevent this on LingoHub, but having a **# Do Not Translate** comment can be helpful to remember which items need to be marked as “untranslatable” on LingoHub.

If you’re encountering an error, try pasting your addition/code [here](#) to check for incorrect syntax.

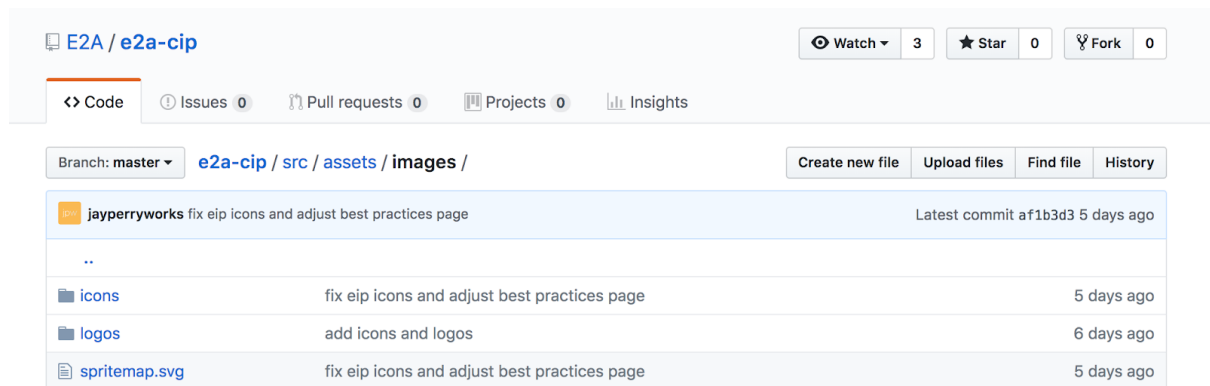
Adding logos, icons

Add icons and logos to their respective folders by navigating as follows:

```
e2a-cip > src > assets > images
```

³ <https://github.com/Animosity/CraftIRC/wiki/Complete-idiot's-introduction-to-yaml>

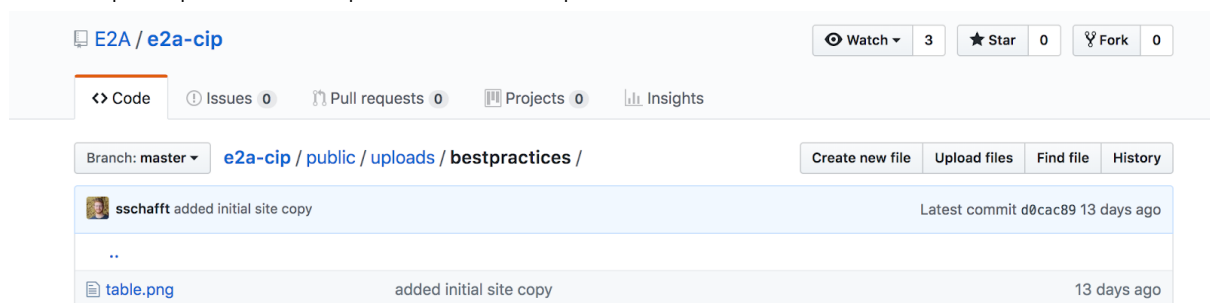
⁴ <https://searchitoperations.techtarget.com/tip/Learn-YAML-through-a-personal-example>



Note: icons should be in .svg format for optimal quality

Adding inline images for best practices

e2a-cip > public > uploads > bestpractices



Sub-folders can be made in the `bestpractices` folder and referenced using the YAML hierarchy explained in the link above. An example of how to insert an image inline can be seen in the `en.yaml` file (copied in screenshot below— see lines 217+)

```

201     bestPractice2:
202       title: "Expanding the range of contraceptive methods available to adolescents\
203         \ and young people (including LARCs and DMPA-SC)"
204       # Do Not Translate
205       icon: eip-contraceptives
206       # You must upload each checklist in the public/uploads/checklists folder using the file name indicated below
207       checklistName: ''
208       teaser: |-
209         Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ergo ita: non posse honeste vivi, nisi honeste vivatur? Nos v
210       body: |-
211         Lorem ipsum dolor sit amet, consectetur adipiscing elit. Inducant pueri, in quibus ut in speculis natura cernitur. Sin
212
213         Lorem ipsum dolor sit amet, consectetur adipiscing elit. Estne, quales, inquam, sitienti in bibendo voluptas? Hoc Hier
214
215         At ego quem huic antepone non audeo dicere; Nam si amitti vita beata potest, beata esse non potest. Ne amores quidem
216       resources:
217         # Add resources below copying the same format and changing the number appended to resource (ie resource1, resource2 et
218         resource1:
219           title: Resource Title
220           teaser: |-
221             Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut aliquid scire se gaudeant? Laelius clamores sofòw ille
222           # Do Not Translate
223           url: /uploads/resources/bestpractice1.png

```

Adding resources, country indicator datasets, and checklists

Navigate as follows:

e2a-cip > public > uploads **(then navigate to the necessary folder)**

Branch: master ▾


e2a-cip / public / uploads /

Create new file

Upload files


Find file


History


 **sschafft** init example checklist


Latest commit f29d23a 18 hours ago

..

 [bestpractices](#) added initial site copy 14 days ago

 [checklists](#) init example checklist 18 hours ago

 [country_indicators](#) adjust indicator to use new location and data structure 18 hours ago

 [resources](#) added initial site copy 14 days ago

Naming conventions

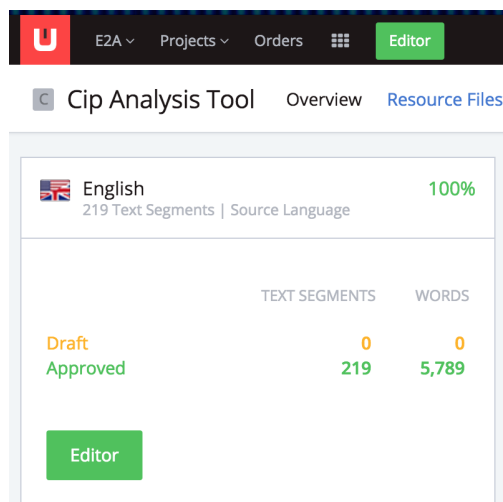
- Avoid using spaces and symbols (#, etc) in file names. Separate words with dashes, underscores, or other conventions — sticking to one convention limits the possibility of user error when working in YAML
- Name branches according to the larger task you’re working on (e.g., “icon-updates”)
- Name commits according to the individual changes you make (e.g., “added new icon for best practice 4”)

LingoHub: for editing content

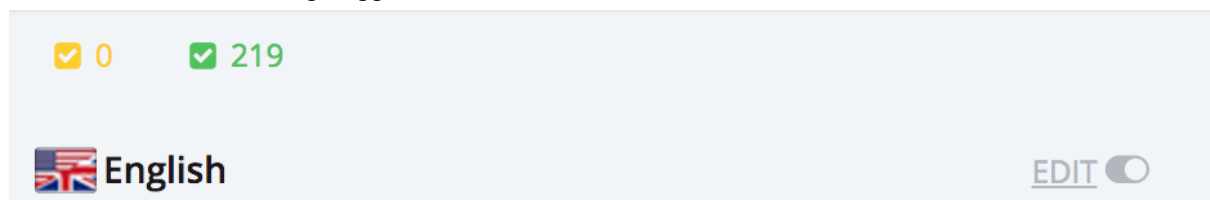
General editing

- [Quick LingoHub overview](#)
- [More comprehensive FAQ](#)

Edit content from the `en.yaml` file by clicking the “Editor” button.



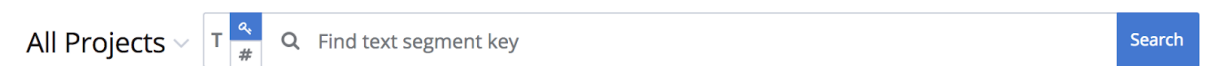
To enable / disable editing, toggle the switch next to “Edit”.



To format text (bold, adding headings, italics, etc), use [Markdown](#) syntax. This can be done easily in [iA Writer](#).

Search

To search for a certain key value (e.g., `downloadMacUrl`), click the key icon in the search bar before entering your term.

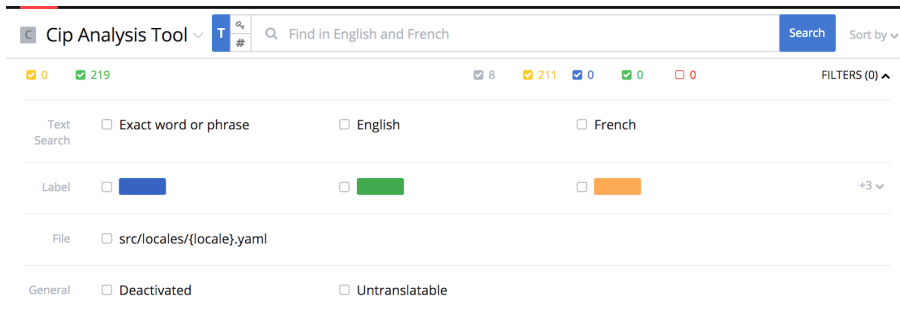


Note that the enclosing folder (in this case, `home`) is indicated in the key name followed by a ‘

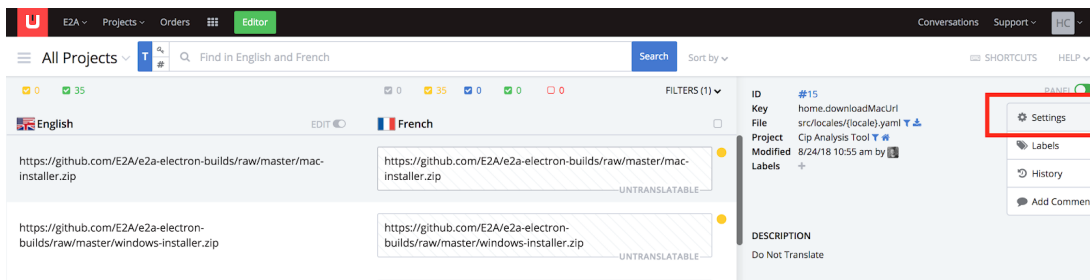


Untranslatable values

Values that are “untranslatable” will only be able to be edited by admins. These can be viewed (and edited by admins) by clicking “Filters” and ‘Untranslatable”.



To toggle a value’s translation permissions, click the value, then “Settings” to the right of the screen.



In the popup box, (un)check the “Translate” checkbox to change its permissions.

#15

Settings

×

Text segment key

home downloadMacUrl

Description

Do Not Translate

Check for character length

min. max.

What's this?

☐ Translatable

If unchecked, translators and reviewer won't be able to change this translation.

☒ Active

If unchecked, this text segment cannot be translated and won't be included in resource files. Find deactivated translations using the 'Deactivated' filter option in the editor.

Cancel

SAVE CHANGES