

Guidelines

Basic edition

1. Check the [DokuWiki syntax](#)
2. You can try it out in the playground [playground](#)
3. Remember that the Wiki is free to access by anyone
4. Remember to always sanitize :
 1. your hands (because of covid-19), and
 2. your data from any sensitive information (password, identities, etc.)
5. It is recommended to use the DokuWiki syntax. However, if you already have documentation in markdown and you cannot “translate” to DokuWiki syntax, then you can try to paste it between `<markdown>` and `</markdown>` elements. Please avoid using this as much as possible.
6. When something is incomplete or To Do, please use  :

```
FIXME
```

Pictures and other media

If you are working with pictures

1. If an image is in your clipboard (screenshot), you can paste it in edit mode (see [imgpaste plugin](#) for more details).
2. If it's in a file, you have to upload it using the media manager. Also make sure you are in the correct namespace.

Create a new page

- Try to use underscore '_' as a separator in pagenames.
- Make sure you are creating the page in the correct namespace
- For example : “How to run a job” in namespace hpc-admin:
 - in the URL bar `baseURL/hpc-admin/how_to_run_a_job`, or
 - with the DokuWiki link syntax

```
[[hpc-admin:how_to_run_a_job]]
```

Browse vs. search vs. TOC

- You can find information with the search engine, but some people prefer to follow links or to read a table of content. So do all three !
- Ideally, try to provide links to all the other pages from your namespace in the start page.
- You can display the TOC of another page in any page. You can also create a page that will contain the TOC of multiple other pages. Just manually add a METATOC to all other pages you

want to see.

- For example: <https://doc.ererech.unige.ch/hpc/toc>
- TOC of the page `faq` in namespace `hpc` titled `FAQ` :

```
{ {METATOC 1-5 >:hpc/faq|FAQ}}
```

Synthesize and summarize

- Try to **synthesize** and give the right amount of explanation
- In case the procedure is too long, please use the following sub-sections :
 1. **TL;DR** - short version, for example once we know how to use it
 2. **Details** - detailed explanation, example, etc. when we need to understand how it works or how it all started.
- All references (URL, Gitlab issues, msgid, etc.) should be marked as a footnote:

```
((my reference in a footnote))
```

- Please don't put explanation in the footnote, only reference (URL,msgid...) to avoid having to read back and forth.
- You can see this [in this example](#), where the TL;DR section contains links because they can be used during the procedure, but the all other references are footnotes.
- If something needs basic knowledge of something “external” or a pre-requisite, please add a “Reference” section for example with links (i.e. : Salt stack documentation, How to use Linux for beginners, etc.).

Cleanup your outputs

- Remove any extra spaces, typos or commands that are not required to understand what is going on (especially in console outputs).

code : scripts, console output and commands

- When you give some example of code (scripts, console output or commands), please use the right format
- For source code, scripts or a list of command, use the code block and specify the language (if available), such as `php`, `bash`, etc.
- If you refer to a script name or a path or a simple command, use two simple quotes so it stays inline.
- For console outputs, please use the code block with language `console`
 - N.B. : this is a simple CSS style that turns the text white and the background black.

```
<code console>
```

From:

<https://doc.eresearch.unige.ch/> - **eResearch Doc**

Permanent link:

<https://doc.eresearch.unige.ch/guidelines?rev=1592933584>

Last update: **2025/06/11 12:27**

