

Jekyll

focus on
practice

Collections

Group related content like members of a team or talking at a conference

Setup

- define a collection in the "_config.yml"
- define as a hashmap (hashmap can specify metadata; output: true metadata)
- ↳ will be rendered default
- can specify a directory to store the collection

• ↳ collection_dir: my_collections

directory: my_collections/_books

~~my_collections~~ books is the collection

collections:

staff_members:

people: true

output: true

collections_dir: my_collections

→ specify a collection with a hashmap

→ ~~generate and write output to the site~~

→ specify collection directory

↳ will output every staff page as a single page or, just as a variable

directory

.../jekyll

↳ ~~collection_dir~~

↳ my_collections

↳ _books

books and recipes are the collection's name

✓ recipes the name must start with "_"

the name of collection can not start with "_"

Collection is not the post template
collection is the collection

post 只需要有一个有效的文件名就会被渲染
即使没有一个 front matter

collection Permalinks

↳ 在URL中的部分

collection Custom Sorting (> 4.0)

collections:
 tutorials:
 sort-by: lesson

指定排序字段。该字段存在于 front-matter 中。
① 默认情况下使用 date 字段排序
② 可以指定其他字段排序
③ manually ordering Document

order:
 - hello-world.md
 - introduction.md
 - basic-concepts.md
 - advanced-concepts.md

if "sort-by" defined both, "order" take precedence

~~if the "sort-by" and the "order" metadata both defined, "order" list takes precedence~~

● Collections' Liquid Attributes

site.collections ← _config.yml

label → my-collection

docs → an array of docs

files → an array of static files in the collections

relative directory → view path to collection's source directory, relative to the site source

output → whether the collection's documents will be output as individual files

• the "posts" hard-coded collection

site.collections 中包含 "posts" 这个特殊的 collection

可以 filter 实现过滤 `{{ site.collections | where: 'label', 'myCollections' | first }}`

• more fine-grained control over ~~the~~ document being output to disk

- published: false (default is true)

• Document attributes

content → unrendered content (所有标签 "front matter" 过滤掉)

output → rendered content based on "content"

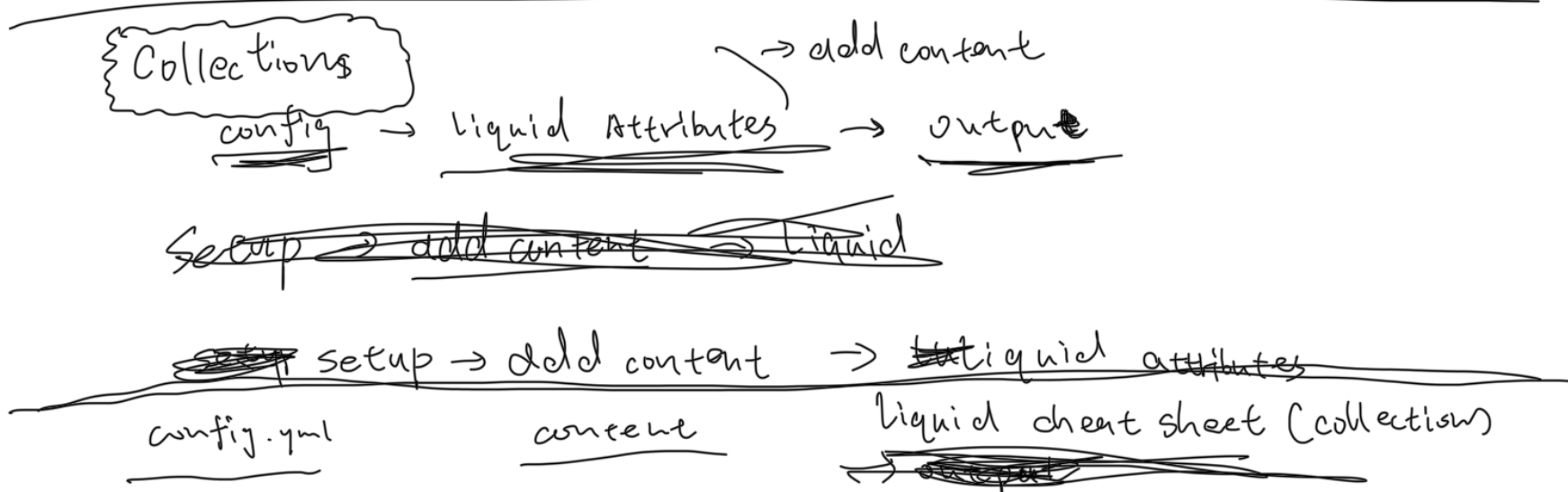
path → the full path to the document's source file

relative path → path relative to the site source

url → the url of the rendered collection (~~not~~ the doc's url???)

collection → the name of the collection

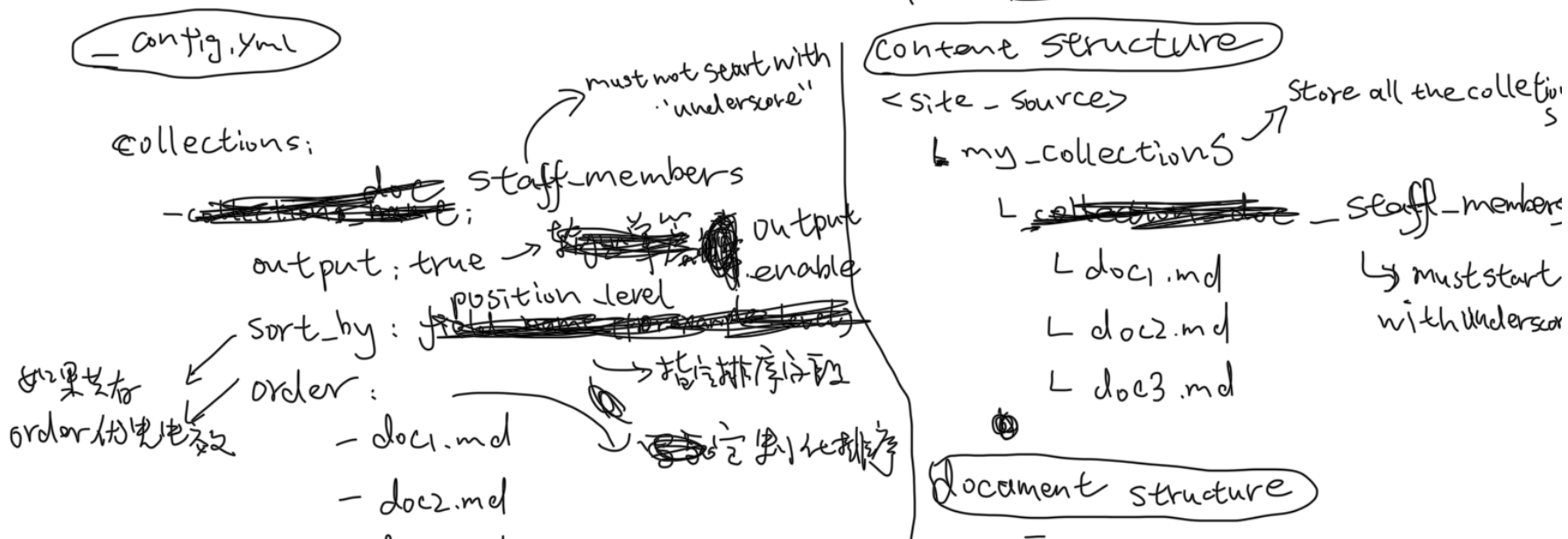
date → the date of the collection



the collection is different from the "tag system"

the collection is different from the "posts", its content is not apply the "post" template

~~template~~



- doc3.md
permalink: '/~~the first name~~' staffs "

collectionsdir: my_collections

Liquid Attributes

• Site.collections (iterik)

- label - directory
- docs - output
- files
- relative_directory

"posts" is a default collection. You can filter your collection use the below code

{{ site.collections | where: "label", "staff_member" | first }}

• Document (item.xxx)

- content - relative_path
- output (content) - url
- path - collection
- date

name: Joe
date: 2023-10-10
position_level: 10 → use for "sort by"
published: false → fine-grained control output

Joe is a senior engineer → the
iter the collection compilation content

There is not a single page list the collections, or list the docs of a collections, unless you make one, usually the list of collections for the list of the doc of a collection will be emmed in a template, for example in a index.html

the code

2023/12/14:05

Pages

- the most basic building block for content
- useful for standalone content.
- for example (index.html, about.html, contact.html) typical usage
- ext suffix is md, html, for normal
- can set a permalink
- can enable the excerpts

configs for pages, file structure

_config.yml

~~excerpt~~

page-excerpts: true

<site-source>

L about.md

L contact.md

L index.md

L some_folder

L some_pages.md

L some_page2.html

page content demo

title: a page's title

date: 2022-10-10

permalink: /some-page/

Some page content ...

Posts

- must begin with "front matter"
 - specify a layout
 - specify some metadata (title, date, author)
- Include images and resources
 - ! [an image] @ (/assets/image/1.png)

[get the pdf] (/assets/1.pdf)

- index of posts

- iter the "site.posts", ~~post.url, post.title~~

- post-title, post-url

- tags

- tag, or tags key in the frontmatter

- (define tag in the frontmatter)

- tag: a

- tags: a b → usually use tags

-

- category define category

- ~~•~~ in the front matter

- c

- categories: a b → always use categories

- category: a

-

~~always use~~

- define category by a post's file path

/movies/ - posts / 2023-10-10-forrest-grump

↳ the tag!!!

- the tags can ~~not~~ be use for generating URL as a variable, but category can.

- post variables

Site.posts

- post.url

- post.title

- post.excerpt

- post.author

- posts ~~postion~~ folder

<Site source>

└ _posts

└ 2023-10-10-forrest-grump.md

Typical posts folder

<site-source>

└ _posts

L 2023-10-10 - forrest-gump.md

L music

L -drafts

L -posts

- draft-post.md

will be used as ~~the post~~
the category of this post

L 2023-10-10 - my-heart-will-go-on.md

L -drafts

- draft-post.md

2023-10-10-forrest-gump.md

title: forrest, run, forrest!

date: 2023-10-10

layout: post

tags: ~~xxx~~ inspirational, xxx, xxx

categories: 2023-movie

only can define by the front
matter
can also define in the
front matter

[[{"image": "/assets/forrest.png"}]]

[[{"pdf": "/assets/forrest.pdf"}]]

This is a great movie.

index of posts

the web

index of tags

index of categories

Liquid variable(posts)

Site.posts Site.tags Site.categories
- post.url - tag[0], name - category[0]
- post.title - tag[1], posts - category[1]
- post.excerpt

	tags	categories
define	by front matter	1. by front matter 2. folder name
use a variable	absolutely same	
generate url	not used in generated url	can used in generated url as variable "category"