

hexo

kevinluo

Contents

1	install	2
1.1	hexo module 安裝	2
1.2	hexo blog init	2
1.3	configuration	3
1.4	Commands	3
1.5	Migration	3
2	Basic Usage	3
3	Customization	3
4	theme 分享	3
4.1	幾款簡單的 theme	3
4.2	melody	4
4.2.1	melody-設置	4
4.2.1.1	fireworks, live2d 白貓 animation	4
4.3	next	4
4.3.1	next-設置	4
4.3.1.1	busuanzi_count 閱讀次數/訪問人數	4
4.3.1.2	symbols_count_time 文章字數統計與閱讀時長	5
4.3.1.3	hexo-Next 修改內容區域的寬度	5
4.3.1.4	實現點擊出現桃心效果 animation	5
4.3.1.5	目錄欄鏈接選中顏色	6
4.3.1.6	custom path	6
4.4	Maupassant - Hexo 最簡潔主題 - cho	6
4.4.1	安裝	6
4.4.2	設置	7
5	hexo plugin 插件	8
5.1	Generate flowchart diagrams for Hexo.	8
5.2	hexo-generator-feed	8
5.3	hexo-generator-search	9
5.4	hexo-symbols-count-time for next theme	9
5.5	hexo-generator-category	9
5.6	hexo-generator-tag	9
5.7	hexo-directory-category	9
5.8	some module install	10
6	hexo 高級教程	10
6.1	腳本 Script	10
6.2	hexo 擴展	11
7	tips	11
7.1	tortoiseGit 使用密鑰，為何每次還是需要輸入用戶名密碼	11
7.2	怎麼改掉網頁底部的 COPYRIGHT 缺省內容？	11
7.3	help 網址	11
7.4	theme-front-matter	12
7.5	hexo editor 編輯器有哪些？	14
7.6	怎麼使用 yeoman 生成基礎代碼？	14
8	FAQ	15
8.1	Hexo 網站名中文亂碼	15

8.2	怎麼列出 hexo 依賴插件的完整性?	15
8.3	路徑名和分類名分別設置, 需要怎麼辦呢?	15
8.4	package.json 是什麼?	16
8.5	fontawesome 是什麼?	16
8.6	怎麼添加點擊紅心和漢字?	16
8.7	國內 JQuery CDN 有哪些?	16
8.8	怎麼解決 githubpages 不能識別下劃線開頭的目錄?	17
9	hexo deploy 網站部署	17
9.1	hexo d 法	17
9.2	直接 git clone 法	18
9.3	CI 法,	18
9.3.1	十大 CI 工具:	18
9.3.2	travis CI:	18
9.3.2.1	travis CI 配置步驟:	18
9.3.3	travis CI 配置實例	20
9.3.3.1	創建源碼新分支需要改動的文件	20
9.4	需求: 在 github pages 子目錄建立 hexo 博客	21
9.5	my deploy: kevinluolog.github.io	21
9.5.1	Repo of sites:	21
9.5.2	Repo of hexo source: private	22

目錄

1 install

Installation guide on hexo.io

手把手教你使用 Hexo + Github Pages 搭建個人獨立博客

1.1 hexo module 安裝

```
npm install hexo-cli -g
```

hexo module 安裝到 node 目錄 node_modules, 根目錄有 hexo.cmd

1.2 hexo blog init

```
$ cd d:/hexo
$ npm install hexo-cli -g
$ hexo init blog
$ cd blog
$ npm install
$ hexo g # 或者hexo generate
$ hexo s # 或者hexo server, 可以在http://localhost:4000/ 查看
```

另外還有其他幾個常用命令:

```
$ hexo new "postName" #新建文章
$ hexo new page "pageName" #新建頁面
```

常用簡寫

```
$ hexo n == hexo new
$ hexo g == hexo generate
$ hexo s == hexo server
$ hexo d == hexo deploy
```

常用組合

```
$ hexo d -g #生成部署
$ hexo s -g #生成預覽
```

現在我們打開 <http://localhost:4000/> 已經可以看到一篇內置的 blog。

1.3 configuration

hexo.io/docs/configuration

1.4 Commands

1.5 Migration

2 Basic Usage

3 Customization

Permalinks

Themes

[hexo.io-spec-配置](#)

Templates

Variables

Helpers

Internationalization (i18n)

Plugins

4 theme 分享

4.1 幾款簡單的 theme

[【Hexo】推薦 5 款簡潔美觀的主題](#)

推薦第一款。理由簡單，全文字，色調淺色系不扎眼。

1. [hexo-theme-polarbear](#)

[demo](#)

2. [hexo-theme-xoxo](#)

[demo](#)

- 3. [hexo-theme-sky](#)
[demo](#)

4.2 melody

[hexo-theme-melody-doc](#)

4.2.1 melody-設置

4.2.1.1 fireworks, live2d 白貓 animation [detail guide on melody doc](#)

- fireworks

Like the [anime.js](#) clicking effects

Set the melody.yml

```
fireworks: true
```

- Live2D Animated model pendant

install the Live2D module, which needs to be executed in the root directory of the blog through the terminal:

```
npm install --save hexo-helper-live2d
```

The corresponding module is downloaded [here](#) , For example, tororo(Cute White Cat)

copy all the files in packages to the node_modules folder in the root directory of the blog.

or install as following:

```
npm install {packagename}
```

The package name is the folder name in packages/ such as:

```
live2d-widget-model-chitose
```

```
live2d-widget-model-tororo
```

4.3 next

4.3.1 next-設置

4.3.1.1 busuanzi_count 閱讀次數/訪問人數

- 原理:

頁面植入 busuanzi 提供的 js 鏈接代碼, 在 \themes\next\layout_partials\analytics\busuanzi-counter 中

```
<script{{ pjax }} async src="https://busuanzi.ibruce.info/busuanzi/2.3/busuanzi.pure.mini.js"></script>
```

同時在相應的頁面模板加入閱讀次數等數據。其提供單頁文章和全站的字數和次數信息。next 分別把它放在頁面標題下面和 footer 底部。其渲染過程仍不清楚,

頁面標題下面:

在 \themes\next\layout_macro\post.swig 中, 祇是 `` 沒有實體, 不知什麼時候, 渲染進去的?

footer 底部, 全站的訪問人數等數據:

在 \themes\next\layout_partials\analytics\busuanzi-counter.swig 中，此處我增加了一個 theme 變量來控制

```
{%- if theme.kl_footer_eye === true %}
```

4.3.1.2 symbols_count_time 文章字數統計與閱讀時長 同樣全站的顯示的 FOOTER, 單文章顯示在文章 title 下面。

配置使能要注意：

模塊說明裏就說明，使能要在 root/_config.yml 中加入，

symbols_count_time: ## 此處定義是用來控制模塊計算的。下面的變量和next.yml中的配置一起控制字

```
symbols: true
time: true
total_symbols: false ##kl+ true
total_time: false ##kl+ true
exclude_codeblock: false ##kl+ false
```

同時在 theme 的 _config.yml 中，同樣要使能，

symbols_count_time: ## 此處定義是用來控制顯示的。klblog_ config.yml中是用來控制模塊計算的。

```
separated_meta: true ##kl+ true 換行顯示
item_text_post: true ##kl+ true
item_text_total: true ##kl+ false
awl: 4
wpm: 275
```

- 刪除 footer 底部的全站字數，時長信息。

祇要在 root_config.yml 中

```
total_symbols: false ##kl+ true
total_time: false ##kl+ true
```

- 文章字數統計與閱讀時長，取不到數據

hexo clean 一下，再編譯就可以了，原因不明。

4.3.1.3 hexo-Next 修改內容區域的寬度 如何更改內容區域的寬度？

leeze Hexo 之修改內容區域的寬度

在 \themes\next\source\css_variables\base.styl

```
//kl+ new, 參見
//當屏幕寬度 < 1600px
$content-desktop = 900px;
//當屏幕寬度 >= 1600px
$content-desktop-large = 900px;
$content-desktop-largest = 900px;
```

4.3.1.4 實現點擊出現桃心效果 animation 在 /themes/*/source/js/src 下新建文件 click.js，接着把以下粘貼到 click.js 文件中。

代碼如下：

```
!function(e,t,a){function n(){c(".heart{width: 10px;height: 10px;position: fixed;background: #
```

在 \themes*\layout_layout.swig 文件末尾 body 內添加：

<!-- 頁面點擊小紅心 -->

<script type="text/javascript" src="/js/clicklove.js"></script>

4.3.1.5 目錄欄鏈接選中顏色 copy 到

```
// Sidebar
// -----
$sidebar-nav-hover-color      = $orange;
$sidebar-highlight            = $orange;

$toc-link-color                = $grey-dim;
$toc-link-border-color        = $grey-light;
$toc-link-hover-color          = black;
$toc-link-hover-border-color   = black;
$toc-link-active-color         = $sidebar-highlight;
$toc-link-active-border-color  = $sidebar-highlight;
$toc-link-active-current-color = $sidebar-highlight;
$toc-link-active-current-border-color = $sidebar-highlight;
```

4.3.1.6 custom path

4.4 Maupassant - Hexo 最簡潔主題 - cho

大道至簡——Hexo 簡潔主題推薦

4.4.1 安裝

注：若 `npm install hexo-renderer-sass` 安裝時報錯，可能是國內網絡問題，請嘗試使用代理或者切換至[淘寶 NPM 鏡像](#)安裝。`npm install hexo-renderer-sass`

出現問題：hexo 3.8.0 用淘寶鏡像裝 hexo-renderer-sass，生成的網頁有問題，裝 hexo-renderer-scss，就沒問題了。kl: 建議盡量用 npm 來安裝。

1. 安裝主題和渲染器：

```
$ git clone https://github.com/tufu9441/maupassant-hexo.git themes/ maupassant
$ npm install hexo-renderer-pug --save
$ npm install hexo-renderer-sass --save
```

2. 編輯 Hexo 目錄下的

`_config.yml`，將 `theme` 的值改為 `maupassant`。

3. hexo-wordcount 字數統計，閱讀時長：缺省沒裝

這是在 `_config.yml` 中設置 `wordcount: true` ##kl+ `false` 時報錯的。

參考 [Hexo-文章字數統計與閱讀時長](#)

```
npm i --save hexo-wordcount
```

因缺省已經使能。下面修改不用了，可以作用改動參考

1. 在 maupassant 主題下的新建一個 wordcount.pug 文件

`themes\maupassant\layout_partial\wordcount.pug` `wordcount.pug` 文件增加內容：

```

span(class="post-time")
  span.post-meta-item-text= " | "
  span(class="post-meta-item-icon")
    i(class="fa fa-keyboard-o")
    // span.post-meta-item-text= " 字數統計: "
    span.post-count= ' '+wordcount(page.content)
    span.post-meta-item-text= ' 字 '
span(class="post-time") &nbsp;|&nbsp;
  span(class="post-meta-item-icon")
    i(class="fa fa-hourglass-half")
    // span.post-meta-item-text= " 閱讀時長: "
    span.post-count= ' '+min2read(page.content)
    span.post-meta-item-text= " 分鐘"

```

2. 在 themes\maupassant\layout\post.pug 文件中引入 wordcount.pug 文件（我自定義的位置在 busuanzi 與 Disqus 之間）

```

if theme.busuanzi == true
  script(src='https://dn-lbstatics.qbox.me/          busuanzi/2.3/busuanzi.pure.mini.js')
  span#busuanzi_container_page_pv= ' | '
    span#busuanzi_value_page_pv
    span= ' ' + __('Hits')
include _partial/wordcount.pug
if theme.disqus

```

4.4.2 設置

1. in _config.yml

```

fontawesome:
  fa-home
  fa-th categories
  fa-tags
  fa-history
  fa-user
  fa-book

show_category_count: false
wordcount: true ## 統計字數
widgets_on_small_screens: true
busuanzi: true ##kl+ false, 網頁訪問統計

menu:
  - page: home
    directory: .
    icon: fa-home

widgets:
  - search
  - category
  - tag

```


5 hexo plugin 插件

5.1 Generate flowchart diagrams for Hexo.

[hexo-filter-flowchart](#)

- install

npm install --save hexo-filter-flowchart

setting: 實測下面的不改動也可以顯示出來

Config

In your site's _config.yml:

```
flowchart:
  # raphael: # optional, the source url of raphael.js
  # flowchart: # optional, the source url of flowchart.js
  options: # options used for `drawSVG`
```

This plugin is based on [flowchart.js](#), so you can defined the chart as follow:

```
`flow st=>start: Start|past:>http://www.google.com[blank] e=>end: End:>http://www.google.com
op1=>operation: My Operation|past op2=>operation: Stuff|current sub1=>subroutine:
My Subroutine|invalid cond=>condition: Yes or No?|approved:>http://www.google.com
c2=>condition: Good idea|rejected io=>inputoutput: catch something...|request
st->op1(right)->cond cond(yes, right)->c2 cond(no)->sub1(left)->op1 c2(yes)->io->e
c2(no)->op2->e`
```

5.2 hexo-generator-feed

[github download](#)

Install

```
$ npm install hexo-generator-feed --save
```

Hexo 3: 1.x

Hexo 2: 0.x

Use

In the front-matter of your post, you can optionally add a description, intro or excerpt setting to write a summary for the post. Otherwise the summary will default to the excerpt or the first 140 characters of the post.

Options

You can configure this plugin in _config.yml.

```
feed:
  type: atom
  path: atom.xml
  limit: 20
  hub:
  content:
  content_limit: 140
  content_limit_delim: ' '
  order_by: -date
  icon: icon.png
```

type - Feed type. (atom/rss2) path - Feed path. (Default: atom.xml/rss2.xml) limit - Maximum number of posts in the feed (Use 0 or false to show all posts) hub - URL of the PubSubHubbub hubs (Leave it empty if you don't use it) content - (optional) set to 'true' to include the contents of the entire post in the feed. content_limit - (optional) Default length of post content used in summary. Only used, if content setting is false and no custom post description present. content_limit_delim - (optional) If content_limit is used to shorten post contents, only cut at the last occurrence of this delimiter before reaching the character limit. Not used by default. order_by - Feed order-by. (Default: -date) icon - (optional) Custom feed icon. Defaults to a gravatar of email specified in the main config.

5.3 hexo-generator-search

產生搜索功能，search.XML

```
$ npm install hexo-generator-search --save
```

5.4 hexo-symbols-count-time for next theme

```
$ npm install hexo-symbols-count-time --save
```

5.5 hexo-generator-category

```
$ npm install hexo-generator-category --save
```

option:

tag_generator:

per_page: 10

order_by: -date

5.6 hexo-generator-tag

```
$ npm install hexo-generator-tag --save
```

Options

tag_generator:

per_page: 10

order_by: -date

5.7 hexo-directory-category

Automatically add front-matter categories to Hexo article according to the article file directory.

Directory is means relative form article file path to Hexo source _posts folder.

[github-hexo-directory-category](#)

```
npm install --save hexo-directory-category
```

auto_dir_categorize:

enable: true # options:true, false; default is true

force: false # options:true, false; default is false

enable - Enable the plugin. Defaults to true.

force - Overwrite article front-matter categories, even if it has option categories.Defaults to false

5.8 some module install

```
$ npm install hexo-generator-search --save
$ npm install hexo-symbols-count-time --save
```

```
$ npm install hexo-generator-category --save
option:
```

```
tag_generator:
  per_page: 10
  order_by: -date
```

```
$ npm install hexo-generator-tag --save
```

Options

```
tag_generator:
  tag_generator:true
  per_page: 10
  order_by: -date
```

6 hexo 高級教程

參考

[Hexo 高級教程](#)

[hexo 脚本编写指南 \(一\)](#)

[Hexo Docs \(三\) - 高級進階](#)

[脚本需要掌握 hexo api](#)

[nodejs doc-en](#)

[nodejs debug guide](#)

[nodejs doc-zh 中文](#)

教程

[Node.js 教程](#)

[hexo-generator-category 源碼分析](#)

[hexo-generator-tag 源碼分析](#)

[hexo-generator-index 源碼分析](#)

[next 主題的模板引擎 swig 語法介紹](#)

6.1 脚本 Script

只需要把 JavaScript 文件放到 scripts 文件夾，在啓動時就會自動載入。

6.2 hexo 擴展

1. 控制臺 (Console)
2. 部署器 (Deployer)
3. 過濾器 (Filter)
4. 生成器 (Generator)
5. 輔助函數 (Helper)
6. 遷移器 (Migrator)
7. 處理器 (Processor)
8. 渲染引擎 (Renderer)
9. 標籤 (Tag)

7 tips

7.1 tortoiseGit 使用密鑰，為何每次還是需要輸入用戶名密碼

tortoiseGit 使用密鑰，為何每次還是需要輸入用戶名密碼

Q:tortoiseGit 使用密鑰，為何每次還是需要輸入用戶名密碼

A: 那個 URL 應該選擇 ssh 的，也就是以 `git@git.oschina.net:{username}/{repo name}` 這種形式的，你雖然把 SSH key 加進去了，但是你如果仍使用的是 https 方式，當然要提示輸入用戶名密碼了。如，`git@github.com:kevinluolog/hexo-klblog-src.git`

不過用 https 方式訪問，也有方法可以免手工輸入用戶名密碼的。

方法 1：直接帶入，`https://{用戶名}:{□□\protect_xunadd_text_character:nN{\textbraceright}}@github.com/{username}/{repo name}` 如，`https://kevinluolog:XXX@github.com/kevinluolog/hexo-next-muse.git`

方法 2：利用 token，當然先要創建。如，`https://gh_token@github.com/kevinluolog/hexo-next-muse.git`

7.2 怎麼改掉網頁底部的 COPYRIGHT 缺省內容？

```
\hexo\klBlog\themes\maupassant-hexo\layout\_partial\footer.pug(7): a(rel='nofollow', target=
```

```
#footer= 'Copyright © ' + date(Date.now(), 'YYYY') + ' '
  a(href=url_for('.'), rel='nofollow')= config.title + '.'
  | Powered by
  a(rel='nofollow', target='_blank', href='https://hexo.io') Hexo.
  a(rel='nofollow', target='_blank', href='https://github.com/tufu9441/maupassant-hexo') The
  | by
  a(rel='nofollow', target='_blank', href='https://github.com/pagecho') Cho.
```

7.3 help 網址

非常詳細的教程，看完照做就可以了。這個寫手做事非常細致。就象博主自己講的 -- 我走的很慢，但我從不退。

[Hexo 使用攻略 index](#)

[Hexo 博客從搭建部署到 SEO 優化等詳細教程](#)

[Hexo seo org](#)

[hexo-theme-maupassant-中文 help-大道至簡——Hexo 簡潔主題推薦](#)

[hexo 主題中添加相冊功能 <http://lwzhang.github.io/>](#)

[hexo-generator-index 源碼分析](#)

[很好的主題開發文章](#)

[Hexo 主題開發經驗雜談 org](#)

參考 hexo 渲染的事件，可以找到 generateBefore 這個鉤子 hook，祇要在這個鉤子觸發的時候，判斷一下存不存在 data files 裏的配置文件 `_data/next.yml`，存在的話就把這個配置文件替換或者合并主題本身的配置文件。Next 主題採用的是覆蓋，melody 主題採用的是替換。各有各的好處，並不是絕對的。

寫法是就是在我們的 temp 主題目錄下的 scripts 文件夾裏（沒有就創建一個），寫一個 js 文件，內容如下：

```
/**
 * Note: configs in _data/temp.yml will replace configs in hexo.theme.config.
 */
hexo.on('generateBefore', function () {
  if (hexo.locals.get) {
    var data = hexo.locals.get('data') // 獲取_data文件夾下的內容
    data && data.temp && (hexo.theme.config = data.temp) // 如果temp.yml 存在，就把內容替換掉主
  }
})
```

[Hexo 主題開發經驗雜談](#)

字體大清晰。文字 title [hexo-theme-random](#)

[Hexo 主題開發指南-random-不可能不確定](#)

7.4 theme-front-matter

在 `\source*.md` 文件的開頭

```
---
title: sublime
date: 2019-09-03 17:31:37
toc: true
mathjax: true
tags:
- 技術
- 文本編輯器
categories:
- 技術
- sublime
---
```

- 分類 frontmatter-categories

編輯文章的時候，直接在 categories: 項填寫屬於哪個分類，但如果分類是中文的時候，路徑也會包含中文。

訪問路徑是：

`*/categories/編程`

想要把路徑名和分類名分別設置，參見[路徑名和分類名分別設置，需要怎麼辦呢？](#)

- 標籤 frontmatter-tags

在編輯文章的時候，tags: 後面是設置標籤的地方，如果有多個標籤的話，可以用下面兩種辦法來設置：

tags: [標籤1, 標籤2, ... 標籤n]

tags:

- 標籤1
- 標籤2
- ...
- 標籤n

- 文章摘要 description:

首頁默認顯示文章摘要而非全文，可以在文章的 front-matter 中填寫一項 description: 來設置你想顯示的摘要，或者直接在文章內容中插入 `<!--more-->` 以隱藏後面的內容。

若兩者都未設置，則自動截取文章第一段作為摘要。

- 添加頁面 layout: 在 source 目錄下建立相應名稱的文件夾，然後在文件夾中建立 index.md 文件，並在 index.md 的 front-matter 中設置 layout 為 layout: page。若需要單欄頁面，就將 layout 設置為 layout: single-column。

- 文章目錄 frontmatter-toc:

在文章的 front-matter 中添加 toc: true 即可讓該篇文章顯示目錄。

- 文章評論 comments:

文章和頁面的評論功能可以通過在 front-matter 中設置 comments: true 或 comments: false 來進行開啓或關閉（默認開啓）。

- 數學公式 frontmatter-mathjax:

要啓用數學公式支持，請在 Hexo 目錄的 _config.yml 中添加：

1. mathjax: true

並在相應文章的 front-matter 中添加 mathjax: true，例如：

```
title: Test Math
date: 2016-04-05 14:16:00
categories: math
mathjax: true
---
```

數學公式的默認定界符是 `$$...$$` 和 `\[...]`（對於塊級公式），以及 `$...$` 和 `\(...)`（對於行內公式）。

2. mathjax2: true

但是，如果你的文章內容中經常出現美元符號“\$”，或者說你想將“\$”用作美元符號而非行內公式的定界符，請在 Hexo 目錄的 _config.yml 中添加：

而不是 mathjax: true。相應地，在需要使用數學公式的文章的 front-matter 中也添加 mathjax2: true。

- donate: donate:

enable: false ## If you want to show the donate button after each post, please set the value to true and fill the following items according to your need. You can also enable donate button in a page by adding a "donate: true" item to the front-matter.

- timeline: (layout: timeline)

網站歷史時間綫，在頁面 front-matter 中設置 layout: timeline 可顯示。

7.5 hexo editor 編輯器有哪些？

- Markdown 工具 HexoEditor

一款清新的 Markdown 工具 HexoEditor，重要的是支持 Hexo 框架

[github repo-hexo editor](#)

7.6 怎麼使用 yeoman 生成基礎代碼？

現在開始項目之前，我都會搜索一下 yeoman 有沒有庫，生成 Hexo 主題就有 generator-hexo-theme。如果還沒有安裝 yeoman，那先用 npm 全局安裝。

```
npm i -g yo
```

安裝生成器的庫：

```
npm i -g generator-hexo-theme
```

到博客目錄下，進入到 themes 目錄，創建一個用主題名命名的新文件夾，比如 test，進入新文件夾，開始生成代碼：

```
yo hexo-theme
```

然後選擇一些基本的配置，比如使用什麼模板引擎，使用什麼 CSS 預編譯等，這裏分別選擇 Swig 和 Stylus。完成之後，主題目錄下就會生成一些如下結構的文件：

```
_config.yml // 主題配置文件
languages // 多語言文件夾
layout
  archive.swig // 存檔頁模板
  category.swig // 分類文章列表頁模板
  includes // 各頁面共享的模板
    layout.swig // 頁面布局模板，其它的頁面模板都是根據它擴展來的
    pagination.swig // 翻頁按鈕模板
    recent-posts.swig // 文章列表模板
  index.swig // 首頁模板
  page.swig // 頁面詳情頁模板
  post.swig // 文章詳情頁模板
  tag.swig // 標籤文章列表頁模板
source
  css
    theme.styl // 主題自定義 CSS 文件
  favicon.ico
  js
    theme.js // 主題 JavaScript 文件
```

在 Hexo 的主配置文件中使用的舊主題，到博客根目錄下找到 _config.yml 文件，找到 theme 行，修改如下：

```
theme: test
```

hexo s 啟動博客，到瀏覽器看效果。

8 FAQ

8.1 Hexo 網站名中文亂碼

因為站點配置文件沒有使用 utf-8 編碼造成的，所以在站點配置文件 `_config.yml` 中寫中文網站名，然後把站點配置文件保存為 utf-8 格式。

```
title: 歲月留痕
subtitle: kevinluo's BLOG
```

8.2 怎麼列出 hexo 依賴插件的完整性？

```
npm ls --depth 0
```

klBlog:

```
hexo-site@0.0.0 H:\tmp_H\001.work\004.env\01prjsp\hexo\klBlog
+-- eslint@6.3.0
+-- hexo@3.9.0
+-- hexo-deployer-git@1.0.0
+-- hexo-filter-flowchart@1.0.4
+-- hexo-generator-archive@0.1.5
+-- hexo-generator-category@0.1.3
+-- hexo-generator-feed@2.0.0
+-- hexo-generator-index@0.2.1
+-- hexo-generator-tag@0.2.0
+-- hexo-renderer-ejs@0.3.1
+-- hexo-renderer-jade@0.4.1
+-- hexo-renderer-marked@2.0.0
+-- hexo-renderer-pug@0.0.5
+-- hexo-renderer-sass@0.4.0
+-- hexo-renderer-stylus@0.3.3
+-- hexo-server@0.3.3
+-- hexo-wordcount@6.0.1
```

8.3 路徑名和分類名分別設置，需要怎麼辦呢？

設置分類名可以在文章中設置 `frontmatter-categories`:

打開根目錄下的配置文件 `_config.yml`，找到如下位置做更改：

```
# Category & Tag
default_category: uncategorized
category_map:
  編程: programming
  生活: life
  其他: other
tag_map:
```

在這裏 `category_map` 是設置分類的地方，每行一個分類，冒號前面是分類名稱，後面是訪問路徑。

可以提前在這裏設置好一些分類，當編輯的文章填寫了對應的分類名時，就會自動的按照對應的路徑來訪問。

8.4 package.json 是什麼？

[package.json 是什麼？](#)

[關於項目中 package.json 的理解](#)

8.5 fontawesome 是什麼？

[github Font-Awesome lib](#)

[一套絕佳的圖標字體庫和 CSS 框架](#)

8.6 怎麼添加點擊紅心和漢字？

1. js 文件

```
/themes/next/source/js/  
hanzi.js  
clicklove.js
```

2. 頁面文件

```
KL+TEST  
{%- if theme.kl_click_hanzi %}  
    <script src="//lib.baomitu.com/jquery/3.4.0/jquery.min.js"></script>  
    <!-- 頁面點擊漢字 -->  
    <script type="text/javascript" src="/js/hanzi.js"></script>  
{%- endif %}  
{%- if theme.kl_click_love %}  
    <!-- 頁面點擊小紅心 -->  
    <script type="text/javascript" src="/js/clicklove.js"></script>  
{%- endif %}  
</body>
```

next 中應該用 root 相對路徑，當 root 在子目錄中，這樣 JS 也可以引用到，\klBlog\themes\next\layout_lay

```
{%- if theme.kl_click_hanzi %}  
    <script src="//lib.baomitu.com/jquery/3.4.0/      jquery.min.js"></script>  
    <!-- 頁面點擊漢字 -->  
    {{- next_js('hanzi.js') }}  
{%- endif %}  
{%- if theme.kl_click_love %}  
    <!-- 頁面點擊小紅心 -->  
    {{- next_js('clicklove.js') }}  
{%- endif %}
```

8.7 國內 Jquery CDN 有哪些？

1. 新浪 CDN（推薦）

一直好使

```
<script src="http://lib.sinaapp.com/js/jquery/2.0.2/jquery-2.0.2.min.js">  
</script>
```

2. Baidu CDN 有時候不好使

```
<script src="http://libs.baidu.com/jquery/1.10.2/jquery.min.js">
</script>
```

3. 微軟 CDN:

```
<script src="http://ajax.htmlnetcdn.com/ajax/jQuery/jquery-1.10.2.min.js">
</script>
```

8.8 怎麼解決 githubpages 不能識別下劃線開頭的目錄?

githubpages 不能識別下劃線開頭的目錄解決方法

使用 sphinx 創建的文檔，資源文件夾前面會帶着下劃線，本地使用沒有問題，提交到 github 上面，想使用 github pages 的時候提示 404，原因為 github pages 的 jekyll 模版會忽略下劃線開頭的文件，自動忽略下劃線開頭的目錄，從而導致引用不到 CSS,JAVASCRIPT,ETC.，所以要禁用 jekyll

禁用方法就是在文件在項目目錄下添加.nojekyll 文件

CDN

CDN 前端靜態資源庫 [baomitu-used in maupassant-hexo](#)

[75CDN 奇舞團](#)

9 hexo deploy 網站部署

參見 [Hexo 博客從搭建部署到 SEO 優化等詳細教程](#)

9.1 hexo d 法

hexo.io/docs/deployment

安裝擴展:

```
$ npm install hexo-deployer-git --save
```

需要在配置文件 _config.xml 中作如下修改:

```
deploy:
  type: git
  repo: git@github.com:jiji262/jiji262.github.io.git
  branch: master
```

然後在命令行中執行

```
hexo d
```

密碼輸入形式

```
deploy:
  type: git
  repo: https://kevinluolog:xxxxxx[密碼]@github.com/kevinluolog/ kevinluolog.github.io.git
  branch: master
```

```
# repo: git@github.com:kevinluolog/kevinluolog.github.io.git
```

```
#例如你的賬號為:crown3,密碼為 BBB;  
#那你的repo填寫為下面這樣即可  
#github: https://crown3:BBB@github.com/crown3/crown3.github.io.git  
#coding: https://crown3:BBB@git.coding.net/crown3/倉庫名.git
```

9.2 直接 git clone 法

9.3 CI 法,

[hexo_klblog](#)

這個網址裏面提到了常用的持續集成 CI 工具:

[好代碼是管出來的——使用 GitHub 實現簡單的 CI/CD](#)

[Hexo 遇上 Travis-CI: 可能是最通俗易懂的自動發布博客圖文教程](#)

[GitHub 的 CI/CD 與 Travis 配置小記](#)

9.3.1 十大 CI 工具:

Travis CI
Circle CI
Jenkins
AppVeyor
CodeShip
Drone
Semaphore CI
Buildkite
Wercker
TeamCity

9.3.2 travis CI:

Travis 可以執行多種語言的測試及構建, [官方文檔](#)

[Build Lifecycle documentation](#)

9.3.2.1 travis CI 配置步驟: 那既然需要使用 travis 自動化更新你的博客, travis 自然需要讀寫你的 github 上的 repo。github 提供了 token 機制來供外部訪問你的倉庫。

<https://github.com/settings/tokens>

1. 安裝 travis, 并授權管理 repo

marketplace 搜索 travis, 并安裝。

github/{user name}/personal settings/application/ travis CI configure 選擇要 travis 管理的 repo.

2. 配置 github token

分三步,

step 1: 在github中生成token,
 github/{user name}/-> setting->Developer settings-> Personal access tokens -> generat
 step 2: 再在travis網頁項目中配置環境變量\$GH_TOKEN, 填入token
 step 3: 在blog root _config.yml deploy 中, 設置gh_token訪問標記
 step 4: 在.travis.yml中, 在編譯完後, deploy之前用sed替換, step 3 中在 _confi.yml 中的gh_t

3. 配置 travis

在 travis 進入倉庫同步管理, [here](#)

主要是前面的 gh-token 環境變量.

4. 在源碼倉庫根目錄增加.travis.yml 修改 _config.yml

• 增加.travis.yml

注意, 如果源碼是在分支上要修改 branches 為相應的分支名, 缺省是 master:

```
### for branch of hexo-next-muse
```

```
# 指定語言環境
```

```
language: node_js
```

```
# 指定需要sudo權限
```

```
sudo: required
```

```
# 指定node_js版本
```

```
node_js:
```

```
- 10.16.3
```

```
# 指定緩存模塊, 可選。緩存可加快編譯速度。
```

```
cache:
```

```
  directories:
```

```
    - node_modules
```

```
# 指定博客源碼分支, 因人而異。hexo博客源碼托管在獨立repo則不用設置此項
```

```
branches:
```

```
  only:
```

```
    - hexo-next-muse
```

```
before_install:
```

```
  - npm install -g hexo-cli
```

```
# Start: Build Lifecycle
```

```
install:
```

```
  - npm install
```

```
  - npm install hexo-deployer-git --save
```

```
# 執行清緩存, 生成網頁操作
```

```
script:
```

```
  - hexo clean
```

```
  - hexo generate
```

```
# 設置git提交名, 郵箱; 替換真實token到_config.yml文件, 最後depoy部署
```

```
after_script:
```

```
  - git config user.name "kevinluolog"
```

```
  - git config user.email "kevinluolog_72@163.com"
```

```
# 替換同目錄下的_config.yml文件中gh_token字符串為travis後臺剛才配置的變量, 注意此處
```

```
  - sed -i "s/gh_token/${GH_TOKEN}/g" ./_config.yml
```

```
  - hexo deploy
```

```
# End: Build LifeCycle
```

- 修改 _config.yml

1. 主要是修改 deploy 部分, 決定 gh-token 和推送部署到什麼 repo 的什麼分支。如果是 xxxx.github.io 就推到 master, 如果是子目錄 repo, 則推送到 gh-pages 分支。
2. 設置 root 變量, 如果是子目錄 repo, 則需要設置相應的子目錄 repo 名字。這樣在網頁引用 css 等資源時可以直接引用到。因為 hexo 引用 CSS 等資源時用的是絕對目錄, 如 {子目錄 repo 名}/css/xxx.css, sphinx 用的是相對目錄, 如 _static/css/xxx.css。此處因為 sphinx 資源目錄前面帶了下劃線, _ , 因 hexo 和 jekyll 會自動忽略下劃線開頭的目錄, 從而導致引用不到 CSS,JAVASCRIPT,ETC., 所以在根目錄要添加.nojekyll 文件, 詳細請參考[怎麼解決 githubpages 不能識別下劃線開頭的目錄?](#)

```
root: /hexo-next-muse/
# Deployment
deploy:
  type: git
  repo: https://gh_token@github.com/kevinluolog/hexo-next-muse.git
  branch: gh-pages
```

9.3.3 travis CI 配置實例

網站規劃結構參見[my deploy: kevinluolog.github.io](#)

簡單講就是, kevinluolog/hexo-klblog-src repo 作為源碼倉庫, master 分支對應主網站 repo 的 master 分支, 其餘分支各對應主網站的子目錄網站 repo 的 gp-pages 分支。源碼 repo 分支名稱和子目錄網站的倉庫名要取得一樣, 以方便對應。每次源碼有推送時, 觸發對應分支的 travis CI 啟動, 源碼拉取->環境搭建->編譯->部署, 部署時是部署到對應的網站 repo 的 gp-pages 分支上的。

最終的效果是, 寫或修改文章時祇和 source 目錄中的 _post 目錄相關 \hexo\klBlog\source_posts 改動完成提交到對應的分支後, 過 2 分鐘左右, 對應的網站即要自動更新。非常方便和快捷, 不用占用本人的時間也不用占用本機的 CPU 去編譯和部署。同時可以的任何可以上網的地方寫文章, 提交。和寫程序一模一樣, 一個 github 搞定一切。

9.3.3.1 創建源碼新分支需要改動的文件

1. 文件.travis.yml

啟動分支名。

```
# 指定博客源碼分支, 因人而異。hexo博客源碼托管在獨立repo則不用設置此項
branches:
  only:
    - hexo-next-xxx
```

2. 文件 _config.yml

- theme
- theme: next
- root, 子網站根目錄, 要和網站 repo 名字相同
- root: /hexo-next-xxx/
- deloy, 推送目標倉庫名

```
deploy:
  type: git
  repo: https://gh_token@github.com/kevinluolog/hexo-next-xxx.git
  branch: gh-pages
```

3. 其它博客定制相关文件

如 theme 下配置文件 `_config.yml`, next 和 melody 支持独立配置文件在 `_data/next.yml` 和 `melody.yml`

i. `_data/next.yml`

- 風格

```
# Schemes
#scheme: Muse
#scheme: Mist ##kl+
#scheme: Pisces
scheme: Gemini
```

9.4 需求：在 github pages 子目錄建立 hexo 博客

在 githubpages 子目錄建立 hexo 博客

實現：

1. 首先建立 `xxx.github.io` 的 repo, `xxx` 是你的用戶名, 之後開啓 github pages 服務
2. 再建立一個 `bbbb` 的 repo, `bbbb` 是你想要的子目錄
3. 設置 hexo 的 deploy 配置文件 `_config.yml`

```
- type: git
  repo: https://github.com/xxx/bbbb.git
  branch: gh-pages
```

4. 修改 `_config.yml` 中的 `root` 選項, 由 `"/"` 改為 `"/bbbb"`

github page 就大概兩種, 一種 user page 必須 master 分支, 另一種 project page 需要給對應的 project 設置一個 `gh-pages` 分支, 上傳好網頁資源文件之後, 就可以在 `username.github.io/projectname` 這樣的域名訪問了。

網上挺多教程都不太對, 自己解決了之後記錄一下。

9.5 my deploy: kevinluolog.github.io

9.5.1 Repo of sites:

1. kevinluolog.github.io: branch:master
generated by travis Ci from repo of hexo source - branch of master
2. hexo-XXX: branch:gh-pages
can be accessed by kevinluolog.github.io/hexo-xxx
generated by travis Ci from repo of hexo source - branch of hexo-XXX
3. gp: branch:gh-pages
can be accessed by kevinluolog.github.io/gp/xxx
generated by sphinx etc. directly

9.5.2 Repo of hexo source: private

Travis can access with private repo.

- branch: master
hexo source, and deploy to kevinluolog.github.io - master automatically
- branch: hexo-xxx
deploy to REPO: hexo-xxx - gh-pages, automatically.
branch `_config.yml` for set different theme
travis-ci triggered by source file merged in. then compile and deploy.
books, created by sphinx or docutils or pandoc.