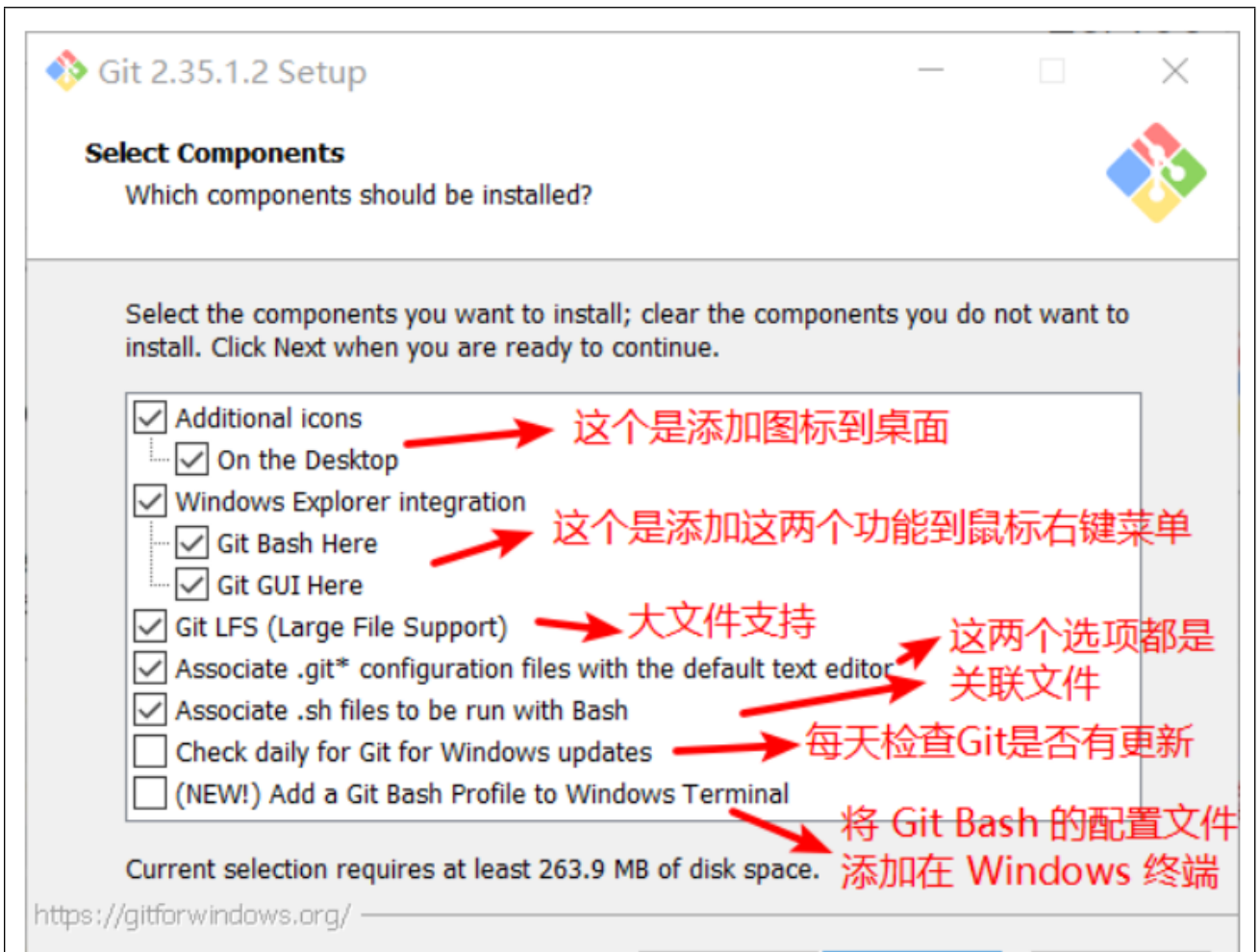


山东大学 计算机科学与技术 学院

云计算技术 课程实验报告

学号：202100202019	姓名：崔奕斐	班级：计科 21.6
实验题目：利用主流云平台搭建个人博客或网站		
实验学时：2	实验日期：2024. 3. 22	
<p>实验目的：熟悉使用主流云平台并搭建个人博客或者网站。</p> <p>具体包括：</p> <p>参考方案：基于主流云平台，设计、实现个人博客或者网站的搭建，撰写实验报告（附带网站链接并可以访问），并在网站上呈现此次实验报告。</p>		
<p>硬件环境：</p> <p>联网的计算机一台</p>		
<p>软件环境：</p> <p>Windows or Linux</p>		
<p>实验步骤与内容：</p> <p>一、环境准备</p> <p>1. 安装 Git</p> <p>Git 是一个开源的分布式版本控制系统，可以有效、高速的处理从很小到非常大的项目版本管理。Git 是 Linux Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件。优点是适合分布式开发，强调个体、公共服务器压力和数据量都不会太大、速度快、灵活、任意两个开发者之间可以很容易的解决冲突、离线工作，缺点是代码保密性差，一旦开发者把整个库克隆下来就可以完全公开所有代码和版本信息、权限控制不友好、如果需要对开发者限制各种权限的建议使用 SVN。</p> <p>git 是软件, 它可在本地建立仓库, 你写的代码的各个版本都可以存着 github 是网上仓库, 你写的代码的各个版本都可以存着。由于我们需要将本地的网页文件上传到 github 上面, github 只支持 git 作为唯一的版本库格式进行托管，所以我们需要对于 git 进行相应的安装。</p> <p>按照网上的教程进行配置，选择安装目录、选择安装组件</p>		



进行一系列配置后，得到最终的结果。在终端窗口打开后，可以看到相应的 git 版本，说明安装成功，如下图所示：

```
C:\Windows\system32\cmd.exe

Microsoft Windows [版本 10.0.19045.3448]
(c) Microsoft Corporation。保留所有权利。

C:\Users\13970>git --version
git version 2.38.1.windows.1
```

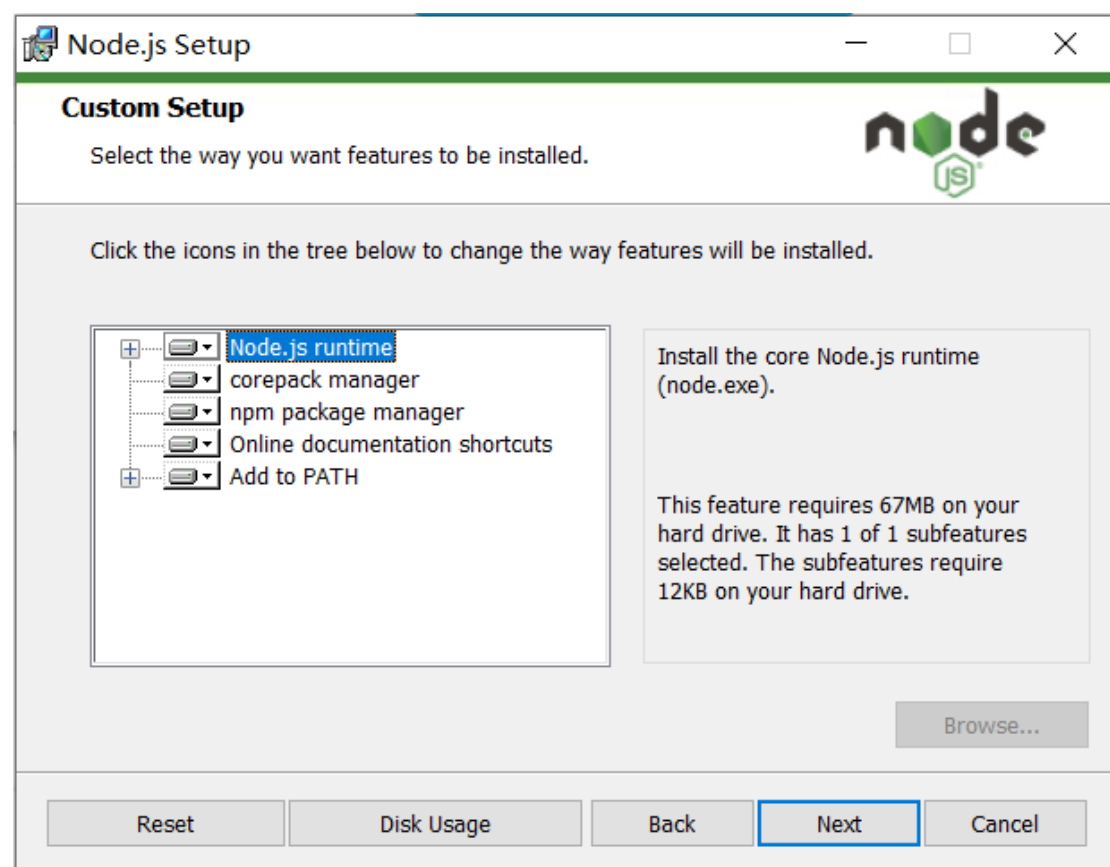
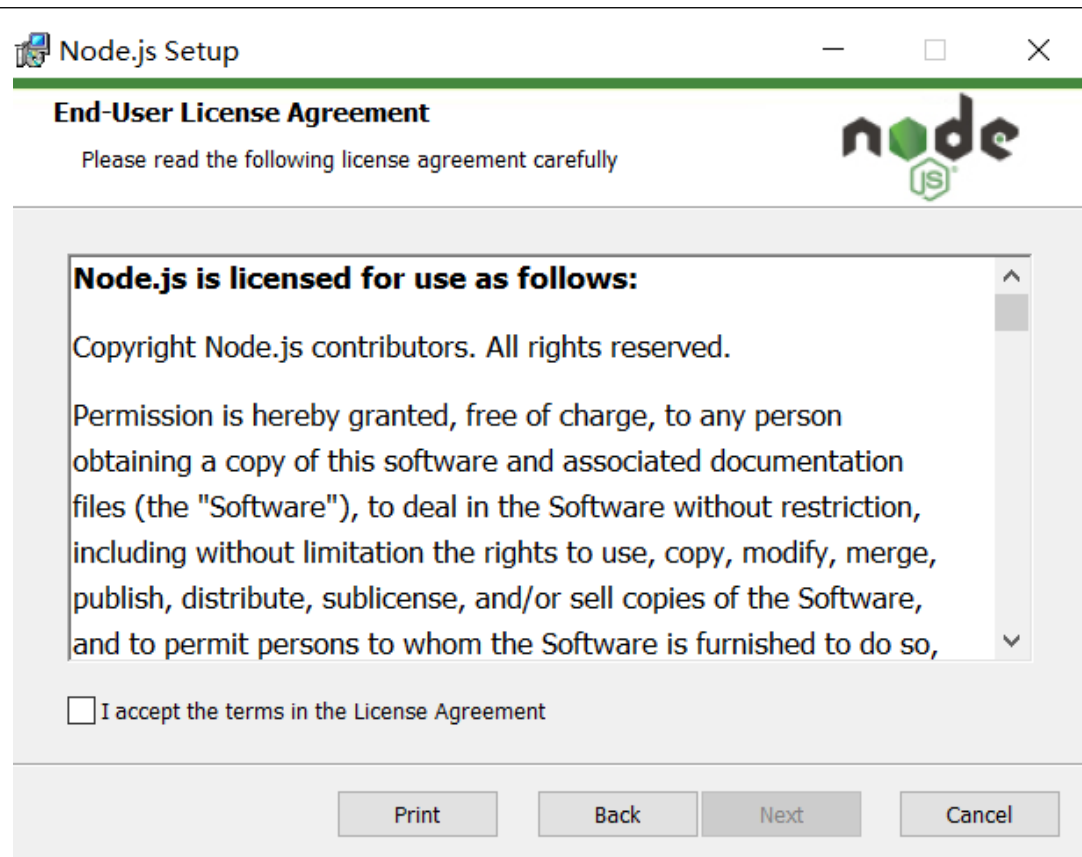


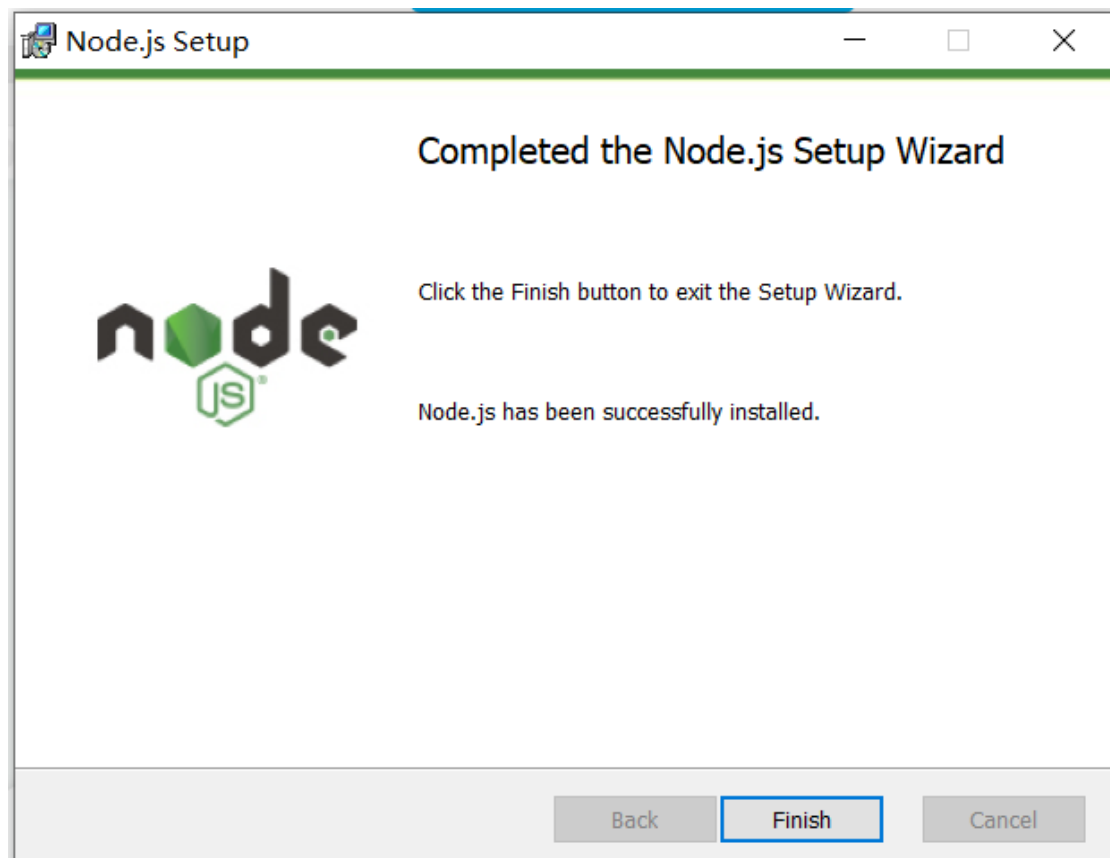
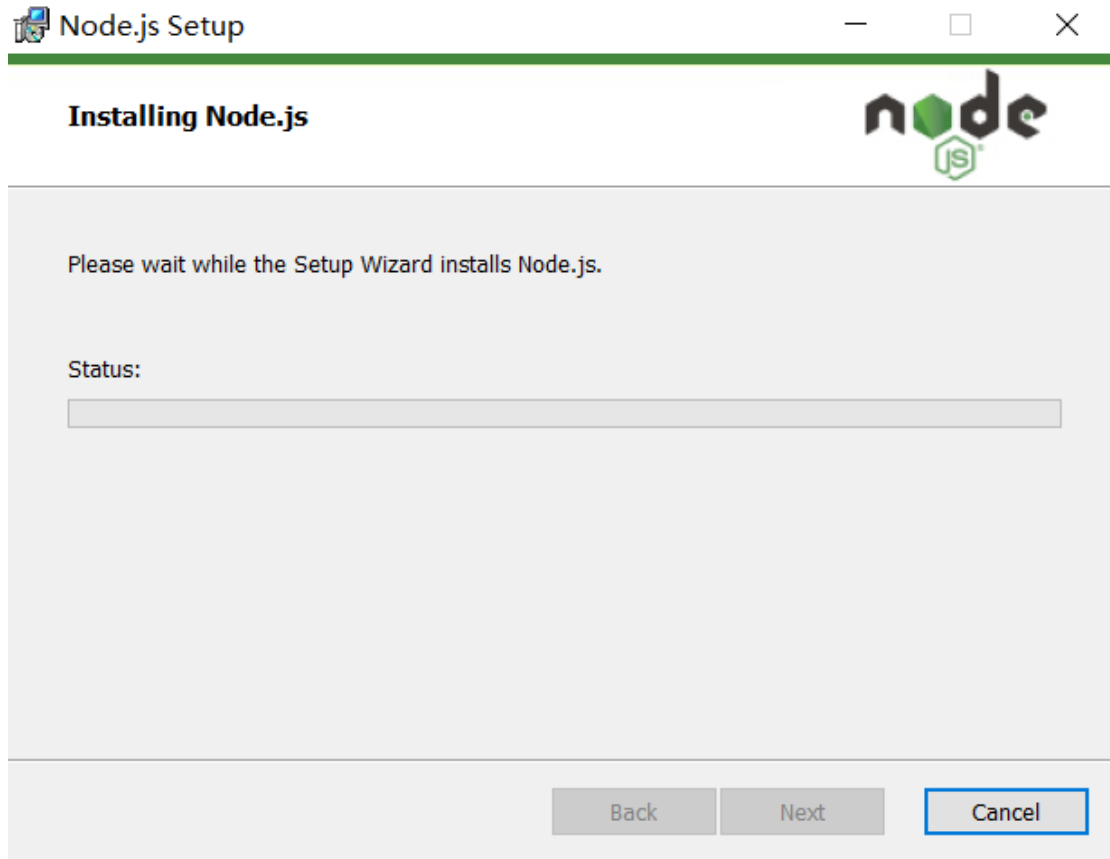
在需要进行版本控制的文件夹下，单击右键，可以得到上图所示结果，表明安装完成。

2. 安装 node.js

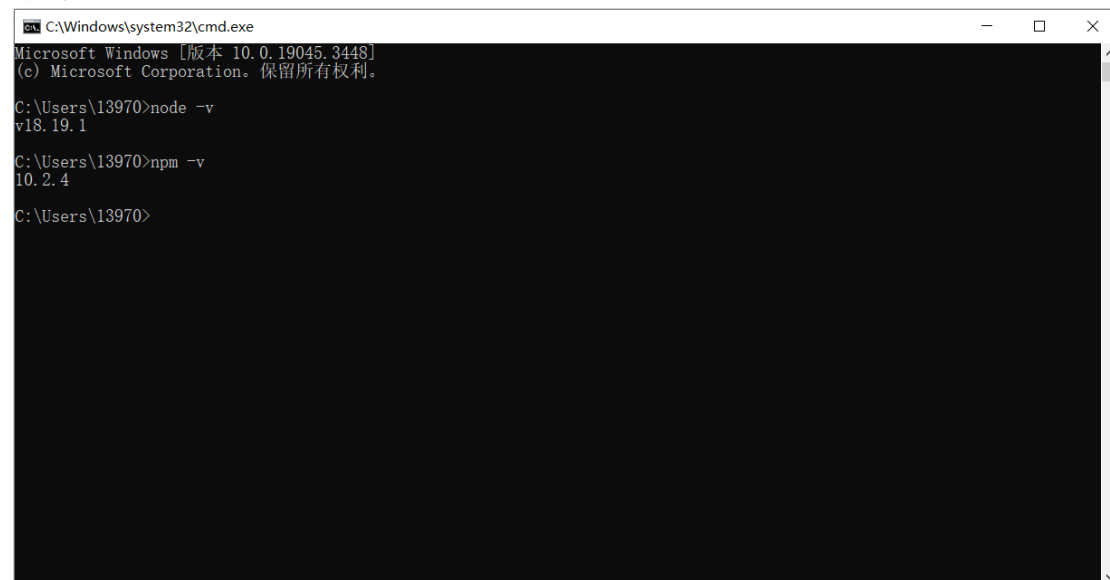
Node.js 是一种开源与跨平台的 JavaScript 的运行环境，能够使得 javascript 脱离浏览器运行。它是一个可用于几乎任何项目的流行工具，允许我们通过 JavaScript 和一系列模块来编写服务器端应用和网络相关的应用。Node.js 主要用于编写像 Web 服务器一样的网络应用，这和 PHP 和 Python 是类似的。但是 Node.js 与其他语言最大的不同之处在于，PHP 等语言是阻塞的，而 Node.js 是非阻塞的。

Node.js 是基于 V8 JavaScript 引擎 构建的 JavaScript 运行时，可以用它搭建本地服务器，这里我们需要用到它的 npm 命令。按照教程进行默认配置即可，部分实验截图如下所示：





检测是否安装成功：命令行运行 `node -v` 和 `npm -v`（注意中间要有空格），出现对应版本号即代表安装成功。



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.19045.3448]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\13970>node -v
v18.19.1

C:\Users\13970>npm -v
10.2.4

C:\Users\13970>
```

3. Hexo 的安装与配置：

Hexo 是一个快速、简洁且高效的博客框架。Hexo 使用 Markdown（或其他渲染引擎）解析文章，在几秒内，即可利用靓丽的主题生成静态网页。即把用户的 markdown 文件，按照指定的主题解析成静态网页。

安装过程如下所示：

(1) 创建一个文件夹 MyBlog，以后在这个文件夹操作，存放项目文件以及博客文章。

AndroidStudioproje	2024-03-20 23:03	文件夹
AndroidStudioTool	2024-03-08 9:04	文件夹
AS_Install	2024-03-08 9:05	文件夹
cmputer_design_test	2024-03-06 17:27	文件夹
guigumysql	2024-02-08 11:10	文件夹
MATLAB课件	2024-03-06 17:26	文件夹
MyBlog	2024-03-22 23:45	文件夹

(2) 运行命令：`npm install -g hexo-cli`：安装 hexo

```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.19045.3448]
(c) Microsoft Corporation。保留所有权利。

:\Users\13970>npm install -g hexo-cli
npm ERR! code CERT_HAS_EXPIRED
npm ERR! errno CERT_HAS_EXPIRED
npm ERR! request to https://registry.npm.taobao.org/hexo-cli failed, reason: certificate has expired

npm ERR! A complete log of this run can be found in: C:\Users\13970\AppData\Local\npm-cache\_logs\2024-03-22T11_46_05_15Z-debug-0.log

:\Users\13970>npm cache clean --force
npm WARN using --force Recommended protections disabled.

:\Users\13970>npm config set strict-ssl false

:\Users\13970>npm install -g hexo-cli

added 54 packages in 9s

:\Users\13970>
```

(3) 运行命令：hexo v：查看是否安装成功，出现对应版本号即代表安装成功

```
C:\Windows\system32\cmd.exe

C:\Users\13970>npm cache clean --force
npm WARN using --force Recommended protections disabled.

C:\Users\13970>npm config set strict-ssl false

C:\Users\13970>npm install -g hexo-cli

added 54 packages in 9s

C:\Users\13970>hexo v
hexo-cli: 4.3.1
os: win32 10.0.19045
node: 18.19.1
acorn: 8.10.0
ada: 2.7.2
ares: 1.20.1
base64: 0.5.0
brotli: 1.0.9
cjs_modulelexer: 1.2.2
cldr: 43.1
icu: 73.2
llhttp: 6.1.0
modules: 108
napi: 9
nghttp2: 1.57.0
nghttp3: 0.7.0
ngtcp2: 0.8.1
openssl: 3.0.13+quic
simdutf: 3.2.18
```

(4) 运行命令：hexo init：在当前目录（博客文件夹目录，而且必须为空目录）初始化一个使用 hexo 框架的项目（默认）。

```
MINGW64:/e/MyBlog

13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ hexo init
INFO Cloning hexo-starter https://github.com/hexojs/hexo-starter.git
INFO Install dependencies
INFO Start blogging with Hexo!
```

(5) 运行命令：hexo generate：生成项目

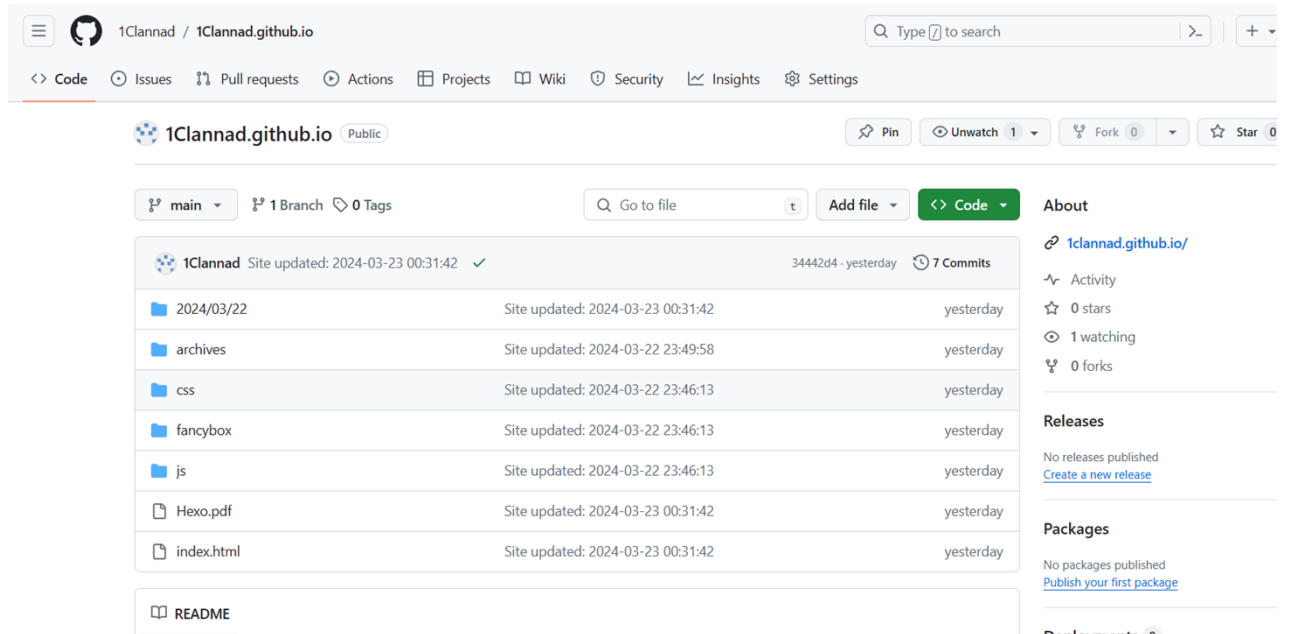
```
13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ hexo generate
INFO Validating config
INFO Start processing
INFO Files loaded in 388 ms
INFO Generated: archives/index.html
INFO Generated: index.html
INFO Generated: archives/2024/03/index.html
INFO Generated: fancybox/jquery.fancybox.min.css
INFO Generated: js/script.js
INFO Generated: css/style.css
INFO Generated: js/jquery-3.6.4.min.js
INFO Generated: fancybox/jquery.fancybox.min.js
INFO Generated: archives/2024/index.html
INFO Generated: css/images/banner.jpg
INFO Generated: 2024/03/22/hello-world/index.html
INFO 11 files generated in 297 ms
```

(6) 运行命令：hexo s/hexo server：启动本地服务器，按 Ctrl 键点击链接即可用默认浏览器访问，用按 Ctrl+C 停止运行。

二、使用 GitHub 提供的免费服务器

1. GitHub 简介：一个项目开放源代码的国外的代码托管平台，我们将用这里的 GitHub 仓库的 Pages 功能使用 GitHub 为我们提供的个人服务器。

2. 注册 github 账号 (<https://github.com/>)，并创建一个仓库（默认选项创建就行）--> 仓库名格式要求：你的 GitHub 用户名.github.io。



The screenshot shows the GitHub interface for the repository `1Clannad / 1Clannad.github.io`. The repository is public and has 0 stars, 1 watching, and 0 forks. The file tree shows the following structure:

File/Folder	Site updated	Commit
2024/03/22	2024-03-23 00:31:42	yesterday
archives	2024-03-22 23:49:58	yesterday
css	2024-03-22 23:46:13	yesterday
fancybox	2024-03-22 23:46:13	yesterday
js	2024-03-22 23:46:13	yesterday
Hexo.pdf	2024-03-23 00:31:42	yesterday
index.html	2024-03-23 00:31:42	yesterday

The right sidebar shows the repository's activity, including a link to `1clannad.github.io/`, 0 stars, 1 watching, 0 forks, and no releases or packages published.

3. 初始化 Git 的用户名和邮箱（加--global 是为所有 git 项目设置，不加则是在当前目录

的 git 项目设置)

(1) 初始化用户名-- 运行命令: `git config --global user.name "yourname"` (注意 user.name/email 和英文双引号"之间要有一个空格)

(2) 初始化邮箱--运行命令: `git config --global user.email "youremail"`

(3) 确认设置用户名和邮箱成功: 执行以下命令

用户名: `git config --global user.name`

邮箱: `git config --global user.email`

```
13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ git config --global user.name "lClannad"

13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ git config --global user.email "1397076387@qq.com"

13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ git config --global user.name
lClannad

13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ git config --global user.email
1397076387@qq.com

13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ npm install hexo-deployer-git --save

added 10 packages in 2m
```

4. 进行 ssh 连接, 只有连接成功才能上传。

(1) 下载 github 部署插件: 运行命令: `npm install hexo-deployer-git --save`
本地生成获取 ssh 密钥。

```
13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ git config --global user.name "lClannad"

13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ git config --global user.email "1397076387@qq.com"

13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ git config --global user.name
lClannad

13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ git config --global user.email
1397076387@qq.com

13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ npm install hexo-deployer-git --save

added 10 packages in 2m
```

(2) 生成密钥：运行命令：ssh-keygen -t rsa -C "个人的邮箱"。

```
13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ ssh-keygen -t rsa -C "1397076387@qq.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/13970/.ssh/id_rsa):
Created directory '/c/Users/13970/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/13970/.ssh/id_rsa
Your public key has been saved in /c/Users/13970/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:D12UL1PXFyOIYuh6Cidy+PVOb1h62Sm6Pep2XtKQcvw 1397076387@qq.com
The key's randomart image is:
+---[RSA 3072]-----+
|      .  o..o o. |
|      . o oo.. o o |
|      . . .O..  . |
|      o oo..    |
|      . o * So   |
| oo.o.+.*       |
| .o+.o=ooE.     |
| ..++0oo        |
|      o=X=o      |
+---[SHA256]-----+
```

(3) 获取密钥：在 Git Bash 运行命令 cat ~/.ssh/id_rsa.pub，然后复制这行命令运行的结果。

```
13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDSLZSuxDvq1gwj5T6vvPePgNzHfs/c+J7zN3FZOD3t
KsY3U3TkqxUmdUlejGgn9+8EWSvIHp/ZoSEz9NjBEpStiuHT1mpyZDrWjoQ8rrEQDOUs1gEHLrQLC/
WtgvyY45o7BL6kSD+CXmtI2obzx1Xh/PVDHujYZhu8MqSJoZhIAD/CVhGnRDshZgP8tpHjtxB2+18ng5E
IJhwonS60/TdFz149Hhga/HPgcLkjZrW/1t1FC+ZNwhbVR9gQpM0fX65kDyTZBdXXYo4Hox5bZMyASZK
UC6fAMSDHb4aE6mtsVN+ZD/018DAYqnZkeFHBBpVvnZUGDH9S1INMfkKAVK6ZF9YnLaur1tw+vZiWisY
/sCL46nhYfZjsCobqRg3o+B6nd+dcjWajrWpZD0Hu7h6yA5s3Ui5U1Y31+DEyABB+1h14TAp8+i8o21E
51f2Md0sWzsnUBDBNMI67PSCZ2Fat8r2QaS93pw55T1rPC9N+MdEpKX1CMiXPrPR0uHE1k= 1397076
387@qq.com

13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ ssh -T git@github.com
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Hi 1Clannad! You've successfully authenticated, but GitHub does not provide shell
access.
```

github 创建 ssh 密钥

(4) 打开 github，在右上角的头像下面点击 settings，再点击 SSH and GPG keys，新建一个 SSH，名字随便取一个都可以，把你的 id_rsa.pub 里面的信息复制进去，点击 create/创建即可完成创建。



SSH keys

New SSH key

0 K/s
0.3 K/s

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

	ssh key1 SHA256: D12UL1PXFyOIYuh6C1dy+PV0b1h62Sm6Pep2XtKQcvm Added on Mar 22, 2024 Never used — Read/write	Delete
	ssh key2 SHA256: bbrdh6eOu46tkparpnwnzHwDOx+4CmG+Mgxsw0TG1sQ Added on Mar 22, 2024 Last used within the last week — Read/write	Delete

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

(5) 验证 ssh 连接是否成功: `ssh -T git@github.com`, 出现用户名, 即代表访问成功

```
13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDLSuxDvq1gWj5T6vvPePgNzHfs/c+J7zN3FZOD3t
KsY3U3TkqxUmdUlejGgn9+8EWSvIHp/ZoSEz9NjBEDPsTiuHTlmpyZDrwjoQ8rrEQDOUslgEhrLrQLC/
WtgvY45o7BL6kSD+CXMtI20bzx1Xh/PVDHujYZhu8MqSJoZhIAD/CVhGnRDshZgP8tpHjtxB2+18ng5E
IJhwonS60/TdFz149Hhga/HPqcLkjZrw/1t1FC+ZNwhbVR9gQpM0fX65kDyTZBdXXYo4Hox5bZMyASZK
UC6fAMSDHb4aE6mtsVN+ZD/018DAYqnZkeFHBbpVvnZUGDH9S1INMfkKAVK6ZF9YnLaur1tw+vZiWisY
/sCL46nhYfZjsCobqRg3o+B6nd+dcjWajrWpZDOHu7h6yA5s3Ui5U1Y31+DEyABB+1h14Tap8+i8o21E
51f2MdOsWzsnUBBDBNMI67PSCZ2Fat8r2QaS93pw55TlPC9N+MdEpKX1CMiXPrPR0uHE1k= 1397076
387@qq.com

13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ ssh -T git@github.com
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCoqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Hi 1Clannad! You've successfully authenticated, but GitHub does not provide shell
access.
```

5. 修改 config.yml 文件, 将部署的部分进行修改, deploy 下的 repository 要配置成自己的仓库地址, type 修改为 git, branch 修改为 main。

6. 使用 hexo 的命令修改网页。

- (1) 使用 hexo clean 命令清除之前项目的一些问题。
- (2) 使用 hexo g 命令在本地生成项目。
- (3) 使用 hexo d 命令将项目部署到 github 上。

```
13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ hexo d
INFO Validating config
INFO Deploying: git
You have to configure the deployment settings in _config.yml first!
```

```
Example:
  deploy:
    type: git
    repo: <repository url>
    branch: [branch]
    message: [message]

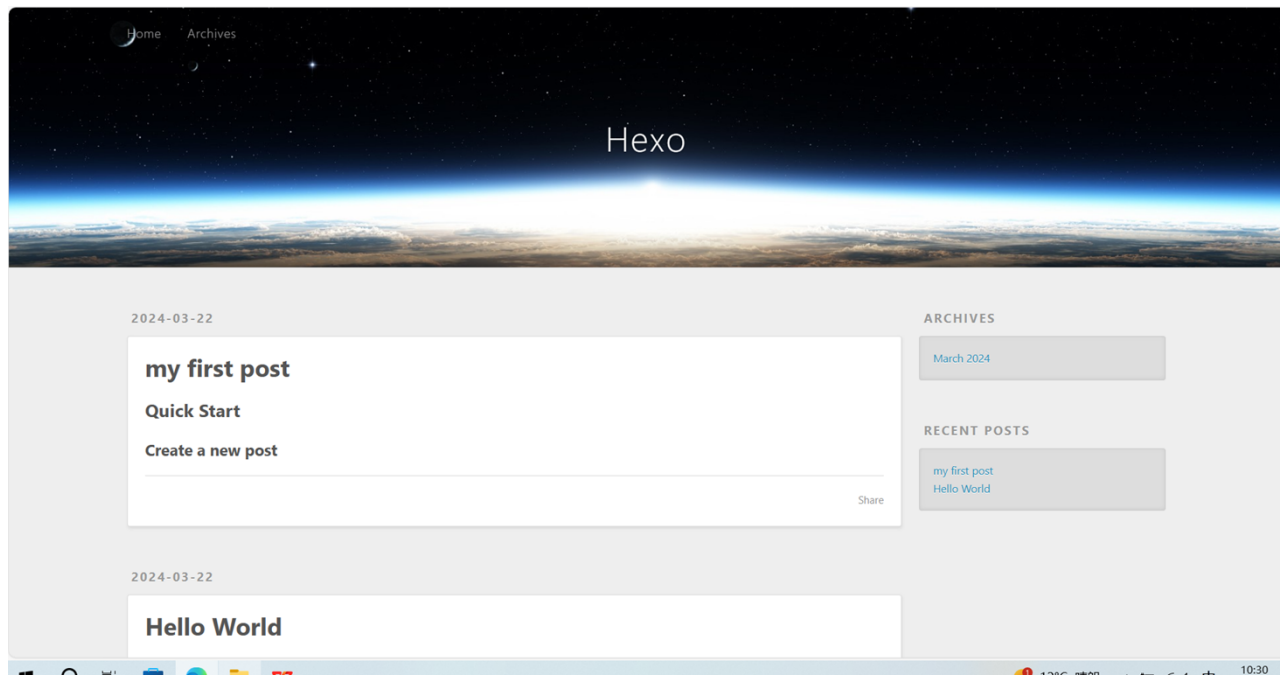
    extend_dirs: [extend directory]
```

```
For more help, you can check the docs: https://hexo.io/docs/deployment.html
INFO Deploy done: git
```

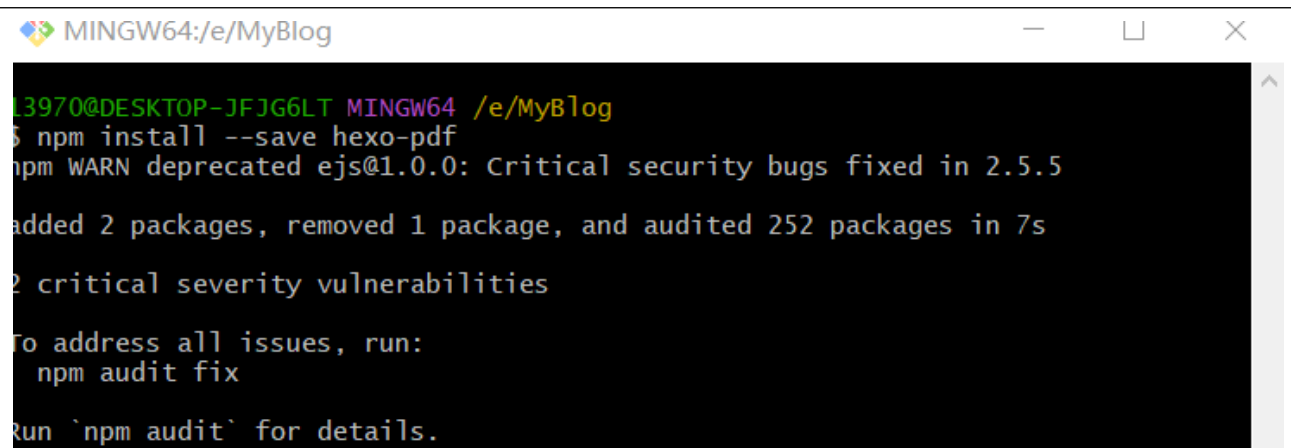
```
it
warning: in the working copy of 'js/search/meilisearch.js', LF will be replaced by CRLF the
touches it
[master 23765ab] Site updated: 2024-03-22 23:42:02
23 files changed, 26914 insertions(+), 1811 deletions(-)
create mode 100644 css/Readme.html
create mode 100644 css/first.css
delete mode 100644 css/images/banner.jpg
delete mode 100644 fancybox/jquery.fancybox.min.css
delete mode 100644 fancybox/jquery.fancybox.min.js
create mode 100644 js/app.js
delete mode 100644 js/jquery-3.6.4.min.js
create mode 100644 js/plugins/aplayer.js
create mode 100644 js/plugins/parallax.js
create mode 100644 js/plugins/rightMenus.js
create mode 100644 js/plugins/tags/contributors.js
create mode 100644 js/plugins/tags/friends.js
create mode 100644 js/plugins/tags/sites.js
delete mode 100644 js/script.js
create mode 100644 js/search/algolia.js
create mode 100644 js/search/hexo.js
create mode 100644 js/search/meilisearch.js
Enumerating objects: 48, done.
Counting objects: 100% (48/48), done.
Compressing objects: 100% (27/27) done
```

```
warning: in the working copy of '2024/03/22/hello-world/index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '2024/03/22/my-first-post/index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'archives/2024/03/index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'archives/2024/index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'archives/index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'css/style.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'fancybox/jquery.fancybox.min.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'js/jquery-3.6.4.min.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'js/script.js', LF will be replaced by CRLF the next time Git touches it
[master 34442d4] Site updated: 2024-03-23 00:31:42
3 files changed, 5 insertions(+), 5 deletions(-)
create mode 100644 Hexo.pdf
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 62.53 KiB | 233.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:1Clannad/1Clannad.github.io.git
30092a2..34442d4 HEAD -> main
branch 'master' set up to track 'git@github.com:1Clannad/1Clannad.github.io.git/main',
```

最终效果如下所示：

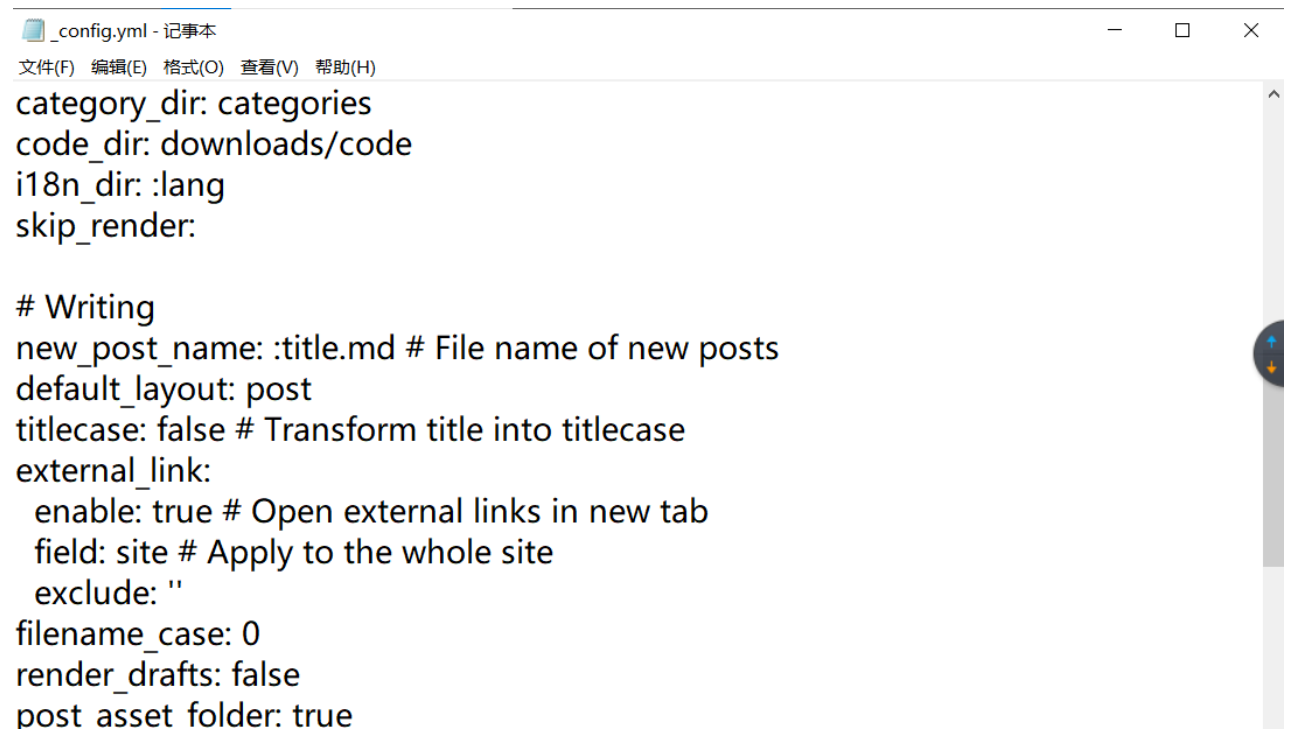


7. 将实验报告上传到个人博客上。
 - (1) 安装 hexo-pdf 文件。



```
MINGW64:/e/MyBlog
13970@DESKTOP-JFJG6LT MINGW64 /e/MyBlog
$ npm install --save hexo-pdf
npm WARN deprecated ejs@1.0.0: Critical security bugs fixed in 2.5.5
added 2 packages, removed 1 package, and audited 252 packages in 7s
2 critical severity vulnerabilities
To address all issues, run:
  npm audit fix
Run `npm audit` for details.
```

2、修改项目的 `_config.yml` 配置



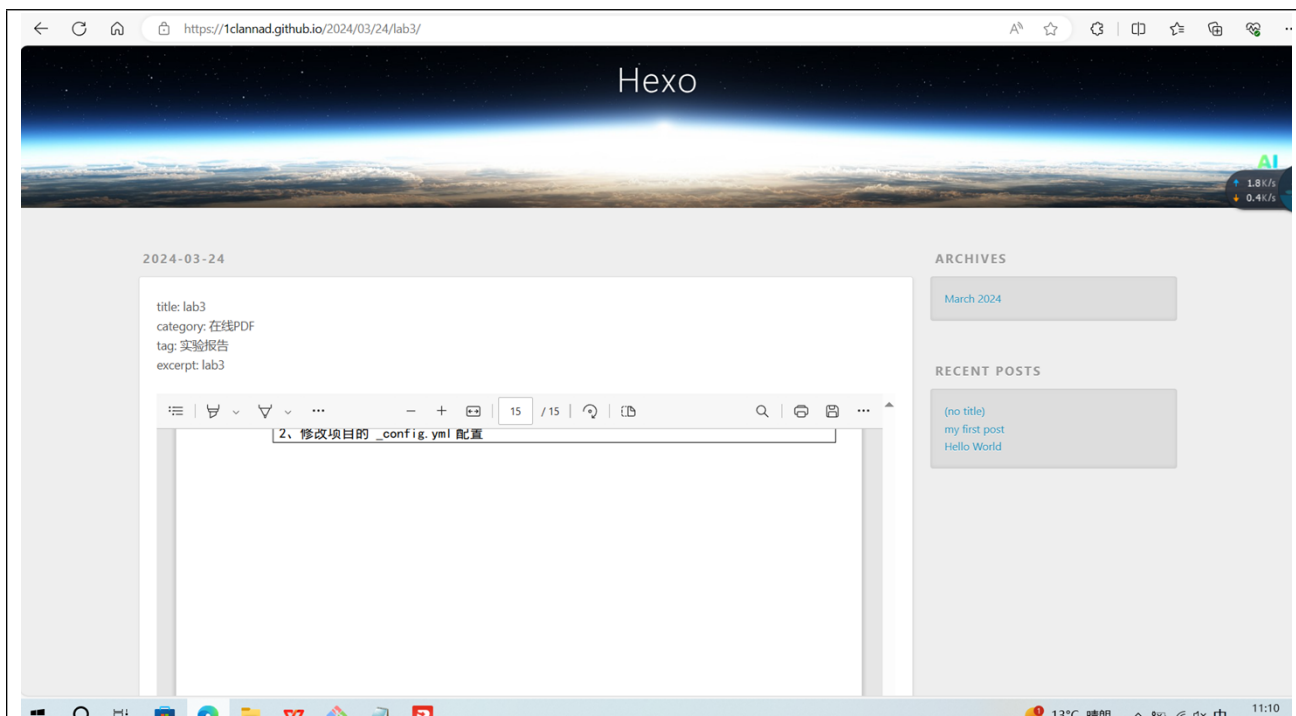
```
_config.yml - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
category_dir: categories
code_dir: downloads/code
i18n_dir: :lang
skip_render:

# Writing
new_post_name: :title.md # File name of new posts
default_layout: post
titlecase: false # Transform title into titlecase
external_link:
  enable: true # Open external links in new tab
  field: site # Apply to the whole site
  exclude: ""
filename_case: 0
render_drafts: false
post_asset_folder: true
```

3、在 `source_posts` 下创建目录，放入 pdf 文件

4. 创建与上面目录相同名的 md 文件

最终重新上传、部署即可，最终的效果图如图所示：



链接如下所示：

<https://1clannad.github.io/2024/03/24/lab3final/>

结论分析与体会：

①通过本次实验，我对于个人博客环境的准备有了新的认识。git 是软件,它可在本地建立仓库,你写的代码的各个版本都可以存着 github 是网上仓库,你写的代码的各个版本都可以存着。由于我们需要将本地的网页文件上传到 github 上面, github 只支持 git 作为唯一的版本库格式进行托管, 所以我们需要对于 git 进行相应的安装; Node.js 是一种开源与跨平台的 JavaScript 的运行环境, 能够使得 javascript 脱离浏览器运行。 它是一个可用于几乎任何项目的流行工具, 允许我们通过 JavaScript 和一系列模块来编写服务器端应用和网络相关的应用; Hexo 是一个快速、简洁且高效的博客框架。Hexo 使用 Markdown (或其他渲染引擎) 解析文章, 在几秒内, 即可利用靓丽的主题生成静态网页。即把用户的 markdown 文件, 按照指定的主题解析成静态网页。

②通过本次实验, 对于 hexo 框架我有了一定的了解。Hexo 初始化后 node_modules 存放 node 依赖包的文件夹, 在 js 文件中可以直接导入使用其中的依赖包, 运行 npm i 指令时会把 package.json 中 dependencies 的内容下载到该文件夹下。_config.yml 配置文件, 可在该文件中配置大部分的参数。包括标题, 作者, 叙述, 主题等等。source 存放用户资源的地方, _posts 中就存放着我们写的文章。

③在本次实验中, 我也遇到了一些问题。hexo deploy 时出现了警告: LF will be replaced by CRLF, 在查阅相关资料后, 我意识到原因是 windows 下换行符为 CRLF, Linux 下换行符为 LF (使用 Git 命令行 Git Bash, 实际上就是相当于在 linux 环境), 而 git 工作区默认换行符为 CRLF, 当执行 git add ... 时就会出现警告! 当最终 push 到远程仓库时 git 会统一格式全部转化为用 CRLF 作为换行符, 故不必对其做转换操作, 文本文件保持原样, 只需执行以下命令 git config --global core.autocrlf false,禁用自动转换即可。

此外, 在本次博客搭建的过程中, 也是遇到了许多坎坷。起初我使用的是阿里云的轻量应

用服务器，但是需要花费一个月 60 的费用进行购买，看到微信群老师不限制平台后，我决定使用 github 的免费服务器。但是在搭建的过程中，我的本地服务器始终拒绝连接，切换端口也会有相应的报错，最终查阅大量资料后，我才意识到是浏览器代理的问题。最后，还有一个问题是始终无法得到相应的页面部署，在查阅相关资料后，我意识到重新建一个仓库即可，最终得到正确的页面部署。