

WEB概念与环境配置

《WEB实训》

作者：刘文灿

目录

- ◆ WEB前端发展概况
- ◆ 前端工程师的职业方向
- ◆ 开发环境的搭建
- ◆ 第一个网页
- ◆ 基本命令行
- ◆ git基础用法
- ◆ 部署上线

WEB前端发展概况

前端领域的重要技术变革

- ◆ v0.1: 基于模板渲染的动态页面
- ◆ v1.0: 基于 AJAX 的前后端分离
- ◆ v2.0: 组件化和工程化
- ◆ v3.0: Serverless全栈化和全端化

WEB前端发展概况~v1.0

- ◆ WEB前端=
HTML/CSS/JS+jQuery+Bootstrap+Ajax?
- ◆ 传统WEB开发主要特征：
 - 1.以HTML文件为中心，引入JS,CSS;
 - 2.前后端混合;
 - 3.库/框架以jQuery为主，其他为辅;
 - 4.以DOM为中心，没有组件化开发的概念;

WEB前端发展概况~v2.0

- ◆ 现代WEB前端特征:

- 1.以JS为中心;
- 2.组件化开发;
- 3.数据流管理;
- 4.UI设计规范;
- 5.统一的模块管理;
- 6.工具链: 包管理、构建、部署
- 7.使用改进的语言, 如ES6, Sass等;

WEB前端发展概况~v3.0

以全栈为目标的前端有一个梦想：走后端的路，让后端无路可走。这当然是过于理想，不过借助Serverless，前端确实可以解决中小规模的后端开发；而后端则更靠后，以提供基础设施和解决复杂的后端问题为主。

不论是全栈还是全端，都是目前已经在实践的技术，但是它们还没有形成一致的标准。

- ◆ 全栈化：

1. 以Serverless为基础设施
2. 使用node实现中小规模的后端

- ◆ 全端化：

1. 使用同样的框架输出原生的小程序、APP和桌面应用

WEB前端发展概况

◆ 如何跟得上时代?

当你到网络或者书店查找资料，从古老的jQuery方案，到最新的组件化方案都呈现在你面前，很难判断到底自己在看的资料是否已经过时了？

比如，JS包管理工具，有bower, npm, yarn之后，选哪个？

比如，JS构建工具，grunt, browserify, gulp, webpack，选哪个？

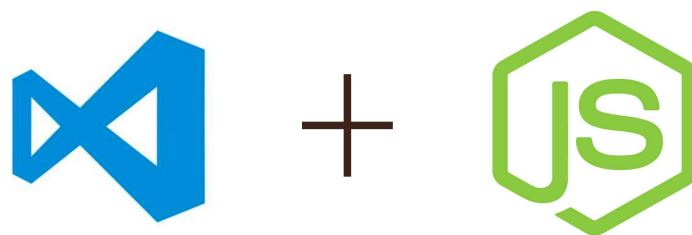
比如，组件化框架，angular, react, vue，选哪个？

这既需要实际的开发经验，也需要一定的理论知识。本课程在这个问题上，一个是采用目前主流的新方案做项目，一个是教授各位掌握一定的理论知识，帮助学生少走弯路，更快跟上时代。

前端工程师的职业方向

- ◆ 以视觉和体验为主的前端工程师（企业站、商城等）
- ◆ 以逻辑为主的前端工程师（管理后台、APP）

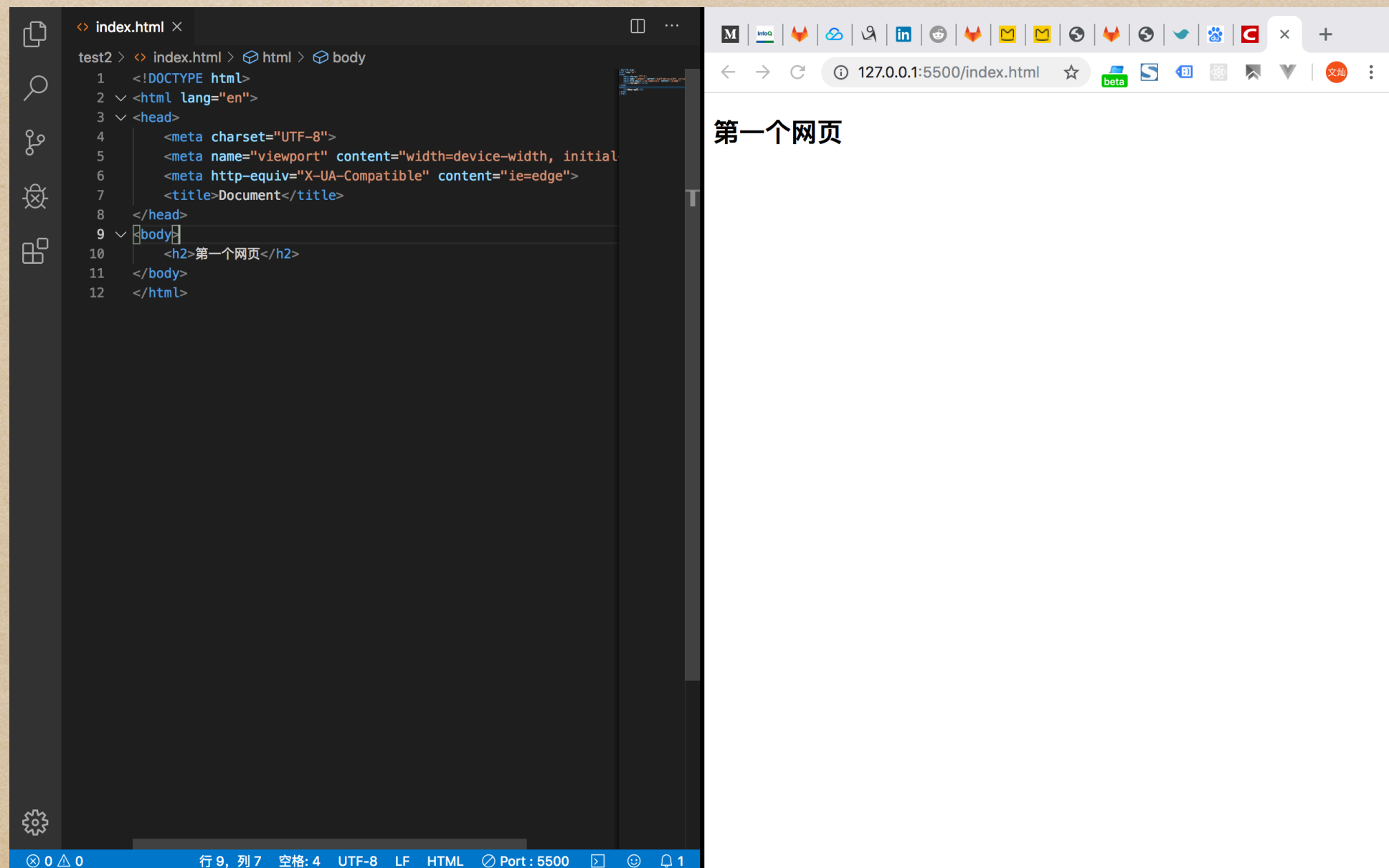
开发环境构建



- ◆ 编辑器:统一使用VSCode (其他如Sublime, Atom, Vim, 各位在掌握后可自行了解)
- ◆ VSCode插件:中文插件、Live Server
- ◆ Node 10.15.0
- ◆ 版本控制: git
- ◆ 浏览器: Chrome

开发环境构建

VSCode使用Live Server实时预览



基本命令行

- ◆ 命令行是落后还是效率?
- ◆ DOS与Linux终端一样吗?
- ◆ 前端也要学习Linux终端吗? (1.实际生产环境只能靠命令操作
2.日常工作许多操作需要命令完成)
- ◆ 常用文件与目录操作命令: `ls`, `cd`, `pwd`, `mkdir`, `rmdir`, `cp`, `rm`, `mv`
- ◆ 常用文本编辑命令: `echo`, `cat`, `touch`
- ◆ SSH相关: `ssh-keygen`, `ssh`

git基础用法

- ◆ 版本控制的简单历史: vss → cvs → svn → git
- ◆ 所有人都要做版本控制, 就算不与别人协作
- ◆ git基础用法: 创建仓库、跟踪与提交、查看状态和修改、查看提交历史、撤销跟踪和修改、分支创建与切换、远程仓库获取与同步、其他(.gitignore, .git)

git基础用法

◆ 创建仓库、跟踪与提交

```
liuwencan:src liuwencan$ mkdir test3
liuwencan:src liuwencan$ cd test3/
liuwencan:test3 liuwencan$ echo 'Hello World!' > readme.md
liuwencan:test3 liuwencan$ git status
fatal: Not a git repository (or any of the parent directories): .git
liuwencan:test3 liuwencan$ git init
Initialized empty Git repository in /Users/liuwencan/src/test3/.git/
liuwencan:test3 liuwencan$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    readme.md

nothing added to commit but untracked files present (use "git add" to track)
liuwencan:test3 liuwencan$ git add readme.md
liuwencan:test3 liuwencan$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   readme.md

liuwencan:test3 liuwencan$ git commit -m "第一次提交"
[master (root-commit) e6d6ba7] 第一次提交
 1 file changed, 1 insertion(+)
 create mode 100644 readme.md
liuwencan:test3 liuwencan$ git status
On branch master
nothing to commit, working tree clean
liuwencan:test3 liuwencan$ echo 'Hello, Solaris!' >> readme.md
liuwencan:test3 liuwencan$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   readme.md
```

◆ 查看状态、修改、提交历史

```
liuwencan:test3 liuwencan$ git diff
diff --git a/readme.md b/readme.md
index 980a0d5..6ced565 100644
--- a/readme.md
+++ b/readme.md
@@ -1,2 @@
 Hello World!
+Hello, Solaris!
liuwencan:test3 liuwencan$ git add -u
liuwencan:test3 liuwencan$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   readme.md

liuwencan:test3 liuwencan$ git commit -m "第二次提交"
[master 80e552e] 第二次提交
 1 file changed, 1 insertion(+)
liuwencan:test3 liuwencan$ git status
On branch master
nothing to commit, working tree clean
liuwencan:test3 liuwencan$ git log
commit 80e552e8ca5cd22f93a3e2364c9e5f9665c4a89b (HEAD -> master)
Author: LIUWENCAN <pheye@qq.com>
Date:   Sun Oct 20 10:47:33 2019 +0800

    第二次提交

commit e6d6ba7245c57f189ccede2e6cf23b2ea6a85008
Author: LIUWENCAN <pheye@qq.com>
Date:   Sun Oct 20 10:46:09 2019 +0800

    第一次提交
liuwencan:test3 liuwencan$ cat readme.md
Hello World!
Hello, Solaris!
```


git基础用法

◆ 撤销跟踪和修改

```
liuwencan:test3 liuwencan$ echo 'Hello, Galaxy!' >> readme.md
liuwencan:test3 liuwencan$ git diff
diff --git a/readme.md b/readme.md
index 6ced565..9b38cf7 100644
--- a/readme.md
+++ b/readme.md
@@ -1,2 +1,3 @@
 Hello World!
 Hello, Solaris!
+Hello, Galaxy!
liuwencan:test3 liuwencan$ git add -u
liuwencan:test3 liuwencan$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   readme.md

liuwencan:test3 liuwencan$ git reset HEAD
Unstaged changes after reset:
M       readme.md
liuwencan:test3 liuwencan$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   readme.md

no changes added to commit (use "git add" and/or "git commit -a")
liuwencan:test3 liuwencan$ git diff
diff --git a/readme.md b/readme.md
index 6ced565..9b38cf7 100644
--- a/readme.md
+++ b/readme.md
@@ -1,2 +1,3 @@
 Hello World!
 Hello, Solaris!
+Hello, Galaxy!
liuwencan:test3 liuwencan$ git reset --hard HEAD
HEAD is now at 80e552e 第二次提交
liuwencan:test3 liuwencan$ git status
On branch master
nothing to commit, working tree clean
liuwencan:test3 liuwencan$
```

◆ 分支创建与切换

```
liuwencan:test3 liuwencan$ git branch
* master
liuwencan:test3 liuwencan$ ls
readme.md
liuwencan:test3 liuwencan$ git checkout -b develop
Switched to a new branch 'develop'
liuwencan:test3 liuwencan$ echo '<h2>23</h2>' > index.html
liuwencan:test3 liuwencan$ ls
index.html readme.md
liuwencan:test3 liuwencan$ git add index.html
liuwencan:test3 liuwencan$ git commit -m "增加第一个网页"
[develop 3ec3a02] 增加第一个网页
 1 file changed, 1 insertion(+)
 create mode 100644 index.html
liuwencan:test3 liuwencan$ git log
commit 3ec3a02e3b3a01cecc73dd371cddab4d3ad44819 (HEAD -> develop)
Author: LIUWENCAN <pheye@qq.com>
Date:   Sun Oct 20 11:01:51 2019 +0800

    增加第一个网页

commit 80e552e8ca5cd22f93a3e2364c9e5f9665c4a89b (master)
Author: LIUWENCAN <pheye@qq.com>
Date:   Sun Oct 20 10:47:33 2019 +0800

    第二次提交

commit e6d6ba7245c57f189ccede2e6cf23b2ea6a85008
Author: LIUWENCAN <pheye@qq.com>
Date:   Sun Oct 20 10:46:09 2019 +0800

    第一次提交
liuwencan:test3 liuwencan$ git checkout master
Switched to branch 'master'
liuwencan:test3 liuwencan$ git log
commit 80e552e8ca5cd22f93a3e2364c9e5f9665c4a89b (HEAD -> master)
Author: LIUWENCAN <pheye@qq.com>
Date:   Sun Oct 20 10:47:33 2019 +0800

    第二次提交

commit e6d6ba7245c57f189ccede2e6cf23b2ea6a85008
Author: LIUWENCAN <pheye@qq.com>
Date:   Sun Oct 20 10:46:09 2019 +0800

    第一次提交
liuwencan:test3 liuwencan$ ls
readme.md
liuwencan:test3 liuwencan$
```




部署上线

- ◆ github注册帐号
- ◆ 创建仓库train
- ◆ 推送代码
- ◆ 配置Github Pages

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Owner Repository name *


 pheyeye / train 

Great repository names are short and memorable. Need inspiration? How about [ubiquitous-engine](#)?

Description (optional)

企业实训练习演示

☒ Public  Anyone can see this repository. You choose who can commit.

☐ Private  You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.



☐ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾ | Add a license: None ▾ ⓘ

Create repository

* 请使用相同的工程名，方便作业检查

Quick setup — if you've done this kind of thing before

 Set up in Desktop or **HTTPS** **SSH** `git@github.com:pheye/train.git` 

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).


...or create a new repository on the command line

```
echo "# train" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:pheye/train.git
git push -u origin master
```



...or push an existing repository from the command line

```
git remote add origin git@github.com:pheye/train.git
git push -u origin master
```



获取工程并将第一个网页推送上去

github.com/pheye/train/settings

GitHub Pages

~~GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.~~

Your site is ready to be published at <https://pheye.github.io/train/>.

Source
Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

master branch ▼

Theme Chooser
Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

Choose a theme

Custom domain
Custom domains allow you to serve your site from a domain other than pheye.github.io. [Learn more.](#)

Save

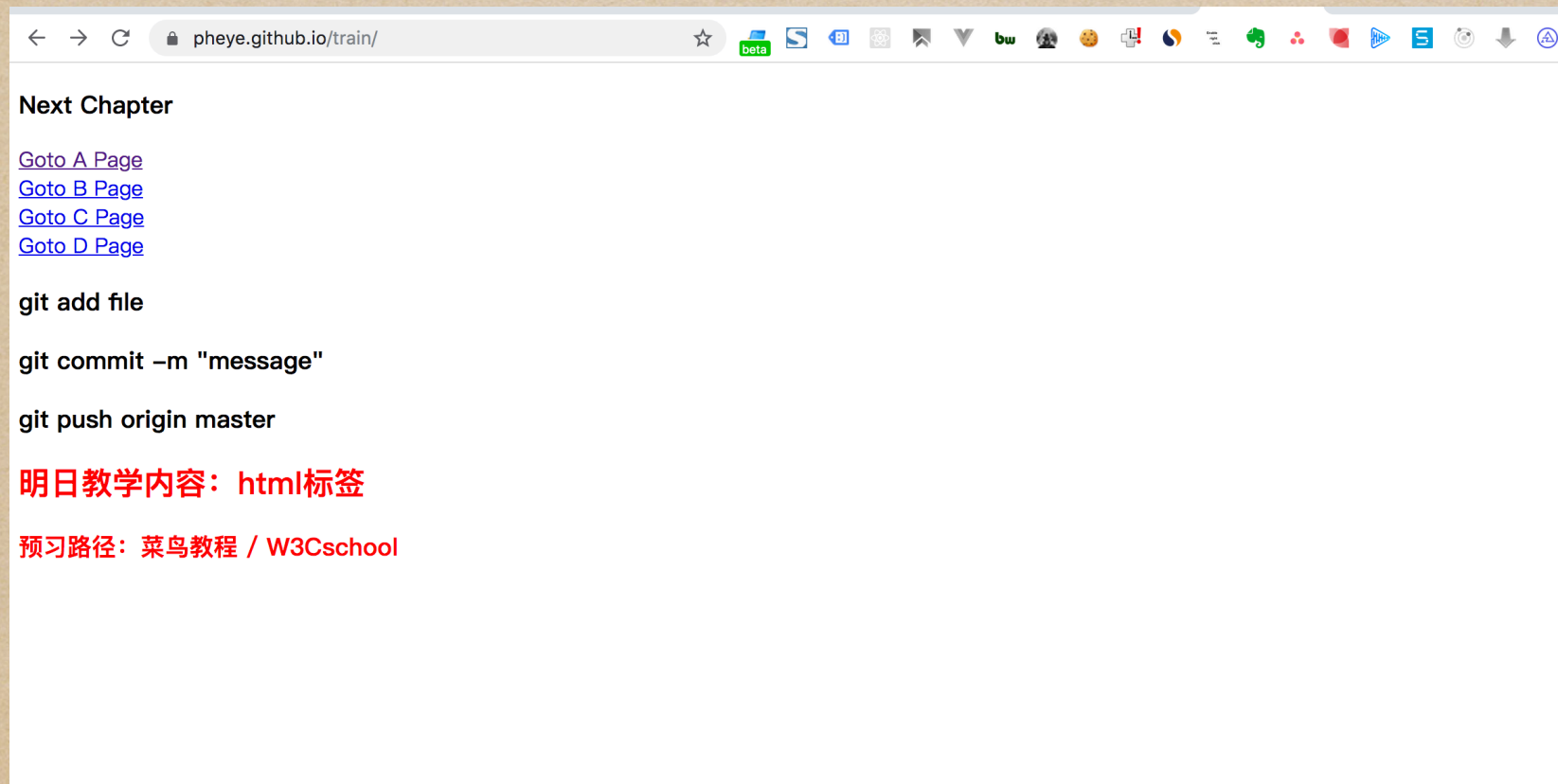
☒ **Enforce HTTPS**
Required for your site to use the default domain (pheye.github.io)

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS. [Learn more.](#)

配置GitHub Pages, 将得到域名

部署上线

- ◆ 练习：确保train工程有3个HTML文件
index.html,a.html,b.html, index.html可以正常跳转到a.html和b.html。



课后复习

- ◆ Git的用法(菜鸟教程git)
- ◆ 熟悉Linux终端基础命令(菜鸟教程Linux教程)
- ◆ 预习HTML/CSS/
JavaScript(w3school.com.cn或者菜鸟教程均可)