

gitstudy

Motoi Shimizu <shimizu.motoi@gmail.com>

Dec 14, 2012

ws24 trigger+matilde, Ogaki

この勉強会を通して...

- Gitの基本的な使い方を覚える
- GitHubを使って他人と共同作業出来る様になる

勉強会の内容

- 課題1: GitHubのアカウントを作ってください。
- 課題2: ファイル編集をして自分のremote reposにpushしてください。他の人のリポジトリをforkして編集、pull requestを送ってみて下さい。
- 課題3: SubversionとGitの違いと、Gitを使う事のメリットを教えて下さい。
- 課題4: Gitの運用モデル git-flow について教えて下さい。
- 課題5: gitstudy.codemafia.asiaにログインして、自分のリポジトリを作ってください。
- 課題6: 課題5で作ったリポジトリへhtmlファイルをpushすると、
gitstudy.codemafia.asia/{自分のdir} にpush内容が反映されるようにしてください。

- Gitの基本的な使い方を覚える
- GitHubを使って他人と共同作業出来る様になる

基本...



- ソースコードの編集履歴、差分確認が容易にできる
- ソースコードを過去のバージョンに一瞬で戻せる。失う事を怖がらずに強火で進める
- 複数人で同じソースコード編集した場合に自動でマージしてくれる（ある程度）

- Gitの基本的な使い方を覚える
- GitHubを使って他人と共同作業出来る様になる

基本...



- 名前の通り、GitのハブになるWebサイト
- 色々なプロジェクトが公開されてる（コピペばかりはダメ!）
- 個人アカウント（有償・無償あり）にいくつかrepository作れる

課題 1

- 課題 1: GitHubのアカウントを作ってください。



<http://github.com>

課題 2

- 課題 2: ファイル編集をして自分のremote reposにpushしてください。

(reference: <http://git-scm.com/docs>)

reposを複製する

\$ git clone

例:

\$ git clone https://github.com/motoishmz/mo_tilde.git

※URLは自分のreposに差し替えて下さい。

課題 2

- 課題 2: ファイル編集をして自分のremote reposにpushしてください。

(reference: <http://git-scm.com/docs>)

新規 or 編集済みfileをStageへ上げる

\$ git add

新規追加したファイルや、編集したファイルをGit管理下に置く。

例: workng dir以下全てのファイルをaddする

\$ git add .

課題 2

- 課題 2: ファイル編集をして自分のremote reposにpushしてください。

(reference: <http://git-scm.com/docs>)

Stageに上がった変更を記録

\$ git commit -m “.....”

.git/objects/.... に記録されて、indexが作られる。

例:

```
$ git commit -m “bugfix: IBActions connection conflict”
```

課題 2

- 課題 2: ファイル編集をして自分のremote reposにpushしてください。
(reference: <http://git-scm.com/docs>)

indexをremoteへ送る

\$ git push

例:

\$ git push origin master

origin... remoteのserver情報ショートカット (.git/config に書いてある)

master... branch名

- 課題 2: ファイル編集をして自分のremote reposにpushしてください。
(reference: <http://git-scm.com/docs>)

課題 2

branchを切る

\$ git branch

#まずはブランチ作成

```
git branch branch_test
```

#作ったブランチに作業を移行

```
git checkout branch_test
```

#修正を加えてコミットしてから

```
git commit -a // add allしてからcommitのショートカット
```

#修正を本体にマージする

```
git checkout master  
git merge branch_test
```

課題 3

- 課題 3: SubversionとGitの違いと、Gitを使う事のメリットを教えてください
- こういうツールは一般的に**CVS**と呼ばれる
- Subversion等の中央集権型: 1つの**repos**の消滅 = 全記録消滅
- Git等の分散型: 全員が同価値の**repos**を持つ事が出来る

課題 4

- 課題 4: Gitの運用モデル git-flow について教えてください

分散型、でも中央集権...!

Main branches: master, develop
Support branches: release, feature-*, hotfix-*

A successful Git branching model <http://nvie.com/posts/a-successful-git-branching-model/>

A successful Git branching model を翻訳しました <http://keijinsonyaban.blogspot.jp/2010/10/successful-git-branching-model.html>

課題 5

- 課題 5: gitstudy.codemafia.asiaに、自分のリポジトリを作ってください

ログイン情報

hostname: gitstudy.codemafia.asia
username: mo_tilde
password: trigger

Web公開dir

/home/mo_tilde/gitstudy.codemafia.asia/public

使ってるレンタルサーバー

<http://dreamhost.com/>

各自の公開URL

(課題6で使います)

<http://gitstudy.codemafia.asia/{今から作るrepos名}/>

server

\$ ssh mo_tilde@gitstudy.codemafia.asia

[heze]\$ ls

Maildir gitstudy.codemafia.asia logs

[heze]\$ mkdir Repos

[heze]\$ ls

Maildir Repos gitstudy.codemafia.asia logs

[heze]\$ cd Repos/

[heze]\$ mkdir motoishmz.git

[heze]\$ cd motoishmz.git/

[heze]\$ git init --bare --shared=true

Initialized empty shared Git repository in /home/mo_tilde/Repos/motoishmz.git/

local

\$ cd Desktop

\$ mkdir Test

\$ cd Test

\$ pwd

/Users/{ユーザー名}/Desktop/Test

\$ git init

Initialized empty Git repository in /Users/{ユーザー名}/Desktop/Test/.git/

\$ ls

\$ pwd

/Users/{ユーザー名}/Desktop/Test

\$ ls -a

. .. .git

\$ echo "this is a readme.md" > readme.md

\$ ls

readme.md

\$ git add .

\$ git status

On branch master

#

Initial commit

#

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

#

new file: readme.md

#

\$ git commit -m'initial commit'

[master (root-commit) edd40ee] initial commit

1 file changed, 1 insertion(+)

create mode 100644 readme.md

\$ git remote add origin ssh://mo_tilde@gitstudy.codemafia.asia/home/mo_tilde/Repos/motoishmz.git

\$ git push origin master

mo_tilde@gitstudy.codemafia.asia's password:

Counting objects: 3, done.

Writing objects: 100% (3/3), 236 bytes, done.

Total 3 (delta 0), reused 0 (delta 0)

To ssh://mo_tilde@gitstudy.codemafia.asia/home/mo_tilde/Repos/motoishmz.git

* [new branch] master -> master

課題 6

- 課題6で作ったリポジトリへhtmlファイルをpushすると、
gitstudy.codemafia.asia/{自分のdir} push内容が反映されるようにしてください

!!!: 以下、本当はアウトな部分あるけど
今回コッソリお願いします

```
// man githooks 参考になる
```

```
## server
```

```
$ ssh mo_tilde@gitstudy.codemafia.asia
```

```
[heze]$ cd gitstudy.codemafia.asia/public/
```

```
[heze]$ git clone /home/mo_tilde/Repos/motoishmz.git
```

```
Cloning into motoishmz...
```

```
done.
```

```
[heze]$ cd /home/mo_tilde/Repos/motoishmz.git/hooks/
```

```
[heze]$ ls
```

```
applypatch-msg.sample post-commit.sample post-update.sample pre-commit.sample
```

```
prepare-commit-msg.sample
```

```
commit-msg.sample post-receive.sample pre-applypatch.sample pre-rebase.sample
```

```
update.sample
```

```
[heze]$ cp post-receive.sample post-receive
```

```
[heze]$ vi post-receive
```

```
#!/bin/sh
```

```
#
```

```
# An example hook script for the "post-receive" event.
```

```
#
```

```
# The "post-receive" script is run after receive-pack has accepted a pack
```

```
# and the repository has been updated. It is passed arguments in through
```

```
# stdin in the form
```

```
# <oldrev> <newrev> <refname>
```

```
# For example:
```

```
# aa453216d1b3e49e7f6f98441fa56946ddcd6a20 68f7abf4e6f922807889f52bc043ecd31b79f814
```

```
refs/heads/master
```

```
#
```

```
# see contrib/hooks/ for a sample, or uncomment the next line and
```

```
# rename the file to "post-receive".
```

```
#. /usr/share/doc/git-core/contrib/hooks/post-receive-email
```

```
(cd /home/mo_tilde/gitstudy.codemafia.asia/public/motoishmz; git --git-dir=.git pull)
```

gitstudy

Motoi Shimizu <shimizu.motoi@gmail.com>

Dec 14, 2012

ws24 trigger+matilde, Ogaki