

はじめに

Gitをインストールして、Githubアカウントを作成してください。

Mac OS X の人はデフォルトで入っていますが、Homebrew (<http://brew.sh>) でインストールして使うと管理が楽でうれしい。

```
$ which git
/usr/bin/git
$ git --version
git version 1.9.3 (Apple Git-50)
$
$ brew install git
$ git --version
git version 2.1.0
```

Windowsの人は <http://git-scm.com> からインストーラをダウンロードしてインストールすればいいのではないのでしょうか。

Githubアカウントは (<https://github.com>) から登録します。
IDはRGのログイン名みたいにわかりやすいのにしといた方がいい。

Gitとは

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

- <http://git-scm.com>



Linux Kernelのソースコードを管理する目的で
Linus Benedict Torvaldsが開発

- * レポジトリという単位でバージョン管理を行うアプリ
- * リモートサーバに常に最新のバージョンを置いておく
- * 全ユーザが全履歴を含むレポジトリの完全な複製を保持
- * CIツールやデプロイツールと併用することで自動化に貢献



Githubとは

*Build software better, together.
Powerful collaboration, code review, and code
management for open source and private projects.*
- <http://github.com>

Github社が運営する、世界で最も使われている
Gitホスティングサービス



- * オープンソースなら無料のGitリモートサーバとして利用可能
- * バージョン履歴などの情報をWebベースで視覚化
- * ForkやPull Requestによるプロジェクトワークフローの提供
- * IssueやMilestoneによるチケット機能の提供
- * コードをベースとしたSNS的な使い方が可能

GitとGithubでなにができるか

オープンソースソフトウェアの開発

例えばRuby on RailsはGithub上で開発されている
VimmerはとてもGithubを活用する

社内でのプロジェクト管理に利用

Yahoo! Japan, CookPad, CyberAgent, はてな, ドワンゴなど
日本でも多くの企業で利用されている

ドキュメントやチュートリアル公開

AngularJSのチュートリアルはGithub上で管理されている
mixiがMarkdownで書いてGithubに上げたマニュアルが一時期話題になった

業務委託や採用時の判断に利用

プログラマの力量を測るためにGithub上のコードを見る企業が増加中
CyberAgentは「Github採用」を始めている

Gitのはじめかた (1 / 2)

Gitで管理を始める

.gitディレクトリが作成され、カレントディレクトリ以下のバージョン情報がそこへ保存されるようになる。

```
$ cd /path/to/project/root
```

```
$ git init
```

```
Initialized empty Git repository in /path/to/project/root
```

Gitでの管理を終える

```
$ cd path/to/project/root
```

```
$ rm -rf .git
```

単に情報が保存されている.gitディレクトリを消去すればGitでの管理が終了する

Gitのはじめかた (2/2)

Gitの設定をする

\$HOME/.gitconfigファイルにGitの設定が保存される

```
$ git config --global user.name "Atsuki Demizu"  
$ git config --global user.email demmy@sfc.wide.ad.jp  
$ git config --global core.editor vim
```

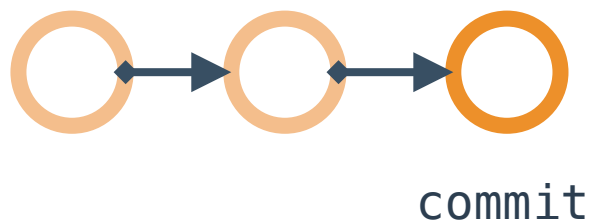
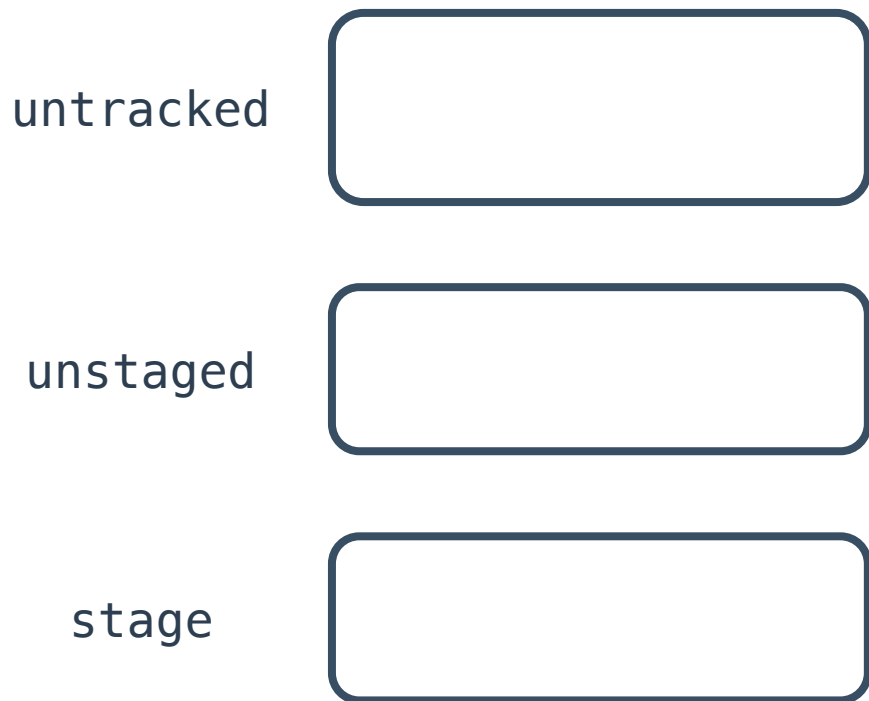
Gitの状態を確認する

```
$ git status  
On branch master  
  
Initial commit
```

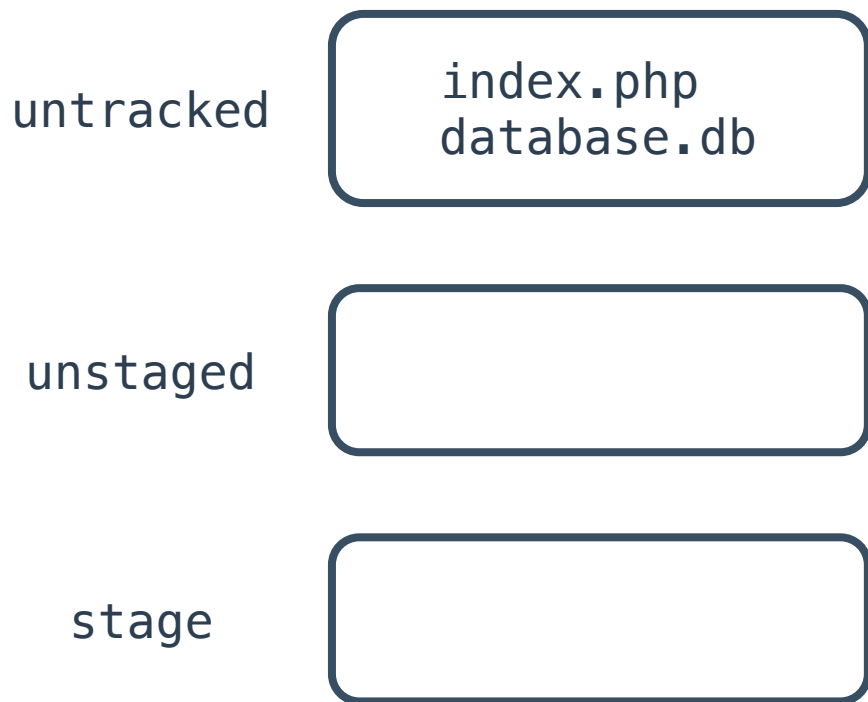
困ったら叩く。
書かれていることをちゃんと読めば大抵次にやるべきことが分かる。

```
nothing to commit (create/copy files and use "git add" to track)
```

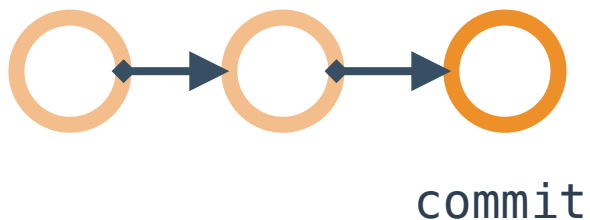
Gitの基本的な使い方 (1 / 7)



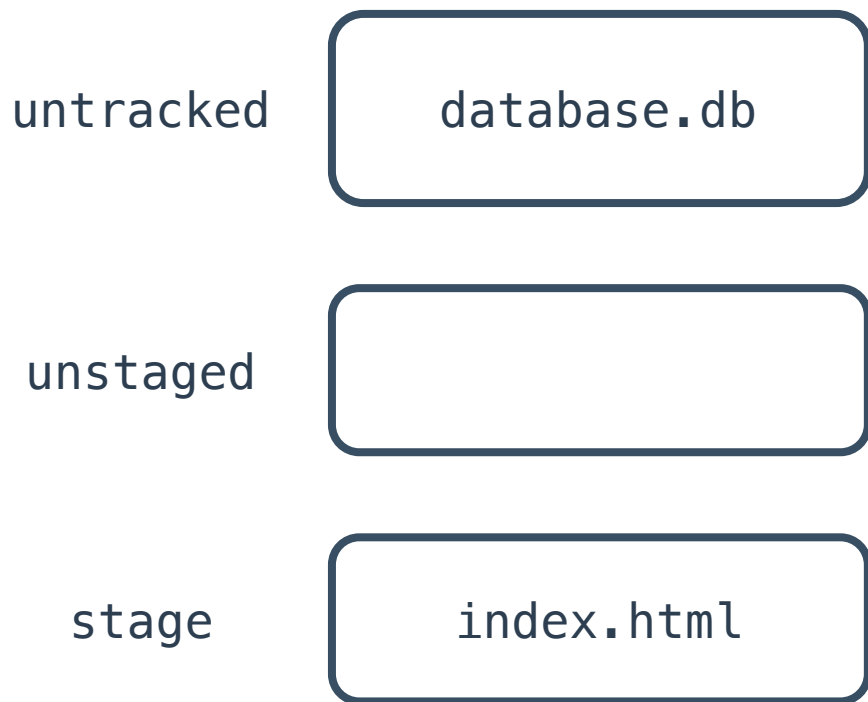
Gitの基本的な使い方 (2/7)



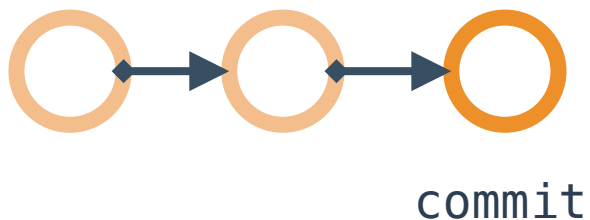
```
$ touch index.php database.db  
$
```



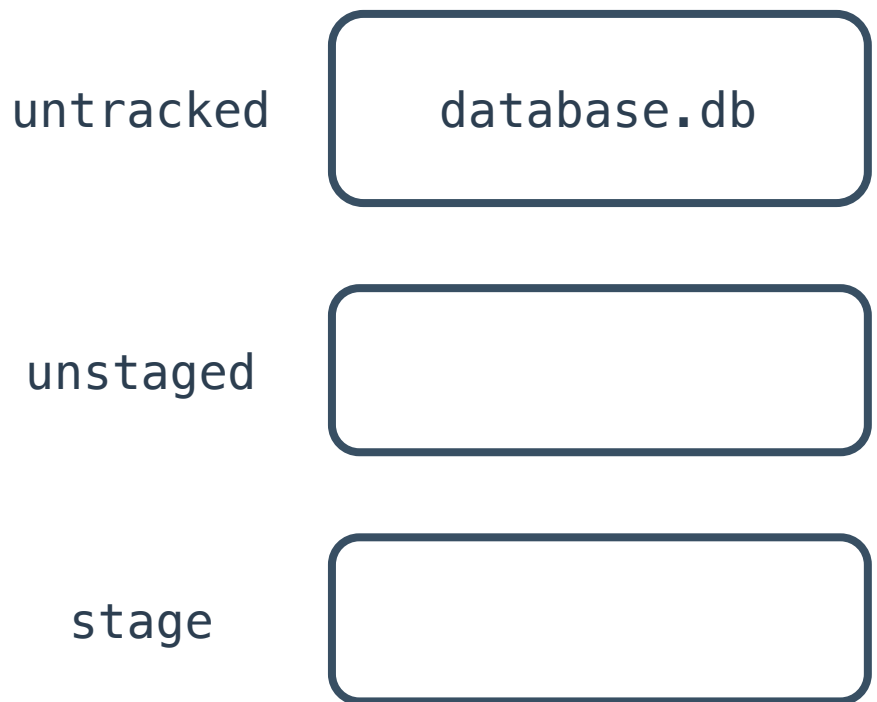
Gitの基本的な使い方 (3/7)



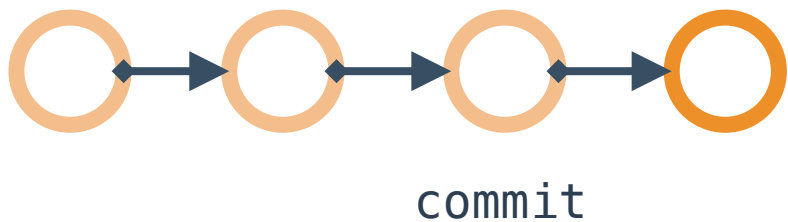
```
$ touch index.php database.db  
$ git add index.php  
$
```



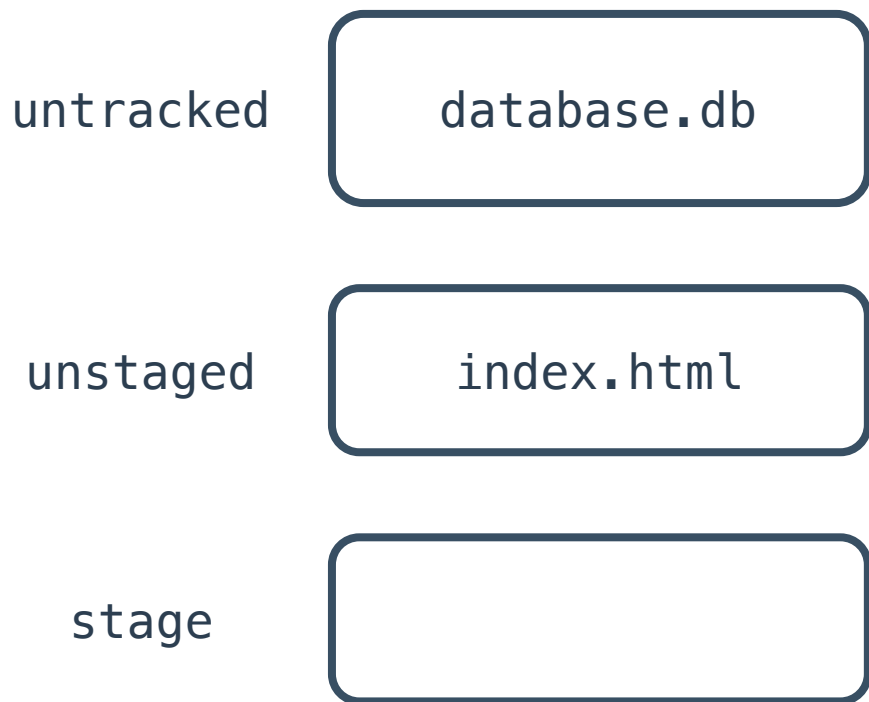
Gitの基本的な使い方 (4/7)



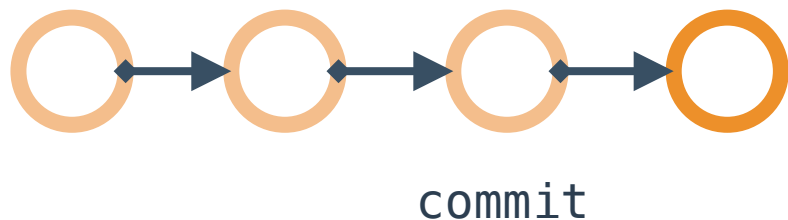
```
$ touch index.php database.db  
$ git add index.php  
$ git commit  
$
```



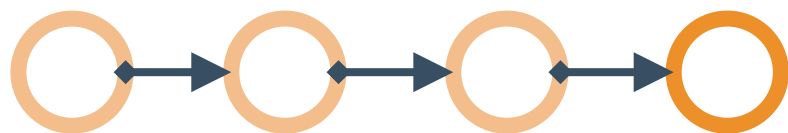
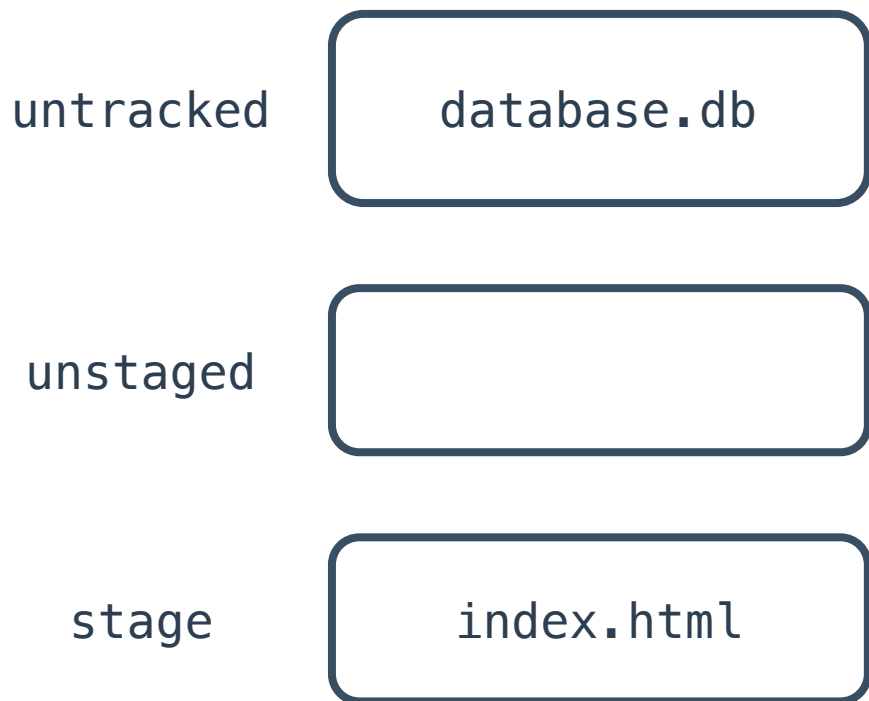
Gitの基本的な使い方 (5/7)



```
$ touch index.php database.db  
$ git add index.php  
$ git commit  
$ vim index.html  
$
```



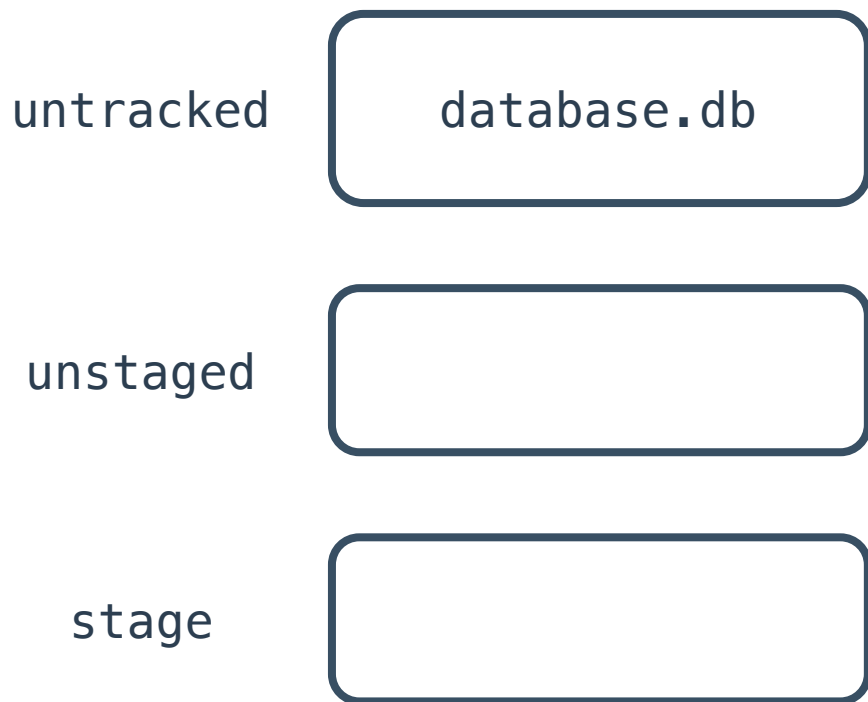
Gitの基本的な使い方 (6/7)



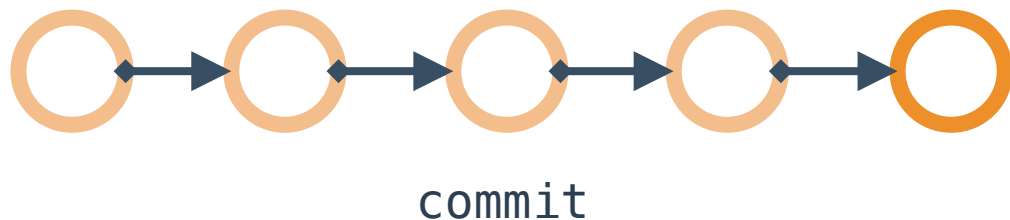
commit

```
$ touch index.php database.db  
$ git add index.php  
$ git commit  
$ vim index.html  
$ git add index.php  
$
```

Gitの基本的な使い方 (7/7)



```
$ touch index.php database.db  
$ git add index.php  
$ git commit  
$ vim index.html  
$ git add index.php  
$ git commit  
$
```





便利なコマンド

これまでのコミットを確認する

```
$ git log
```

unstagedなファイルの変更を確認する

```
$ git diff
```

trackedなファイルの中を検索する

```
$ git grep <regexp>
```

ブランチ

新しいブランチを作成する

```
$ git branch develop
```

すべてのブランチを確認

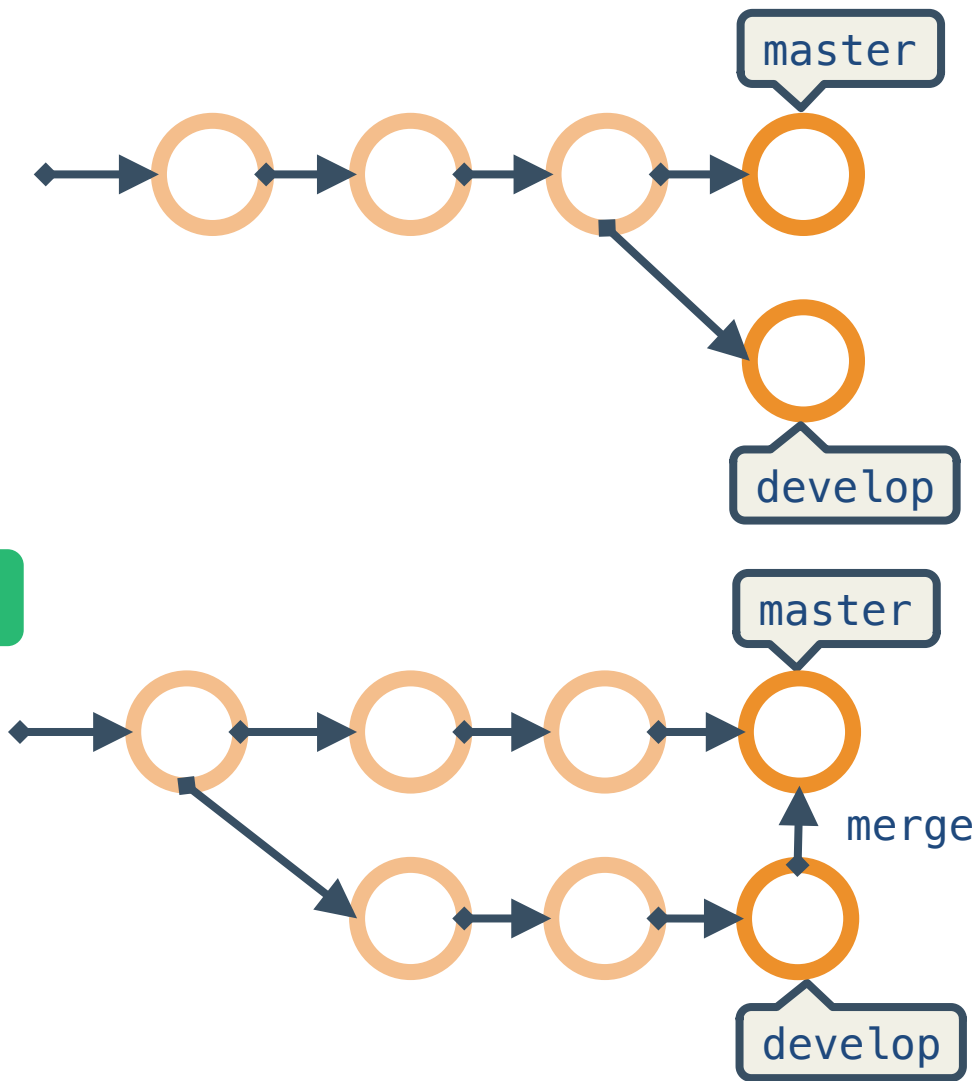
```
$ git branch
```

現在のブランチを切り替える

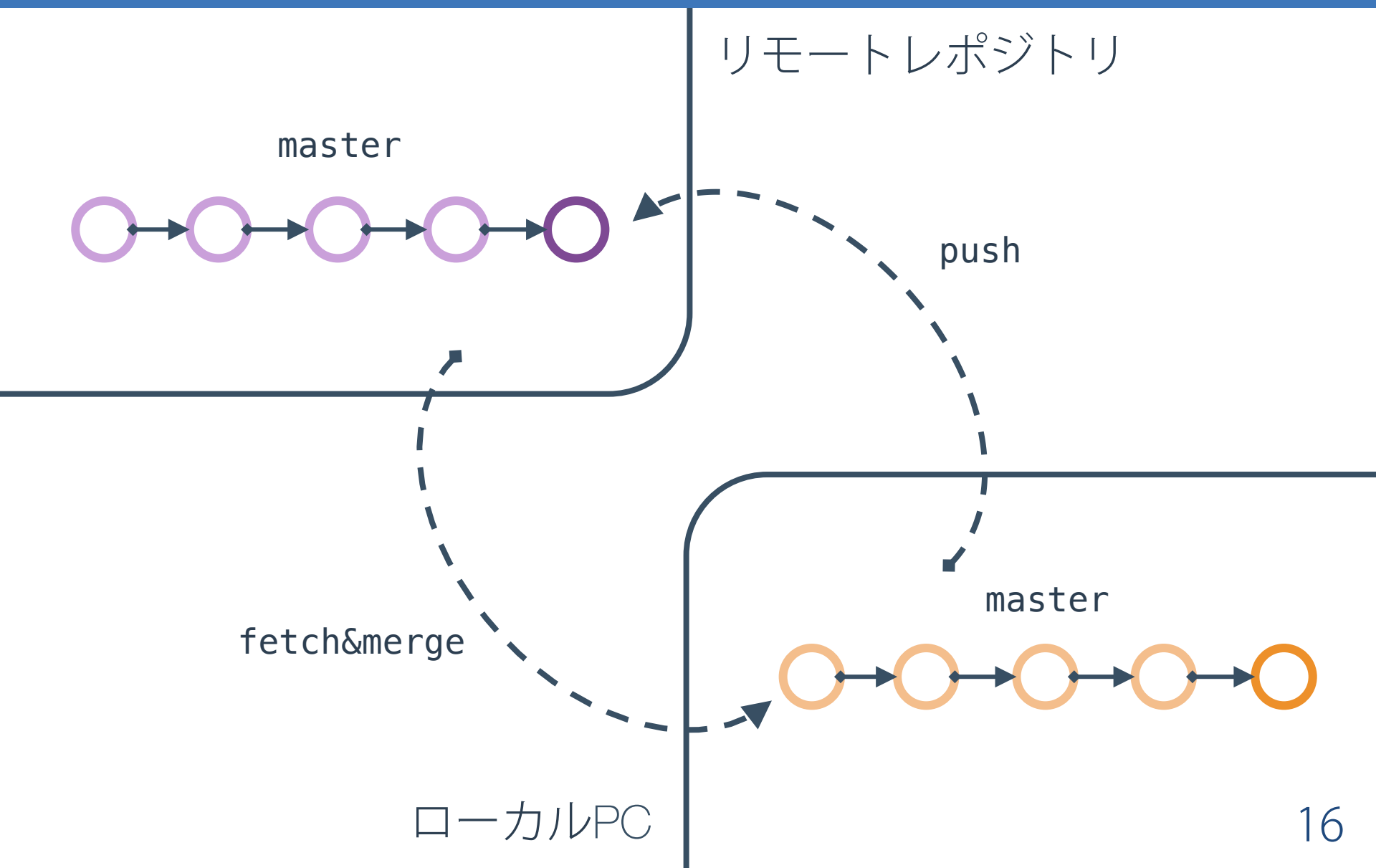
```
$ git checkout develop
```

ブランチを統合する

```
$ git merge develop
```



GitとGithubを使用した開発



Githubの基本的な使い方

SSH鍵の作成・登録

```
$ ssh-keygen
```

レポジトリの作成・紐付け

```
$ git remote add origin git@github.com:demmys/sample.git
```

リモートレポジトリへの送信

```
$ git push origin master:master
```

他の人の変更をコピーしたブランチを作成

```
$ git fetch origin
```