gitについて学ぼう！

プロジェクトの作成

git init

リポジトリの複製

git clone https://github~~

コミット状況の確認など

git status

ログ関連

全ログ表示

git log

コメントのみ

git log --oneline

統計表示

git log --stat

変更表示

git log -p

統計と変更表示

git log -p --stat

空白の変更を無視

git log -p -w

SHAの指定

git log -p fdf5493

最新コミットのみ表示

git show

特定のコミットを指定して表示

git show fdf5493

ステージング

ファイルを指定してステージング

git add index.html

ステージング インデックスから削除

git rm --cached

階層すべてステージング

git add .

コミット

詳細に記述(Editorが開く)

git commit

コメント付きでコミットする。

git commit -m "Initial commit"

良いコミットメッセージ

1. メッセージは短くしてください (60 文字未満)
2. コミットが何をするのかを説明してください（方法や理由ではありません！）

コミット時しないこと

1. 変更が行われた理由を説明しないでください
2. 変更がどのように行われるかを説明しないでください (git log -pが目的です!)
3. 「and」という言葉を使わない 「and」を使用する必要がある場合は、コミット メッセージで行われている変更が多すぎる可能性があります。
   変更を別々のコミットに分割します。
   例: 「背景色をピンクにしてサイドバーのサイズを大きくする」

差分(変更前ファイルのみ)

git diff

ステージングの無視

.gitignore　を作成する。
中に以下のように記述をするとステージングしなくなる。
project.docx
正規表現を使うことで複数無視できる。
samples/*.jpg

```
blank lines can be used for spacing
# - marks line as a comment
* - matches 0 or more characters
? - matches 1 character
[abc] - matches a, b, _or_ c
** - matches nested directories - a/**/z matches
a/z
a/b/z
a/b/c/z
```

Tag

tag名を付ける

git tag -a v1.0

tagの確認

git tag

tagの削除

git tag -d v1.0

過去の変更にタグをつける

git tag -a v1.0 a87984(SHA)

ブランチ

今いるブランチの確認

git branch

ブランチの作成

git branch sidebar

チェックアウト

git checkout sidebar

SHA指定でブランチ作成

git branch alt-sidebar-loc 42a69f

ブランチ削除

git branch -d sidebar

ブランチ強制削除

git branch -D sidebar

チェックアウトコマンドでブランチを作成(ブランチへ移動込み)

git checkout -b footer master

グラフの表示とすべての変更履歴の表示

git log --oneline --decorate --graph --all

マージ

ブランチとマージ

git merge sidebar

間違ったブランチにマージした場合

git reset --hard HEAD^

最後のコミットを変更

git commit --amend

コミットを元に戻す

git revert SHA

コミットのリセット

git reset

フラグ

- --mixed
- --soft
- --hard

バックアップ

git branch backup
git checkout -- index.html
git merge backup

意図せぬブランチに編集を加えた場合

git restore .

リモートリポジトリ作業(Github関連)

現在のリモートリポジトリの確認

git remote -v

リモート先の変更

リモート先の削除

git remote remove origin

リモート先の設定

git remote add origin {url}

リモートへPush(転送)

git push -u origin myrepo

Github系

用語

PullReqest (PR)

自分の開発したブランチを結合してもよいかリクエストを送る。
承認者を設定することができ意図していないコードが紛れ込んでいないか
他者の目を通すことで安全性を確立できる。

マージ

PullReqestを確認しコードの結合を行う。
結合の際にGithubでコミュニケーションを行いながら開発するのが一般的。

Issue

不具合、改善要望、不明点などコミュニケーションに利用する。 jiraとgithubどちらで不具合を管理するかなどはプロセスによって違う。