

01_2 基本操作

01_2 基本操作

エディタ編

vscodeの起動

基本的な考え方・操作

基本のキーバインド

Gitの基本操作

具体例

練習1

練習2

この練習で学べること

練習事項

エディタ編

vscodeの起動

- デスクトップ上のショートカット等通常の起動
 - Windows上での普通の起動
- Anaconda Navigator 上からの起動（Navigatorの起動に時間がかかるのでお勧めしない）
 - anaconda配下の設定で動作する
- Anacondaのpython仮想環境で作業を行う
 - ターミナル内で以下のコマンドを実行

```
conda activate
```

```
conda activate 環境名
```

- ターミナルをエディタ内で起動した場合に、`(base) C:\Users\Username` のように表示され仮想環境上で実行可能していることが表示される

基本的な考え方・操作

- vscodeは、プロジェクト管理をフォルダ単位で行なう。
 - 単純にファイルを開いた場合は、すべての機能が利用できない。
 - 作業用のフォルダ（ディレクトリ）を作成し、その中にファイルを作成、管理する。
 - 作業用フォルダを「src」とする（好きな名前でOK）
 - このフォルダ以下にプロジェクトごとのフォルダを作成して管理すると良い。
 - ドキュメント内に直接プロジェクトごとのフォルダを作成しても良いが、散らばらないように管理すること。

基本のキーバインド

- 別表「[keyboard-shortcuts-windows.pdf](#)」を参照

- ヘルプ→キーボードショートカットの参照 から入手可 (`ctrl + K` `ctrl + R`)
- エディタ上では、 `ctrl + K` `ctrl + S` で一覧表示される。
- 最低限覚えておきたいショートカットキー
 - `ctrl + x` , `ctrl + c` , `ctrl + v` 通常のWindowsの編集
 - `ctrl + s` , `ctrl + shift + s` ファイルの保存、別名で保存
 - `ctrl + f` , `ctrl + h` 検索、置換
- 早く覚えて欲しいショートカットキー
 - `ctrl + d` , `ctrl + k` `ctrl + d` 選択領域の検索追加、Skip
 - `alt + click` マルチカーソル追加
- その他、利用できるショートカットを少しずつ増やしていきましょう。

Gitの基本操作

- 初期化する
- 忘れず「.gitignore」ファイルを作成する。
 - 授業用minimum.gitignore (dot.gitignore)

```
.mypy_cache/  
.vscode/  
__pycache__/
```

- 規模が大きくなると、記述する内容が増えるので雛形を利用するのも良い

<https://github.com/github/gitignore>

- .gitignoreの記述方法

<https://www.sejuku.net/blog/72389>

<https://qiita.com/inabe49/items/16ee3d9d1ce68daa9fff>

- git管理した後.gitignoreファイルを作成しても、すでに管理しているファイル群に関しては、管理対象から外れない。
 - そのため、なるべく早めに作成しておくのが良い。
- Python.gitignore

<https://github.com/github/gitignore/blob/main/Python.gitignore>

- ファイルを作成・編集・保存
- Commitしてgitで管理する

具体例

- ファイルを編集する。
- 保存する
- ステージングする
- コミットする
 - vscodeでは、ステージング&コミットが可能

練習1

- git_test1 というフォルダを作成
 - .gitignore を作成
 - 空でも良いので作成しておく
 - sample.txt を作成
 - 3行ほど適当な文字列を書き込む
 - 保存
 - コミットする
 - sample.txt を修正
 - 2行ほど追加する
 - 保存
 - コミットする
 - Git Graphで確認する
 - checkout して、特定のコミットした状態に戻す
- 調べてみよう
 - gitで使用する言葉

- HEAD
- MASTERブランチ
- スタッシュ (stash)
- merge

- 実際に、上記で調べたものを使って練習してください。

練習2

この練習で学べること

- Gitの基本操作（初期化、コミット、修正、削除）
- ファイルのバージョン管理の流れ

練習事項

- コミット時のメッセージは、後で分かるように各自記述すること
- git_test2フォルダをgitで初期化する（使えるようにする）
- 「.gitignore」を作成する。
- ファイル、「rensyuu_1.txt」を作成する。
 - 以下のサンプル文書を入力する。

サンプルの文書

新型コロナウイルスの感染拡大を受けた政府の緊急経済対策の柱で、国民一律に給付される10万円について、
もっとも早い自治体では、来月7日にも給付できるよう準備を進めていることがわかりました。

- 保存する
- コミットする。
- ファイル「rensyuu_2.txt」を作成する。
 - 以下のサンプル文書を入力する。

サンプル文書2

吾輩は猫である。名前はまだ無い。
どこで生れたかとうと見当つかぬ。

- 保存する
- コミットする。
- 修正する。
 - 「rensyuu_1.txt」

Yahooニュース

新型コロナウイルスの感染拡大を受けた政府の緊急経済対策の柱で、国民一律に給付される１０万円について、
もっとも早い自治体では、来月７日にも給付できるよう準備を進めていることがわかりました。

- 「rensyuu_2.txt」

吾輩は猫である

夏目漱石

吾輩は猫である。名前はまだ無い。

どこで生れたかとうんと見当がつかぬ。

何でも薄暗いじめじめした所でニャーニャー泣いていた事だけは記憶している。

- どちらも保存し、コミットする。
- 再度修正する。
 - 「rensyuu_2.txt」

吾輩は猫である

夏目漱石

吾輩は猫である。名前はまだ無い。

どこで生れたかとうんと見当がつかぬ。

何でも薄暗いじめじめした所でニャーニャー泣いていた事だけは記憶している。

吾輩はここで始めて人間というものを見た。

- 保存する。
- ファイル「rensyuu_1.txt」を削除する。
- コミットする。
- この時点で、gitの状況を確認してみよう。
 - 一番最初の状態をcheckoutする。
 - 「rensyuu_1.txt」が存在しているのを確認する。

- 最後の状態をcheckoutする。
 - 「rensyuu_1.txt」が削除されているのを確認する。