

**オブジェクト指向プログラミング及び演習2**

**GitHubを使った個人開発**

# GitHubを使った開発の基本

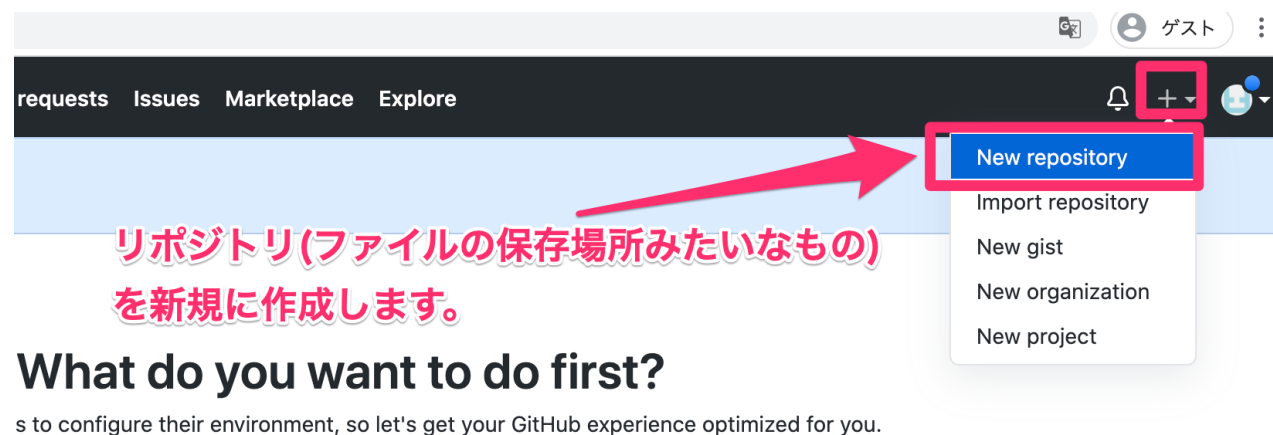
- GitHubを使うことで、複数人での開発作業だけでなく、個人での開発にも使用できます。  
基本の動きとして、個人での開発の流れを確認しましょう。

この資料では、以下の内容を学びます。

- GitHubのリポジトリ(ファイルの保管・履歴管理場所のようなもの)の作成
- ローカルファイルの管理方法(Clone, Add, Commit)
- GitHubへのファイルの追加更新作業(Push)

# リポジトリの作成

WebブラウザにてGitHubにログイン後、画面右上ユーザーアイコンの左隣の「+」ボタンより、「New Repository」を選びます。



# リポジトリ名の指定

その後表示されるページでは、リポジトリ名を入力します。

とりあえずの動作確認なので、名前に迷ったらリポジトリ名は「oop2\_practice」とでも設定しておきましょう。

The screenshot shows the GitHub 'Create a new repository' page. The browser tabs at the top are 'Sign in to GitHub · GitHub' and 'Create a New Repository'. The address bar shows 'github.com/new'. The page title is 'Create a new repository'. Below the title, it says 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'.

The 'Owner' field is set to 'Nagatani-sub'. The 'Repository name' field is set to 'oop2\_practice' and is highlighted with a red box and a red arrow pointing to it from the text '作成したいリポジトリ名を入力' (Enter the repository name you want to create). A green checkmark is visible next to the repository name.

The 'Description (optional)' field is empty.

The 'Public' option is selected, with the description 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option is also visible, with the description 'You choose who can see and commit to this repository.'

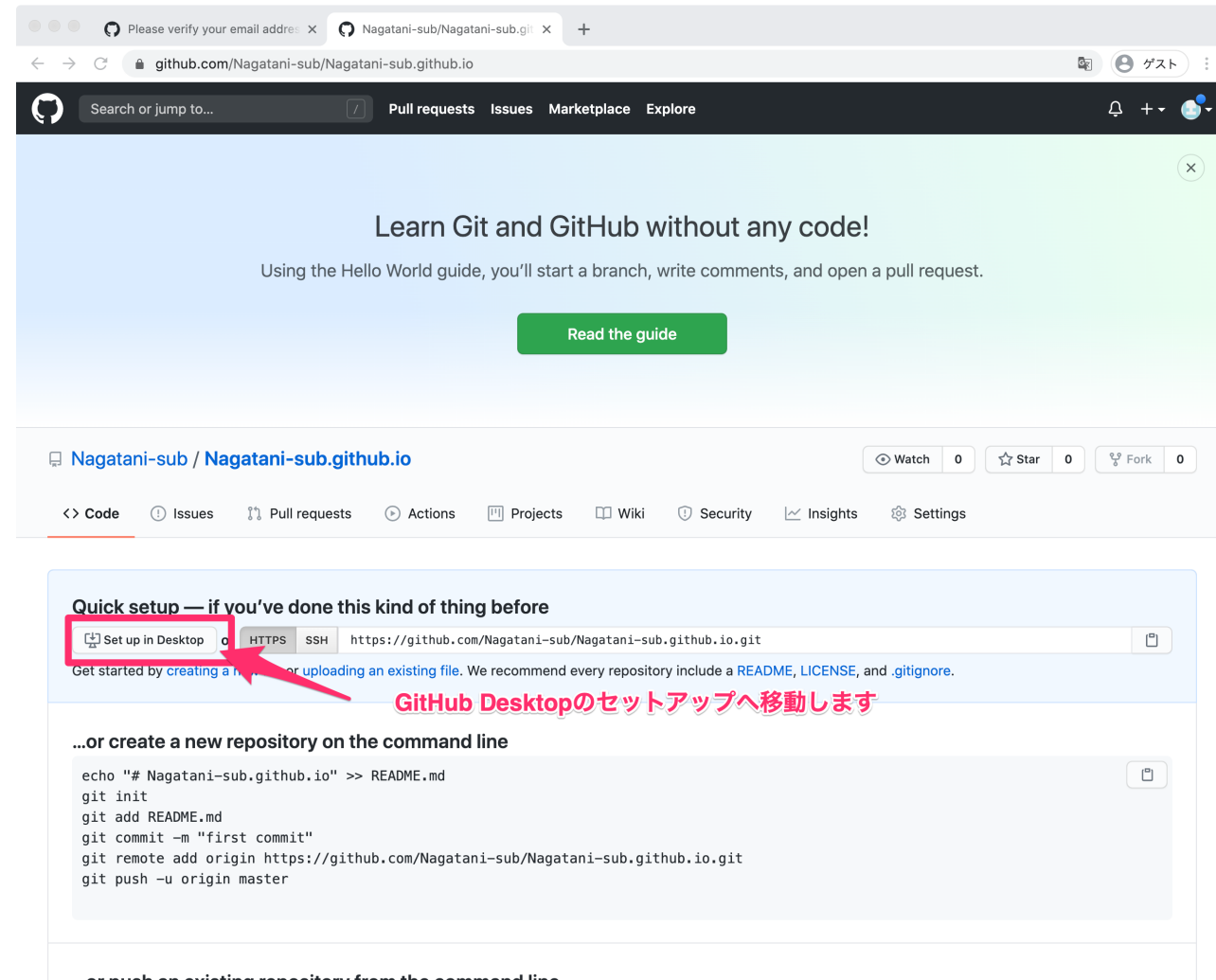
Under 'Initialize this repository with:', there are three checkboxes: 'Add a README file' (unchecked), 'Add .gitignore' (unchecked), and 'Choose a license' (unchecked). Each has a brief description and a 'Learn more' link.

At the bottom, the 'Create repository' button is highlighted with a red box and a red arrow pointing to it from the text '他に設定は変更しないで作成を行う' (Create without changing other settings).

# GitHub Desktopの用意

次は、GitHub Desktopをインストールして、公開するファイルの追加します。

まだインストールをしていない方は右図のボタンより、ダウンロードとインストールを行ってください。



# ファイルの追加編集

GitHub Desktopをインストール、起動、サインインを済ませ、以下の手順に従い、自分のPC上へリポジトリをCloneします。

1. GitHub Desktopのアプリを開いた後、画面上部のメニューより、「File」→「Clone Repository」を選択
2. GitHub.comのタブより、先程作成したリポジトリを選択
3. 自分のPC上で保存する場所をLocal Pathに設定
4. cloneボタンをクリック
  - ローカルにリポジトリのクローンが生成されます  
以降はローカルのファイルに追加変更を加えます。

# GitHub DesktopからVSCodeやFinderを開く

GitHub Desktopのメニューには、「Repository」があり、ここには以下の便利な外部ツール立ち上げ機能があります。

- **View on GitHub** ブラウザを起動して該当のリポジトリページを開く
- **Open in Terminal** ローカルの保存先をターミナルで開く
- **Show in Finder** ローカルの保存先をFinderで開く
- **Open in Visual Studio Code** ローカルの保存先をVSCodeで開く
  - これは、Preferencesで使用するテキストエディタを設定していないと使えません。

# リポジトリへの新規コミット

次はローカルに作成したリポジトリのクローンに対して、変更を加えます。

1. Repositoryメニューの「Open in Visual Studio Code」を選択してVSCoDeを起動(開いた場所を便宜上プロジェクトディレクトリと呼びます。)
2. プロジェクトディレクトリに、`modules` ディレクトリを作成
3. 次項で示す2つのコードを下記の場所にそれぞれ作成
  - `modules` ディレクトリに `SayHello.py` を作成
  - プロジェクトディレクトリ直下に `main.py` を作成



## modules/SayHello.py

```
class SayHello:
    def __init__(self, target="World"):
        self.target = target

    def say(self):
        print(f"Hello, {self.target}!!")

if __name__ == '__main__':
    app = SayHello()
    app.say()
    app = SayHello("Someone")
    app.say()
```

## main.py

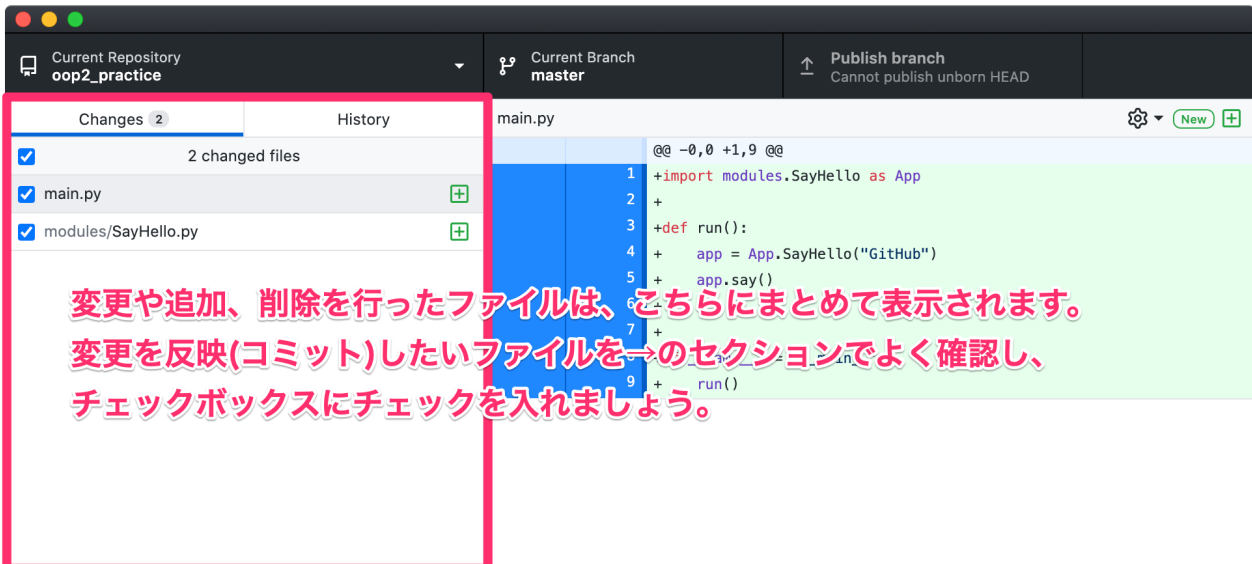
```
import modules.SayHello as App

def run():
    app = App.SayHello("Git")
    app.say()

if __name__ == '__main__':
    run()
```

作成できたら手元で動作確認の上、次項の手順に従い、インターネット上のリポジトリにプッシュしましょう。

# 変更内容を適用(Commit)し、それをリモートへ反映(Push)させます。



Changes 2 | History

2 changed files

- main.py
- modules/SayHello.py

main.py

```
@@ -0,0 +1,9 @@
1 +import modules.SayHello as App
2 +
3 +def run():
4 +     app = App.SayHello("GitHub")
5 +     app.say()
6 +
7 +
8 +
9 +run()
```

変更や追加、削除を行ったファイルは、こちらにまとめて表示されます。  
変更を反映(コミット)したいファイルをこのセクションでよく確認し、  
チェックボックスにチェックを入れましょう。

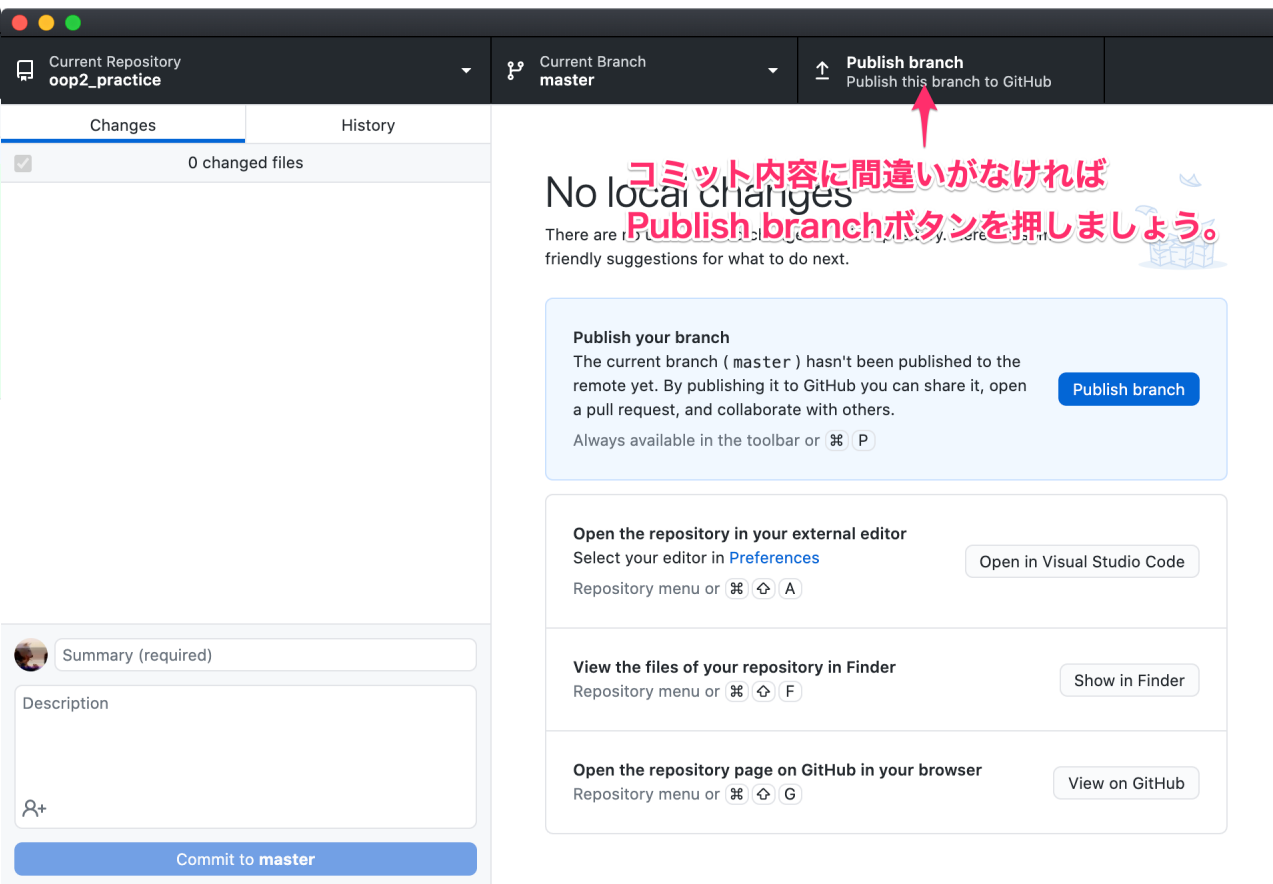
Summary (required)

Description

Commit to master

ファイルの変更に対する要約を入力します。  
今回は最初のコミットなので「最初のコミット」  
などと入力しておきましょう。

要約を書き終えたら、コミットします。



Changes | History

0 changed files

No local changes

There are no local changes. If you're ready to share your work, click the Publish branch button.

**コミット内容に間違いがなければ Publish branch ボタンを押しましょう。**

Publish your branch

The current branch ( master ) hasn't been published to the remote yet. By publishing it to GitHub you can share it, open a pull request, and collaborate with others.

Always available in the toolbar or P

Open the repository in your external editor

Select your editor in [Preferences](#)

Repository menu or A

Open in Visual Studio Code

View the files of your repository in Finder

Repository menu or F

Show in Finder

Open the repository page on GitHub in your browser

Repository menu or G

View on GitHub

Commit to master

# コードを変更する

プッシュが終わると、GitHubのリポジトリにアップロードされているのが確認できるはずです。Repositoryメニューの「View on GitHub」で確認しましょう。

先程作成したプログラムにて、`Hello, Git!!`と出力している箇所を`Hello, GitHub!!`となるように変更しましょう。

変更を保存して、動作確認を行ったら、再度GitHub Desktopに画面を切り替え、次項での説明を読みながら変更をコミット、プッシュしましょう。

# Push後、GitHubのページをリロードして確認しましょう。

Current Repository: oop2\_practice

Current Branch: master

Fetch origin: Last fetched just now

Changes: 1 changed file

main.py

```
@@ -1,7 +1,7 @@
1 import modules.SayHello as App
2
3 def main():
4     app = App.SayHello("Git")
5     app = App.SayHello("GitHub")
6     app.say()
7
```

変更内容をよく確認しましょう。

変更の要約も詳しく記入します。  
必要に応じてDescriptionの方にも詳細な変更を書く場合もあります。

update: Glt -> GitHub

Commit to master

Current Repository: oop2\_practice

Current Branch: master

Push origin: Last fetched 3 minutes ago

Changes: 0 changed files

No local change. はこちらでPushします。

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Push commits to the origin remote

You have 1 local commit waiting to be pushed to GitHub.

Always available in the toolbar when there are local commits waiting to be pushed or `git push`

Push origin

Open the repository in your external editor

Select your editor in [Preferences](#)

Repository menu or `Cmd + A`

Open in Visual Studio Code

View the files of your repository in Finder

Repository menu or `Cmd + F`

Show in Finder

Open the repository page on GitHub in your browser

Repository menu or `Cmd + G`

View on GitHub

Summary (required)

Description

Commit to master

Committed just now

update: Glt -> GitHub

Undo

コミットが間違っていなければこちらでPushします。

間違えてコミットしてしまった場合、Push前ならここで取り消せます。

# 個人開発におけるGitHubの役割

ソースコードの変更ログを残すことができることと、いざというときのバックアップ、他の人へ自分が作ってきたものを見せる場にもなります。積極的にインターネットへ公開しましょう。

もちろん、以下のような注意点もあります。ご注意ください。

- Publicなリポジトリは、インターネット上で公開されている
  - 法的に問題のあるコード(他者の著作権を無視する,攻撃性の高いものを攻撃性があるようにして公開するなど)を公開しないこと
  - 他のシステムへの認証キーなどを誤って公開しないこと

# GitHubでの個人開発まとめ

個人開発では、コードを書いてデバッグしてはインターネットへ公開する単純な作業です。

普段のプログラミング(だけじゃなくてもOK)でも、GitHubのプライベートなりポジトリを作成し、そこへPushしておく癖をつけると、「あのコードどこで書いたっけ...」ということが少なくなります。さらに、PCの故障の際にも強くなります。

個人開発は一人での作業なので予期せぬコードの重複や衝突などは起きにくいためシンプルですが、複数人での開発となると途端に難易度が上がります。が、複数人での開発にはそれに合わせた開発手法があります。次の資料からは、グループでの作業となります。