Git-Flow

Git Flow Cheat Sheet

Git Flow is a set of Git extensions that provide high-level repository operations.

[Sourcetree](#) is an excellent Git GUI and provides Git Flow support.

Initialization

Git Flow needs to be initialized to customize your project setup:

✏️ **Shell**

```shell
git flow init
```

Features

Creating a Branch

✏️ **Shell**

```shell
git flow feature start MYFEATURE
```

- Creates a new feature branch based on `develop`.
- Automatically switches to the new branch.

Merging

✏️ **Shell**

```shell
git flow feature finish MYFEATURE
```

- Merges `MYFEATURE` into `develop`.
- Deletes the feature branch.
- Switches back to the `develop` branch.

Publishing

✏ **Shell**

```
git flow feature publish MYFEATURE
```

- Publishes a feature to the remote repository so others can collaborate on it.

Pulling

✏ **Shell**

```
git flow feature pull origin MYFEATURE
```

- Retrieves a feature branch published by another user.

Tracking

✏ **Shell**

```
git flow feature track MYFEATURE
```

- Tracks a feature on the remote repository.

Make a Release

Starting a Release

✏ **Shell**

```
git flow release start RELEASE [BASE]
```

- Creates a release branch from `develop`.

Publishing a Release

```shell
git flow release publish RELEASE
```

- Publishes the release branch to allow other developers to make release commits.

Tracking a Release

```shell
git flow release track RELEASE
```

- Tracks a remote release branch.

Finishing a Release

```shell
git flow release finish RELEASE
```

- Merges the release branch into `master` and `develop`.
- Tags the release with its name.
- Deletes the release branch.

> **Note**: Don't forget to push your tags:
>
>
> ```shell
> git push --tags
> ```

Hotfixes

Hotfixes are used to address urgent issues in production.

Starting a Hotfix

✎ **Shell**

```shell
git flow hotfix start VERSION [BASENAME]
```

- Creates a hotfix branch from `master`.
- The `VERSION` argument specifies the new hotfix version.

Finishing a Hotfix

✎ **Shell**

```shell
git flow hotfix finish VERSION
```

- Merges the hotfix branch into `master` and `develop`.
- Tags the `master` branch with the hotfix version.
- Deletes the hotfix branch.

Most Important Commands

- Git Flow is a tooling collection; you can still use Git commands normally.
- The `support` feature is in beta and not recommended for use.

Common Commands

Feature Commands

✎ **Shell**

```shell
git flow feature [command]
```

- **start**: Creates a new feature branch.
- **finish**: Merges and deletes the feature branch.
- **publish**: Pushes the feature branch to the remote repository.
- **pull**: Fetches a remote feature branch.

Release Commands

✏️ **Shell**

```
git flow release [command]
```

- **start**: Creates a new release branch.
- **finish**: Merges, tags, and deletes the release branch.
- **publish**: Pushes the release branch to the remote repository.

Hotfix Commands

✏️ **Shell**

```
git flow hotfix [command]
```

- **start**: Creates a hotfix branch.
- **finish**: Merges, tags, and deletes the hotfix branch.

Support Commands

✏️ **Shell**

```
git flow support [command]
```

- **start**: Creates a support branch for maintaining older versions.

Bugfix Commands

✏️ **Shell**

```
git flow bugfix [command]
```

- **start**: Creates a bugfix branch.
- **finish**: Merges and deletes the bugfix branch.

Notes

- Only the most important commands are covered here.
- If you'd like to supply translations, contributions are welcome!