Convert a site installed via FTP to now use Git

- With FTP, you upload / edit replace software files manually. If you edit a file, and later upgrade your site, you will lose your customization.
- With Git, each software file is tracked, including any edits to the files. And thus, your changes are merged into the update.

Context: you are taking over a Tiki site that was installed by FTP, and you now want to use Git for easy future upgrades. You may also have some modified files, and via FTP, it's time-consuming to manage this.

Example

In this example we will turn an FTP tiki 20.x instance (which is tagged as such in Git) managed by Git.

We assume you are familiar with Get code.

- **website**: http://tikiconvert.docker/
- **document root**: /var/www/virtual/tikiconvert.docker/html

First backup your things

1. Create a folder to place your backups
   
   ```
   cd /var/www/virtual/tikiconvert.docker/
   mkdir backup
   ```

2. Backup your database
   
   ```
   eval $(sed -n '/\$/s/\$/p\/html/db/local.php)
   mysqldump -u$user_tiki \n   -p$pass_tiki \n   -h$host_tiki $dbs_tiki \n   | gzip > backup/$dbversion_tiki.sql.gz
   ```

3. Backup your current folder
   
   ```
   tar -czvf backup/html.tar.gz html
   ```

Checkout the same version as your current Tiki install

```
checkout Tiki 20.x to folder newinstall
git clone git@gitlab.com:tikiwiki/tiki.git -b 20.x newinstall
```
To reduce disk space, you can add the following parameter to the command above (before newinstall):

```
--depth 1 --no-single-branch
```

If you have a specific version like for example 20.0

```
checkout Tiki 20.0 tag
git checkout tags/20.0
```

If you're experiencing issues in git clone, you can use https://gitlab.com/tikiwiki/tiki.git instead of git@gitlab.com:tikiwiki/tiki.git.

List of all branches
List of all tags

**Copy your old site over the fresh Git checkout**

Now copy your old site over the new install, if there are any modified or new files, they will override the ones of your checkout (the working copy but not the reference files in .git/ folders).

```
cp -a html/ newinstall/
```

**Check for differences**

```
git status newinstall
git diff newinstall
```

If Git shows modified files, you should solve that before continuing.
If you are experiencing problems with files permissions you can disable git from checking permissions.

```
git config core.fileMode false
```

**Launch new instance**

```
cd /var/www/virtual/tikiconvert.docker
mv html old && mv newinstall html
```

**Test and remove old folder**

Browse your website and check features you use are ok, after that remove old folder, but stick with your backup!
rm -Rf old