# What is git?

Language and Platform independent: SOAP web services can be written in any programming language and executed in any platform.

# Basic git commands

## **Tell Git who you are**

1. Configure the author name and email address to be used with your commits.

* git config --global user.name "Sam Smith"
* git config --global user.email [sam@example.com](mailto:sam@example.com)

## **Create a new local repository**

1. Create a new local repository

* git init

## **Check out a repository**

1. Create a working copy of a local repository:

* git clone /path/to/repository

1. Add one or more files to staging (index):

* git add <filename>
* git add \*

## **Commit**

1. Commit changes to head (but not yet to the remote repository):

* git commit -m "Commit message"

1. Commit any files you've added with git add, and also commit any files you've changed since then:

* git commit -a

## **Push**

1. Send changes to the master branch of your remote repository:

* git push origin master

## **Status**

1. List the files you've changed and those you still need to add or commit:

* git status

## **Connect to a remote repository**

1. If you haven't connected your local repository to a remote server, add the server to be able to push to it:

* git remote add origin <server>

1. List all currently configured remote repositories:

* git remote -v

## **Branches**

1. Create a new branch and switch to it:

* git checkout -b <branchname>

1. Switch from one branch to another:

* git checkout <branchname>

1. List all the branches in your repo, and also tell you what branch you're currently in:

* git branch

1. Delete the feature branch:

* git branch -d <branchname>

1. Push the branch to your remote repository, so others can use it:

* git push origin <branchname>

1. Push all branches to your remote repository:

* git push --all origin

1. Delete a branch on your remote repository:

* git push origin :<branchname>

## **Update from the remote repository**

1. Fetch and merge changes on the remote server to your working directory:

* git pull

1. To merge a different branch into your active branch:

* git merge <branchname>

1. View all the merge conflicts:

* git diff

1. View the conflicts against the base file:

* git diff --base <filename>

1. Preview changes, before merging:

* git diff <sourcebranch> <targetbranch>

1. After you have manually resolved any conflicts, you mark the changed file:

* git add <filename>

## **Tags**

1. You can use tagging to mark a significant changeset, such as a release:

* git tag 1.0.0 <commitID>

1. CommitId is the leading characters of the changeset ID, up to 10, but must be unique. Get the ID using:

* git log

1. Push all tags to remote repository:

* git push --tags origin

## **Undo local changes**

1. If you mess up, you can replace the changes in your working tree with the last content in head:Changes already added to the index, as well as new files, will be kept.

* git checkout -- <filename>

1. Instead, to drop all your local changes and commits, fetch the latest history from the server and point your local master branch at it, do this:

* git fetch origin
* git reset --hard origin/master

## **Search**

1. Search the working directory for foo():

* git grep "foo()"