

Git Fundamentals

Audience Course GIT Fundamentals

The course GIT Fundamentals is intended for anyone who wants to learn how to use Git as a version control system.

Prerequisites GIT Fundamentals

No specific prior knowledge is required to participate in the course GIT Fundamentals. Experience with version management or software development is beneficial to good understanding.

Realization Training GIT Fundamentals

The theory is discussed on the basis of presentation slides and is interspersed with exercises. Demo projects clarify the concepts discussed. The course times are from 9.30 am to 16.30 pm.

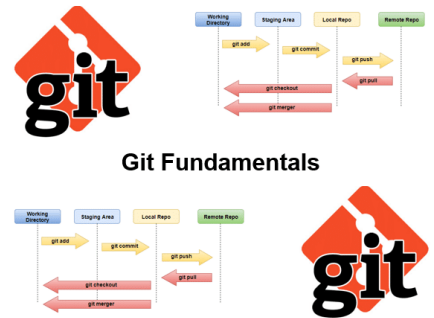
Certification GIT Fundamentals

After successful completion of the course, participants receive an official GIT Fundamentals certificate.

Duration: 1 day

Price: € 699

[Open Schedule](#)



Content Course Git Fundamentals

In the course Git Fundamentals participants learn to use the distributed version control system Git for version control of applications. Git was originally developed by Linus Torvalds for the management of the Linux Kernel but has since grown into a popular and widespread version control system. The course covers both Git from the command line and Git in the GUI SourceTree.

Git Intro

The course Git Fundamentals starts with an explanation of how Git works, which is fundamentally different from the way traditional version control systems such as Subversion and CVS work. Git does not use a central repository where you check files in and out. With Git on the other hand, every developer gets access to all previous versions in compressed form.

Git Commands

Next the Git Configuration options such as system, user, and repository configuration are discussed. After a treatment of fundamental Git concepts such as working directory and staging area, the different types of repositories such as local repositories, remote repositories and bare repositories are explained. Various basic commands in Git are also covered, such as add, commit, checkout, status, revert and reset. Also attention is paid how to reference commits by hash or by giving them a specific tag with git tag.

Git Branching

Part of the course program as well is splitting off branches for testing features with git branch. The meaning of detached HEADS is discussed here. Saving changes for later with git stash is also covered.

Git Merging

And also merging these branches again with git merge is discussed. Conflicts may have arisen during the merge of branches and the options for conflict resolution are treated.

Git Remote

And also working with remote repositories and the commands used for that such as git remote, git fetch, git pull and git push are treated.

Tooling

The course concludes with an overview of the tools available for Git version management. Then attention is also paid to GitHub, also known as Facebook for developers.

Modules Course Git Fundamentals

Module 1 : Git Intro	Module 2 : Git Commands	Module 3 : Branching
What is Git? Version Control Systems Git Installation Git Configuration Git Workflow Working Directory Staging Area Repositories Git Tooling Git Bash Git Gui Shell Integration	git init git add git status git diff git commit git log git checkout git tag git revert git reset git rm Other Commands	Branching Workflows Creating Branches git branch Switching Branches HEAD Branch Management Branch Pointer Master Branch Feature Branch Remote Branches git stash Amending Commits
Module 4 : Merging	Module 5 : Remotes	Module 6 : Tooling
git merge Merge Strategies Merge Conflicts Fork Maintenance Conflict Resolution Conflict Visualization git mergetool Resetting Master Branch Detached Heads git rebase Rebasing versus Merging	Working with Remotes Bare repositories git clone Sharing and Updating git remote Setting Origin git fetch git pull Deleting Remote Branches git push Push Errors	Git Repositories Public Repositories Securing with SSH Github BitBucket Gitlab Git in Open Source Gui Clients Smart Git Github Desktop SourceTree