

GIT for Developers

Audience GIT for Developers Course

The course GIT for Developers is intended for developers who want to use Git and GitHub for distributed version control.

Prerequisites Course GIT for Developers

To participate in the course GIT for Developers knowledge of and experience with software development in a modern language such as Java, C++, C# or PHP is required.

Realization Training GIT for Developers

The theory is discussed on the basis of the presentation slides and is interspersed with exercises. Demo projects are used to clarify the concepts. The course material is in English.

Certification GIT for Developers

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

Duration: 2 days

Price: € 1299

[Open Schedule](#)



GIT for Developers



Content Course GIT for Developers

In the course Git for developers participants learn to understand the concepts of distributed version control and to begin with the GitHub suite of tools effectively.

Attention is paid to repositories that can be located both local and remote, how versions can be committed and possibly undone. It is further treated what Git branching is and which strategies you can use for branching and merging.

The control of Git from the command line is also discussed. Furthermore attention is paid to typical GitHub workflows, to undoing errors and history manipulation in a distributed environment.

Finally some of the available tools are treated like Revision Selection, Interactive Staging, Rewriting History and Debugging with bisect.

Modules Course GIT for Developers

Module 1 : Git Intro	Module 2 : Git Branching	Module 3 : Git on the Server
What is Git? Git and DVCS concepts GitHub platform Repositories Desktop repositories Recording Changes to the Repository Viewing the Commit History Undoing Things Tagging Document versioning Configuration and customization	Branching and Merging Branche Management Branching Workflows Remote Branches Rebasing Branching strategies Merge strategies Conflict resolution Multiple remotes Fork maintenance Temporary branching solutions	The Protocols Getting Git on a Server Generating SSH Public Key Setting up the Server Public Access Hosted Git (GitHub) Local and remote repositories Distributed version control Working with Remotes Remote repository interaction Repository integration
Module 4 : Distributed Git	Module 5 : Git Tools	
Distributed Workflows Contributing to a Project Maintaining a Project Synchronization Collaboration patterns Collaboration workflows Project management Git Configuration Attributes and Hooks Local and remote synchronization Shortcuts and efficiencies	Revision Selection Interactive Staging History reordering History editing Rewriting History Debugging with bisect Submodules Subtree Merging Accidental commits Ignore patterns uses Aliases beyond-the-basics	