

GIT voor Developers

Doelgroep Cursus GIT voor Developers

De cursus GIT voor Developers is bestemd voor developers die Git en GitHub willen gebruiken voor gedistribueerde versie controle.

Voorkennis GIT

Om aan de cursus GIT voor Developers te kunnen deelnemen is kennis van en ervaring met software development in een moderne taal zoals Java, C++, C# of PHP vereist.

Uitvoering Training GIT voor Developers

De theorie wordt besproken aan de hand van presentatie slides en wordt afgewisseld met oefeningen. Demo projecten in dienen te verheldering van de behandelde concepten. De cursustijden zijn van 9.30 tot 16.30.

Certificering GIT voor Developers

De deelnemers krijgen na het goed doorlopen van de cursus een officieel certificaat GIT voor Developers.

Duur: 2 dagen

Prijs: € 1499

[Open Rooster](#)



Inhoud Cursus GIT voor Developers

In de cursus Git voor developers leren de deelnemers de concepten van gedistribueerde versie controle te begrijpen en effectief te beginnen met het gebruik van de GitHub suite van tools.

Aandacht wordt besteed aan Git repositories die zich zowel lokaal als remote kunnen bevinden, hoe versies kunnen worden gecommitt en mogelijk ook weer teruggedraaid.

Verder wordt behandeld wat Git branching is en welke branching en merge strategieen je kunt gebruiken. Ook de aansturing via de command line komt aan de orde.

Verder wordt aandacht besteed aan typische GitHub workflows, het ongedaan maken van fouten, history manipulatie in een gedistribueerde werkomgeving.

Tenslotte worden een aantal van de beschikbare tools behandeld zoals Revision Selection, Interactive Staging, Rewriting History en Debugging with Git (bisect).

Modules Cursus GIT voor 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	