

KAÏNA-COM TRAINING CATALOGUE

GIT

To learn git, its internals and how got works behind the scenes.



Nos locaux
KAÏNA-COM France
LE CARRÉ HAUSSMANN II
6 Allée de la Connaissance
77 127 Lieusaint



Contact
+33(0)9 50 20 91 64



E-mail
info@kaina-com.fr



Site Internet
www.kaina-com.fr

KSEN002 – GIT

Reference KSEN002

Experience

- Beginner
- Intermediate
- Advanced

Duration Training Program:

- 2 days

Training Method

- I: i-learning, individual training (web-based training)
- V: v-learning, virtual class

C: c-learning, classroom training

KAÏNA-COM

LE CARRÉ HAUSSMANN II,
6 Allée de la Connaissance
77127 Lieusaint - France

Price 1.390,50 € HT

Prerequisite Knowledge in VCS (Version control system)

Audience This course is for anyone who have been using any kind of VCS (Version control system) before. In this course the user will get to know git, its internals, how to use efficiently on a daily basis with focus on what are branches and what is the recommended way to use them

Continued on next page



KSEN002 – GIT, Continued

Objective

The main objective of this course is to learn git, its internals and how git works behind the scenes. On one hand Git is the most popular SCM tool but on the other hand git is the most complicated one of them all so users need to get familiar with git and its abilities and internals

Secondary Goals User will learn

How to configure git (configuration, hooks, aliases, gitconfig)

What are the key features of git (DVCS, 3-states)

What are branches and how to use them efficiently

How does merges work (ff, no-ff, rebase)

git flow and why it is recommended as the daily workflow

hooks, how to create hooks and why/when to use them

git "servers" and SourceTree

How to improve their productivity and code quality (pull request)

Tips & Tricks and beyond

Continued on next page



KSEN002 – GIT, Continued

Course Contents

Course Contents :

Table 1: KSEN002 - Course Contents

Chapter	Description
Introduction	<ul style="list-style-type: none"> • History <ul style="list-style-type: none"> – The history of GIT, SCSS, RCS • Key Features <ul style="list-style-type: none"> – DVCS, 3-states, Storage, Heuristics • 3- states <ul style="list-style-type: none"> – What is 3-states and why do we need it. – What can we do with the 3-states – Smudge – Clean • Configuration <ul style="list-style-type: none"> – Aliases, CRLF, .gitconfig, .gitkeep, .gitignore • Commit <ul style="list-style-type: none"> – What is GIT commit, how does GIT store snapshots
Commands	<ul style="list-style-type: none"> • The basic and most common GIT commands <ul style="list-style-type: none"> – init / clone – add / rm – commit – status – checkout – log

Continued on next page



KSEN002 – GIT, Continued

Course Contents, continued

Chapter	Description
Branches	<ul style="list-style-type: none">• What are branches• Branches command<ul style="list-style-type: none">– checkout– fetch– branch– merge– pull / push• SourceTree as a GIT Gui• Practice<ul style="list-style-type: none">– Single branch– Multiple branches– Single remote branch– Multiple remote branches– Multiple branches and users
Merges	<ul style="list-style-type: none">• What is GIT merge• How does GIT merge branches<ul style="list-style-type: none">– ff– no-ff– rebase– merge-commit• How to resolve conflicts
GIT Flow	<ul style="list-style-type: none">• What is GIT flow• Deep understanding of the GIT flow model• How do we scale it• What are the different branches and dots

Continued on next page



KSEN002 – GIT, Continued

Course Contents, continued

Chapter	Description
GIT in the Devops methodology	<ul style="list-style-type: none">• Automatic hooks with Jira, Jenkins, SCA and testing tools• Promotions• Artifacts management
The End	<ul style="list-style-type: none">• Summary• Q&A• Evaluation

