

# KAÏNA-COM

## CATALOGUE DE FORMATION

### **GIT**

---

**Apprendre git, ses composants internes et comment got fonctionne.**



## KSEN002 – GIT

---

**Référence** KSEN002

---

**Niveau**

- Débutant
- Intermédiaire
- Expert

---

**Nombre de jours** Programme de formation :

- 8 heures (4 heures/jour)

---

**Lieu de la formation**

- I: e-learning, Formation individuelle (Formation en ligne)
- V: v-learning, classe virtuelle
- C: c-learning, cours présentiel

---

### **KAÏNA-COM**

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

---

**Prérequis** Connaissance de VCS (Version control system)  
Un niveau d'anglais business moyen est requise car la formation sera dispensée en anglais.

---

**Public** Ce cours s'adresse à tous ceux qui ont déjà utilisé n'importe quel type de VCS pour (Version control system), logiciel de gestion de versions. Dans ce cours, l'utilisateur apprendra à connaître Git, ses composants internes, comment l'utiliser efficacement au quotidien en se concentrant sur ce que sont les branches et quelle est la façon recommandée de les utiliser

---

*Ce sujet continue à la page suivante*



## **KSEN002** – GIT, Suite

---

### **Objectifs**

Git est un logiciel de gestion de versions décentralisées.  
L'objectif principal de ce cours est d'apprendre Git, ses composants internes et comment cela fonctionne. D'une part, Git est l'outil SCM le plus populaire, mais d'autre part, Git est le plus compliqué de tous, donc les utilisateurs doivent se familiariser avec Git et ses capacités.

Objectifs secondaires L'utilisateur apprendra  
Comment configurer Git (configuration, hooks, alias, gitconfig)  
Quelles sont les fonctionnalités clés de git (DVCS, 3-states)  
Que sont les branches et comment les utiliser efficacement  
Comment fonctionne la fusion (ff, no-ff, rebase)  
Flux Git et pourquoi il est recommandé come flux de travail quotidien  
« hooks », comment créer des « hooks » et pourquoi / quand les utiliser  
git «serveurs» et SourecTree  
Comment améliorer la productivité et la qualité du code  
Trucs et astuces et plus

---

*Ce sujet continue à la page suivante*



## KSEN002 – GIT, Suite

**Contenu du cours**      Contenu du cours :

**Table 1: KSEN002 - Contenu du cours**

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

*Ce sujet continue à la page suivante*



## KSEN002 – GIT, Suite

### Contenu du cours, Suite

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

*Ce sujet continue à la page suivante*



## KSEN002 – GIT, Suite

---

### Contenu du cours, Suite

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

---

