NIRG@CODEWIZARD.CO.IL / 054 8122310

Git Basic

	Level	Duration
	Introduction / Intermediate	2 days

Course Description

- 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

Audience and prerequisites

- This course is for anyone who has been using any kind of VCS (Version control system) before.
- `Developers`
- `DevOps`

Main Goals

- 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 do merges work (ff, no-ff, rebase)
- GitFlow and why it is recommended as the daily workflow
- hooks, how to create hooks, and why/when to use them
- git "servers" and clients
- How to improve their productivity and code quality (pull request)
- Tips & Tricks and Beyond

Session	Content
Introduction	
	◆ History of Git
	◆ The history of git SCSS, RCS
	◆ **Key Features**
	◆ DVCS
	◆ 3-states
	◆ Storage
	◆ Heuristics
	◆ **3- states**
	◆ What is 3-states and why do we need it?
	◆ What can we do with the 3-states
	◆ **Smudge - Clean**
	◆ What are they and how can we use them
	◆ **Configuration**
	◆ Configure username / email
	◆ aliases
	◆ CRLF
	◆ .gitconfig
	◆ .gitkeep
	◆ .gitignore
	◆ **Commit**
	◆ What is git commit?
	◆ How does git store snapshots
	◆ What is the difference between git to other CVS tools?
Commands/CLI	
	◆ The basic and most common git commands
	◆ init
	◆ clone
	◆ add
	◆ rm
	◆ commit

	◆ status
	◆ checkout
	◆ log
Branches	
	◆ What are branches
	◆ Branches command
	◆ checkout
	◆ fetch
	◆ branch
	◆ merge
	◆ pull
	◆ push
	◆ switch
	◆ worktree
Hands-on	
	◆ In class hands on working with branches
	◆ **Single** branch
	◆ **Multiple** branches
	◆ **Single remote** branch
	◆ **Multiple remote** branches
	◆ Multiple branches and multiple users
Merge / Rebase	
	◆ What is git merge
	◆ How does git merge branches
	◆ What are the different git merge strategies
	◆ `ff`
	◆ `no-ff`
	◆ `rebase`
	◆ **merge-commit**
GitFlow	
	◆ What is GitFlow
	◆ Deep understanding of the GitFlow model

	◆ Why should we use it
	◆ How can this model improve out productivity
	◆ What are the different branches in the model
	◆ How can we use the GitFlow scripts for automating the flow
	◆ Best practice for GitFlow
Advanced Topics	
	◆ Quick overview of advanced git features (Deep walkthrough is in part of the **advanced course**)
	◆ assume-unchanged
	◆ auto-completion / autocorrect
	◆ bisect
	◆ cherry-pick
	◆ smudge / clean
	◆ fake merge / detailed merge
	◆ git LFS
	◆ hooks
	◆ notes
	◆ reflog
	◆ rerere
	◆ squash
	◆ stash
	◆ submodule / subtree
	◆ tags
	◆ worktree
Pull Request	
	◆ What is pull request
	◆ Why should we always use it
	◆ What does the pull request include
	◆ CI/CD

Back to courses list