CATALOGO DEI CORSI

Clean Code Kata

> “ Any fool can write code that a computer can understand. Good programmers write code that humans can understand. ”
>
> — *Martin Fowler*

Il Clean Code è la pratica di scrivere codice leggibile, semplice, elegante, e quindi facilmente manutenibile. Un codice pulito non si ritorce contro lo sviluppatore, anzi aiuta a comprendere meglio il dominio di interesse e la soluzione da adottare. Purtroppo molto spesso è più facile a dirsi che a farsi.

"Kata" è un termine giapponese che si riferisce a esercizi da ripetere costantemente per automigliorarsi.

Questo corso comincia con la descrizione di tutte quelle pratiche raccolte da Robert C. Martin nel libro chiamato "Clean Code" e prosegue con tre Kata, esercizi di difficoltà crescente in cui si mettono in pratica le conoscenze acquisite, condite con concetti di ingegneria del software e applicando le tecniche di refactoring e test-driven development.

Prerequisiti

Il corso è rivolto a sviluppatori software che conoscono almeno un linguaggio object-oriented.

Durata

La durata prevista del corso è di 16 ore.

Materiale

Sono disponibili le slide utilizzate per la parte teorica e il codice che implementa la soluzione agli esercizi.

Programma

1. Teoria
    1. Il Clean Code
    2. I Kata
2. Pratica
    1. Analisi e refactoring di codice non pulito
    2. Analisi comparata sulla flessibilità di quattro soluzioni differenti per risolvere un problema
    3. Soluzione di un problema complesso attraverso l'implementazione di moduli semplici e riutilizzabili

Inglorious Coderz di Mistretta Matteo Antony
Via Val Della Torre, 39
10149 Torino (TO)

Hangouts: antonymistretta
Skype: antony.mistretta

antony.mistretta@gmail.com
www.ingloriouscoderz.it

Test-Driven Development / Behaviour-Driven Development

> “ Code never lies, comments sometimes do. ”
>
> — *Ron Jeffries*

Una delle pratiche più controverse promossa dalla metodologia Agile chiamata Extreme Programming è proprio il Test-Driven Development: il Cliente mi paga per scrivere codice e lo vuole pronto per domani, e io devo sprecare tempo a scrivere codice di test che non mi è stato richiesto né mi verrà pagato? La risposta è sì. Questo perché il Test-Driven Development (abbreviato TDD), per quanto all'inizio rallenti lo sviluppo, a regime aiuta a mantenere il codice stabile, semplice e addirittura ben documentato, con grande gioia sia dello sviluppatore sia del Cliente stesso.

In questo corso si espone il TDD nel contesto di un processo Agile, mostrandone i benefici a breve e lungo termine, si illustra il Behaviour-Driven Development (BDD) come evoluzione del TDD, e si discute sull'applicabilità di tali concetti nel contesto aziendale della classe. Opzionalmente vengono svolti esercizi pratici di TDD/BDD in un linguaggio di programmazione a scelta.

Prerequisiti

Chiunque lavori in ambito IT può beneficiare della parte teorica: project manager, responsabili IT, programmatori. Per la parte pratica è necessaria qualche conoscenza di programmazione.

Durata

La durata prevista è di 4 ore per la parte teorica, più altre 4 ore per la parte pratica.

Materiale

Sono disponibili le slide utilizzate durante il corso e il codice sorgente relativo alla parte pratica.

Programma

1. Che cos'è il TDD
2. I vantaggi del TDD
3. Che cos'è il BDD
4. Esercizi di TDD/BDD

Inglorious Coderz di Mistretta Matteo Antony
Via Val Della Torre, 39
10149 Torino (TO)

Hangouts: antonymistretta
Skype: antony.mistretta

antony.mistretta@gmail.com
www.ingloriouscoderz.it

Extreme Programming

> “ I'm not a great programmer; I'm just a good programmer with great habits. ”
>
> — *Kent Beck*

Tra le metodologie Agile, Extreme Programming è sicuramente la più travisata. Eppure al giorno d'oggi molte delle sue pratiche, come il Test-Driven Development o la Continuous Integration, sono all'ordine del giorno in un'azienda IT moderna. Extreme Programming (abbreviato XP) è complementare alle altre metodologie Agile: Scrum ad esempio si occupa di project management ad alto livello, mentre XP si applica alla produzione di codice e al suo rilascio in produzione.

In questo corso, di stampo teorico, si descrivono le attività coinvolte in XP, i valori alla base e le pratiche che ne derivano, e si discute sull'applicabilità di tali concetti nel contesto aziendale della classe.

Prerequisiti

L'unico prerequisito è una conoscenza di base su cosa significhi sviluppare software, per cui sono ben accetti programmatori ma anche capi progetto e team leader.

Durata

La durata prevista è di 16 ore.

Materiale

Sono disponibili le slide utilizzate durante il corso.

Programma

1. Attività
    1. Coding
    2. Testing
    3. Listening
    4. Designing
2. Valori
    1. Communication
    2. Simplicity
    3. Feedback
    4. Courage
    5. Respect
3. Pratiche
    1. Pair Programming
    2. Planning Game
    3. Test-Driven Development
    4. Whole Team
    5. Continuous Integration
    6. Design Improvement
    7. Small Releases
    8. Coding Standard
    9. Collective Code Ownership
    10. Simple Design
    11. System Metaphor
    12. Sustainable Pace

Inglorious Coderz di Mistretta Matteo Antony
Via Val Della Torre, 39
10149 Torino (TO)

Hangouts: antonymistretta
Skype: antony.mistretta

antony.mistretta@gmail.com
www.ingloriouscoderz.it

Web 2.0

> “ The world is changed. I feel it in the water. I feel it in the earth. I smell it in the air. ”
>
> *— Galadriel, Lord Of The Rings*

Il mondo sta cambiando. E con esso il web, il quale evolve troppo velocemente per rimanere ancorati a software legacy e mentalità antiquate.

In questo corso di stampo teorico si mostra una panoramica delle tecnologie e delle metodologie più recenti che si sono affermate nel panorama tecnologico internazionale, dalle architetture di rete ai più innovativi framework alle soluzioni cloud. La lezione è accompagnata da diverse dimostrazioni pratiche.

Prerequisiti

Il corso può essere seguito senza problemi da responsabili tecnici, project manager, sistemisti e sviluppatori.

Durata

La durata prevista è di 8 ore.

Materiale

Sono disponibili le slide utilizzate durante il corso, nelle quali sono presenti numerosi link di approfondimento.

Programma

1. Architetture
    1. Evoluzione delle architetture software
    2. Application Server vs. Microservices
    3. REST
    4. Protocolli basati su REST
    5. GraphQL
2. Back End
    1. DBMS relazionali vs. NoSQL
    2. Web Framework: Evoluzione vs. Rivoluzione
    3. Server: Multithread vs. Event-Driven
3. Front End
    1. Evoluzione del web
    2. Framework JavaScript
    3. Framework CSS
    4. Evoluzione del mobile
    5. App native
    6. App compile-to-native
    7. App ibride
    8. Progressive Web Apps
4. Agile
    1. Introduzione a Extreme Programming
    2. Small Releases
    3. Test-Driven Development
    4. Refactoring
    5. Coding Standard
    6. Continuous Integration
5. Deploy
    1. Git
    2. Software di CI/CDE/CD
    3. Virtual Machine
    4. Container
    5. Orchestrator
    6. Cloud/PaaS/Serverless

Inglorious Coderz di Mistretta Matteo Antony
Via Val Della Torre, 39
10149 Torino (TO)

Hangouts: antonymistretta
Skype: antony.mistretta

antony.mistretta@gmail.com
www.ingloriouscoderz.it

Design Patterns

"Reinventare la ruota" è un'espressione idiomatica anglosassone che si utilizza quando una soluzione tecnica generalmente accettata viene ignorata a favore di soluzioni ricreate localmente. Molti problemi in ambito software si possono invece ricondurre a una radice comune che si può risolvere con una specifica soluzione. I design pattern sono soluzioni standard a problemi ricorrenti, pertanto sono un valido aiuto per trovare una soluzione, comunicarla in modo efficace, e soprattutto evitare di reinventare la ruota.

In questo corso si definiscono i design pattern in generale e si mostra una panoramica dei design pattern più noti, confrontandoli con il contesto aziendale della classe. Si espongono anche alcuni anti-pattern, utili a evitare errori comuni in fase di progettazione del software.

Prerequisiti

Nonostante non sia prevista la scrittura di codice, sono comunque necessarie delle solide basi di programmazione orientata agli oggetti.

Durata

La durata prevista è di 8 ore.

Materiale

Sono disponibili le slide utilizzate durante il corso.

Programma

1. Il code smell
2. Che cos'è un design pattern
3. I GRASP
    1. Controller
    2. Creator
    3. High Coesion
    4. Information Expert
    5. Low Coupling
    6. Polymorphism
    7. Protected Variations
    8. Pure Fabrication
    9. Indirection
4. I pattern della Gang Of Four
    1. Abstract Factory/Builder/Factory Method
    2. Prototype/Flyweight
    3. Singleton
    4. Adapter
    5. Composite
    6. Decorator/Proxy
    7. Command
    8. Mediator
    9. Memento
    10. Observer/Pub-Sub
    11. State/Strategy
5. I principi SOLID
    1. Single Responsibility
    2. Open/Closed
    3. Liskov Substitution
    4. Interface Segregation
    5. Dependency Inversion
6. Anti-pattern
    1. Anemic Domain Model
    2. God Object
    3. Spaghetti Code
    4. Premature Optimization

Inglorious Coderz di Mistretta Matteo Antony
Via Val Della Torre, 39
10149 Torino (TO)

Hangouts: antonymistretta
Skype: antony.mistretta

antony.mistretta@gmail.com
www.ingloriouscoderz.it

Java Web Frameworks

Il linguaggio Java è forse il principale motivo per cui la programmazione orientata agli oggetti è esplosa negli anni '90. Da allora il linguaggio si è evoluto, così come le tecnologie basate su di esso, purtroppo però non senza errori: per lungo tempo tecnologie come la piattaforma J2EE hanno introdotto e persino suggerito soluzioni fin troppo complesse e pesanti. Per contrastare questo *over-engineering* negli anni 2000 framework come Struts, Spring e Hibernate hanno utilizzato e incoraggiato nel modo corretto dei design pattern fondamentali alla buona progettazione, consentendo così ancora oggi agli sviluppatori di realizzare soluzioni enterprise semplicemente concentrandosi sulla modellazione del dominio di interesse.

In questo corso, molto pratico ma supportato da diversi concetti di ingegneria del software, si parte dalla creazione di una semplice web application con funzionalità CRUD realizzata con servlet e JSP per poi analizzare progressivamente le difficoltà che comporta scalare l'applicazione. In questo modo si introducono uno a uno i framework Java e i princìpi teorici alla base di essi, fino ad arrivare a una web application completa ma anche semplice, robusta e flessibile.

Prerequisiti

Il corso è inteso per sviluppatori che abbiano una buona conoscenza del linguaggio Java e della programmazione orientata agli oggetti.

Durata

La durata prevista è di 48 ore.

Materiale

Sono disponibili le slide utilizzate durante il corso, che includono link di approfondimento, più il codice sorgente relativo alle esercitazioni.

Programma

1. Maven
    1. Utilizzo per applicazioni da linea di comando
    2. Utilizzo per applicazioni web
    3. Risoluzione delle dipendenze
    4. Automazione del processo di compilazione e deploy
2. Servlet e JSP
    1. Creazione di una web app con funzionalità CRUD
    2. Il design pattern Decorator/Proxy
    3. Filter
3. Struts 2
    1. Il design pattern Observer
    2. Il pattern architetturale MVC
    3. Migrazione della web app da Servlet e JSP a Struts
4. Spring
    1. Singleton come anti-pattern
    2. Il principio Inversion Of Control
    3. Il design pattern Dependency Injection
    4. Migrazione della web app a Spring
    5. Integrazione fra Spring e Struts
    6. Aspect-Oriented Programming
    7. Spring Web
    8. Migrazione della web app da Struts a Spring Web
    9. Spring MVC
    10. Creazione di API REST con Spring MVC
5. Hibernate
    1. Object-Relational Mapping
    2. Migrazione della web app a Hibernate
    3. Integrazione fra Spring e Hibernate
    4. Rappresentazione di relazioni fra tabelle
    5. Realizzazione di una web app simil-Facebook

Inglorious Coderz di Mistretta Matteo Antony
Via Val Della Torre, 39
10149 Torino (TO)

Hangouts: antonymistretta
Skype: antony.mistretta

antony.mistretta@gmail.com
www.ingloriouscoderz.it

ESNext

Uno dei linguaggi che sta avendo l'impatto maggiore sulla tecnologia odierna è JavaScript (da non confondersi con Java): da semplice linguaggio di scripting sul browser è diventato la lingua franca per realizzare server ultra-performanti, applicazioni desktop, mobile e web, e addirittura può essere impiegato su dispositivi embedded e ambienti di realtà virtuale. La sua standardizzazione da parte di ECMA International (che lo ha ribattezzato ECMAScript, o semplicemente ES), e le innovazioni portate dai big player del web quali Google, Microsoft, Facebook e Mozilla, lo hanno portato rapidamente in cima alla classifica delle tecnologie più popolari su StackOverflow.

Purtroppo per molti l'evoluzione di ES sta avvenendo un po' troppo rapidamente: il linguaggio aggiunge elementi di sintassi nuovi ogni anno, comincia ad abbracciare paradigmi di programmazione diversi, e il suo ecosistema è così fervente che si dice che non passi giorno senza che nasca un nuovo framework. Risulta quindi necessario districarsi in questo caos, selezionando accuratamente le informazioni davvero rilevanti e scartando quelle più secondarie o prossime a scomparire nella corsa alla tecnologia web perfetta.

In questo corso si dà un'occhiata alla storia di ECMAScript, ai problemi che si sono risolti, alle nuove opportunità offerte dal nuovo ecosistema e alle sfide ancora da superare. Inoltre si provano alcune delle nuove funzionalità del linguaggio attraverso un'esercitazione pratica.

Prerequisiti

Per seguire con successo questo corso sarebbe utile avere qualche base di programmazione in un qualsiasi linguaggio.

Durata

La durata prevista è di 16 ore.

Materiale

Sono disponibili le slide utilizzate durante il corso, più il codice sorgente relativo alla parte pratica.

Programma

1. L'ambiente di sviluppo JavaScript
    1. Node e NPM
    2. Express
2. Novità introdotte da ES2015
    1. Moduli sincroni e asincroni
    2. Nuova sintassi: template strings, const e let, arrow functions
    3. Destrutturazione e spread operator
    4. JS funzionale (map, reduce, currying)
    5. Fetch API
    6. Promises, generator functions, async/await
    7. JS object-oriented (classi, pattern)
    8. Babel, ESLint, Typescript, Flow, Webpack
3. Transpilazione
    1. Storia: CoffeeScript
    2. TypeScript
    3. Babel/JSX/ESNext
    4. Less/Sass/Stylus
    5. Jade/Pug
4. Automazione
    1. Storia: Grunt/Gulp/Yeoman
    2. Webpack
5. Test
    1. BDD e Jasmine/Jest
6. Framework
    1. Storia: Backbone/Knockout/AngularJS
    2. React
    3. Angular
    4. Vue.js

Inglorious Coderz di Mistretta Matteo Antony
Via Val Della Torre, 39
10149 Torino (TO)

Hangouts: antonymistretta
Skype: antony.mistretta

antony.mistretta@gmail.com
www.ingloriouscoderz.it

JavaScript Frameworks

L'evoluzione rapidissima del linguaggio JavaScript, e l'interesse dimostrato dai big player del web come Google, Microsoft e Facebook, ha portato a una pletora di framework in competizione tra loro. Fra questi i vincitori, cioè quelli che alla fine si sono affermati maggiormente, sono tre:

- Angular, apprezzato moltissimo in ambito enterprise per la sua somiglianza con altri framework utilizzati sul back end e per il fatto di essere supportato da aziende come Google e Microsoft;
- React, il quale ha una curva di apprendimento un po' più alta ma consente di ottenere più espressività con meno codice e performance ottimali; e
- Vue.js, che riesce a conciliare la semplicità d'uso di Angular, le performance di React, e l'eleganza che si vuole raggiungere un giorno con i web components.

In questo corso si prova a implementare la stessa applicazione CRUD con tutti e tre i framework, facendo un'analisi comparata dei pro e dei contro di ciascuno. Inoltre si discute su quale framework convenga maggiormente date le esigenze del Cliente e gli obiettivi che ci si prefigge.

Prerequisiti

Sono necessarie conoscenze di base di programmazione in un qualsiasi linguaggio. Inoltre è preferibile aver seguito il corso ESNext.

Durata

La durata prevista è di 24 ore.

Materiale

Sono disponibili le slide utilizzate durante il corso, più il codice sorgente relativo alla parte pratica.

Programma

1. Angular
    1. Caratteristiche del framework
    2. Realizzazione di una web app
2. React
    1. Framework o libreria?
    2. Caratteristiche del framework
    3. Realizzazione di una web app
3. Vue.js
    1. Caratteristiche del framework
    2. Realizzazione di una web app

Inglorious Coderz di Mistretta Matteo Antony
Via Val Della Torre, 39
10149 Torino (TO)

Hangouts: antonymistretta
Skype: antony.mistretta

antony.mistretta@gmail.com
www.ingloriouscoderz.it

Angular

Angular è il framework più utilizzato in ambito enterprise per la sua somiglianza con altri framwork utilizzati sul back end e perché sviluppato e sponsorizzato da Google in collaborazione con Microsoft. La sua curva di apprendimento è relativamente bassa se si conoscono i concetti della programmazione orientata agli oggetti e i pattern utilizzati in ambito back end come la Dependency Injection.

Ultimamente Angular ha incorporato alcuni concetti e tecnologie prese in prestito da altri ambiti, come Redux e RxJS. In questo caso entra in gioco la programmazione funzionale e le cose si fanno più interessanti.

In questo corso si realizza una web application da zero con funzionalità CRUD e accesso a un server esterno, illustrando man mano le caratteristiche del framework.

Prerequisiti

È necessaria la conoscenza di almeno un linguaggio orientato agli oggetti. Inoltre è preferibile aver seguito il corso ESNext.

Durata

La durata prevista è di 40 ore.

Materiale

1. Da AngularJS ad Angular
2. Architettura del framework
3. Data binding
4. Componenti
    1. Gerarchia di componenti
    2. Comunicazione fra componenti
    3. Ciclo di vita dei componenti
    4. Aggiornamento e render
5. Servizi
6. Dependency Injection
7. Integrazione con RxJS
    1. Gestione funzionale di flussi asincroni
    2. RxJS
    3. Gestione eventi Angular con Observable e Subject
8. Integrazione con Redux
    1. Le basi della programmazione funzionale
    2. I tre princìpi di Redux
    3. Action types, action creators, store.dispatch
    4. Reducers, selectors, store.subscribe
    5. Behaviour-driven development con Jest
    6. ngrx
    7. Middleware e reduxDevTools
    8. Chiamate asincrone con ngrx-effects
9. Filtri
10. Proiezioni
11. Ottimizzazione delle performance
    1. Cicli
    2. Immutabilità
    3. Debouncing
    4. Minimizzare i re-render con ChangeDetectionStrategy.OnPush

Inglorious Coderz di Mistretta Matteo Antony
Via Val Della Torre, 39
10149 Torino (TO)

Hangouts: antonymistretta
Skype: antony.mistretta

antony.mistretta@gmail.com
www.ingloriouscoderz.it

React

React sostiene di essere semplicemente una libreria per creare interfacce grafiche. In realtà questo software sviluppato e sponsorizzato da Facebook è un framework a tutti gli effetti, che consente di realizzare applicazioni anche di grandi dimensioni con performance impareggiabili, grazie alla possibilità di scrivere codice semplice e modulare.

La curva di apprendimento è un po' più alta perché entrano in gioco concetti di programmazione funzionale, ma le regole di base da imparare sono poche e tutto il resto viene logicamente da sé. Se non ci si lascia spaventare si possono imparare dei concetti che servono a comprendere a fondo qualsiasi altra tecnologia moderna. Imparare React significa imparare a programmare bene.

In questo corso si realizza una web application da zero con funzionalità CRUD e accesso a un server esterno, illustrando man mano le caratteristiche del framework.

Prerequisiti

Sono necessarie conoscenze di base di programmazione in un qualsiasi linguaggio. Inoltre è preferibile aver seguito il corso ESNext.

Durata

La durata prevista è di 40 ore.

Materiale

Sono disponibili le slide utilizzate durante il corso, più il codice sorgente relativo alla parte pratica.

Programma

1. React
    1. create-react-app
    2. JSX
    3. Class component: props, state, attributi di istanza
    4. Gerarchia di componenti e comunicazione tra componenti
    5. Ciclo di vita dei componenti e ottimizzazione delle performance con shouldComponentUpdate
    6. Componenti funzionali
    7. PureComponent/React.memo
    8. Comunicazione con il server
    9. Container component e Presentational component
    10. Form: controlled vs. uncontrolled component
    11. Pattern: Higher Order Components e Render props/Function as child component
2. React Router
    1. Sviluppo applicazioni con più schermate e creazione menù di navigazione
    2. Gestione rotte annidate
3. Inline styles vs. CSS modules
4. Recompose
    1. La funzione compose
    2. withState, withReducer, withProps, withHandlers, lifecycle
5. Redux
    1. I tre princìpi
    2. createStore, middleware e reduxDevTools
    3. Action types, action creators, store.dispatch
    4. Reducers, selectors, store.subscribe
    5. Behaviour-driven development con Jest
6. Reselect
7. React-Redux
8. Integrazione flussi asincroni con Redux
    1. Redux-thunk
    2. Redux-saga
9. Context API
10. Hooks

Inglorious Coderz di Mistretta Matteo Antony
Via Val Della Torre, 39
10149 Torino (TO)

Hangouts: antonymistretta
Skype: antony.mistretta

antony.mistretta@gmail.com
www.ingloriouscoderz.it

Vue.js

Vue.js è un framework ancora fin troppo poco conosciuto in Italia, ma all'estero sta spopolando perché riesce a conciliare la semplicità d'uso di Angular, le performance di React e l'eleganza di Polymer. Il famoso framework PHP Laravel l'ha scelto come framework di default per la parte front end, probabilmente anche perché ne condivide la curva di apprendimento lineare. Conoscere Vue.js significa imparare tutti i concetti che la nuova generazione di framework porta con sé, ma partendo con il piede giusto.

In questo corso si realizza una web application da zero con funzionalità CRUD e accesso a un server esterno, illustrando man mano le caratteristiche del framework.

Prerequisiti

Sono necessarie conoscenze di base di programmazione in un qualsiasi linguaggio. Inoltre è preferibile aver seguito il corso ESNext.

Durata

La durata prevista è di 40 ore.

Materiale

Sono disponibili le slide utilizzate durante il corso, più il codice sorgente relativo alla parte pratica.

Programma

1. Che cos'è Vue.js
2. Creazione di un progetto Vue con vue-cli
3. Rendering dichiarativo
4. Debug con Vue.js devtools
5. Istruzioni condizionali e cicli
6. Gestione dell'input utente
7. Composizione di componenti
8. Dati e metodi
9. Ciclo di vita di un componente
10. Sintassi dei template
11. Variabili computate e watcher
12. Applicare fogli di stile
13. Two-way data binding
14. Utilizzo di plugin
15. Unit testing di componenti
16. Chiamate Ajax con axios
17. Single-page application con vue-router
18. Gestione avanzata dello stato con vuex
19. Server-side rendering con Nuxt.js

Inglorious Coderz di Mistretta Matteo Antony
Via Val Della Torre, 39
10149 Torino (TO)

Hangouts: antonymistretta
Skype: antony.mistretta

antony.mistretta@gmail.com
www.ingloriouscoderz.it

Git

> “ I'm an egotistical bastard, so I name all my projects after myself. First Linux, now git. ”
>
> — *Linus Torvalds*

Git è lo strumento di versionamento del codice per eccellenza: ha soppiantato gli storici CVS e SVN, ed è stato persino promosso da Microsoft a scapito di TFS. Il suo successo è dovuto alle elevate performance (è scritto in C puro), alla natura distribuita e a una progettazione accurata che lo rende il miglior amico di ogni sviluppatore.

In questo corso si descrive la filosofia alla base di Git, i comandi più utilizzati, e si espongono le due strategie di branching più diffuse: GitFlow e il Cactus Model.

Prerequisiti

È sufficiente una conoscenza di base della linea di comando.

Durata

La durata prevista è di 8 ore.

Materiale

Sono disponibili le slide utilizzate durante il corso.

Programma

1. Teoria di Git
    1. Il controllo di versione
    2. Versionamento locale/centralizzato/distribuito
    3. Memorizzazione di istantanee
    4. Operazioni in locale
    5. Controlli di integrità
    6. Database incrementale
    7. I tre stati
2. Comandi di base
    1. init, add, commit, rm, mv
    2. status, diff, log
    3. Il file .gitignore
3. Tornare indietro
    1. commit --amend
    2. reset HEAD
    3. checkout -f
4. Repository remoto
    1. clone, remote
    2. fetch, push, pull
5. Tagging
6. Branching
    1. Che cos'è un branch in Git
    2. Spostarsi da un branch all'altro
    3. branch, merge, rebase
    4. Branch remoti
7. Workflow distribuiti
    1. Regole per scrivere commenti
    2. Fork e pull request
8. GitFlow
9. Cactus Model

Inglorious Coderz di Mistretta Matteo Antony
Via Val Della Torre, 39
10149 Torino (TO)

Hangouts: antonymistretta
Skype: antony.mistretta

antony.mistretta@gmail.com
www.ingloriouscoderz.it