

Tree school

Modulo 3

Cit

A. Gasparini



Buone regole e consigli pratici

- **Tenete il microfono mutato**
- **Se avete dubbi, domande o curiosità interrompetemi senza farvi problemi.**
- **Le pause sono concordate, ma se sentite che c'è bisogno di anticipare chiedete senza problemi**
- **Nelle slide troverete tutti i concetti della lezione ma alcune cose nuove possono uscire durante la lezione, prendere appunti potrebbe essere una buona idea.**
- **Le domande su colloqui o esperienze di lavoro sono di interesse per tutti!**
- **Se saltate gli esercizi monopattino perché li ritenete troppo facili e fate gli scooter e moto va benissimo.**



Vi presento
un amico



Stackoverflow

- **Quando avete un problema col codice, non vi ricordate come fare una cosa o per risolvere una curiosità, c'è un grande vostro amico in rete**
- **Stackoverflow è un sito che raccoglie domande e risposte di genere tecnico informatico**
- **Quando avete un problema potete chiedere lì ma al 99,999% quel vostro problema è già stato chiesto e risposto**
- **Spesso, per sapere come fare una cosa, scrivetela su google e aggiungete stackoverflow, che tanto la miglior risposta è lì.**



stackoverflow

Stackoverflow esempio

How to get the user input in Java?

Asked 9 years, 7 months ago · Active 21 days ago · Viewed 1.7m times

I attempted to create a calculator, but I can not get it to work because I don't know *how to get user input*.

327

How can I get the user input in Java?

java input

140

share edit follow flag

edited May 19 at 21:56



Dirk Lachowski

2,973 ● 4 ● 37 ● 59

asked Mar 13 '11 at 4:59



Jtvd78

3,765 ● 3 ● 16 ● 21

9 Uh, what's your question? You just posted some code and said you don't like pointers. Not understanding pointers can still come back to bite you in java if you don't understand pass by reference and pass by value. – Scott Mar 13 '11 at 5:05

4 you should try to learn java reading a book, Java How to Program, 7/e is a nice one – Marco Aviles Mar 13 '11 at 5:06

[add a comment](#)

[start a bounty](#)

28 Answers

Active

Oldest

Votes

You can use any of the following options based on the requirements.

345

Scanner class



```
import java.util.Scanner;
//...
Scanner scan = new Scanner(System.in);
String s = scan.next();
int i = scan.nextInt();
```

Quando provo a unire due pezzi di codice trovati su stackoverflow

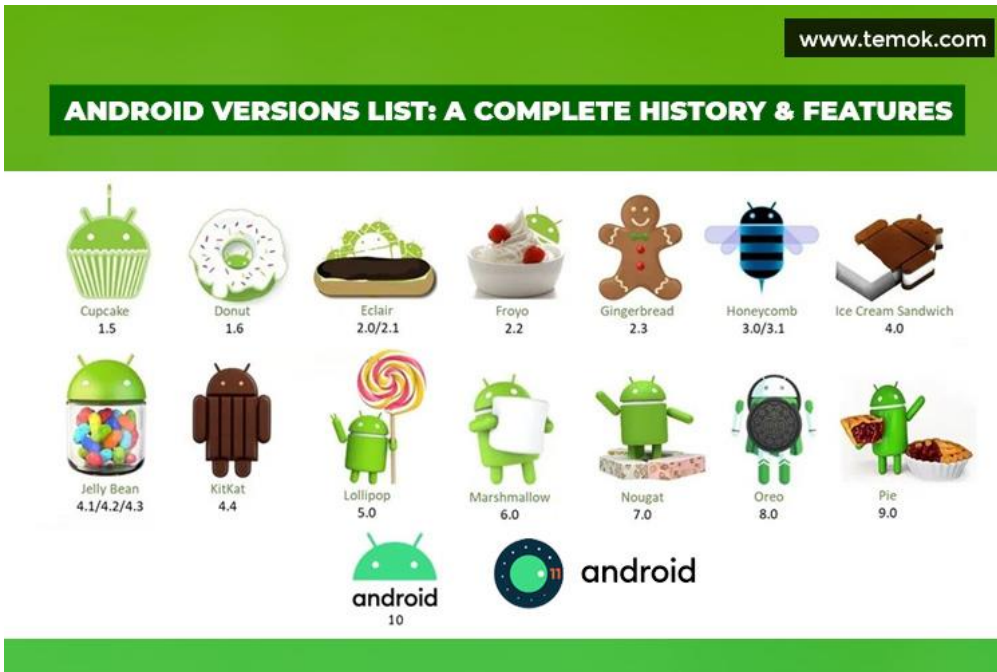


Versionamento del codice



Perché versionare il codice?

- **Nella nostra vita di tutti i giorni, ci capita spesso di leggere il numero di versione di un software.**
- **La versione di un software è utile non soltanto per capire quali sono le feature presenti ma anche per sapere come va utilizzato un determinato software, anche per noi sviluppatori.**
- **In questi casi però, parliamo di versionamento esterno e cioè quello che chi scrive il codice comunica all'esterno e spesso legato alle release.**
- **Esiste invece un versionamento interno del software che invece è molto comodo allo sviluppatore**



Perché il versionamento interno?

- **È molto comune lavorare in un team su un unico progetto.**
- **Si potrebbe pensare di passarsi il codice su chiavetta oppure di usare una cartella drive comune...**
- **Immaginate di lavorare in un team composto da 50 persone e ai problemi che potrebbero uscire fuori!**
- **Anche se si sta lavorando da soli, può succedere di voler tornare ad una vecchia versione del codice che era «stabile».**



Git



Vi presento Git!



- **Git è il software di versionamento del codice più popolare al mondo.**
- **È stato inventato dai programmatori del kernel di Linux perché non erano pienamente soddisfatti del software che stavano utilizzando per versionare il codice.**
- **Nonostante i comandi base di git siano piuttosto semplici da imparare, la guida completa per usare git è un libro disponibile gratuitamente a questo [link](#)**
- **Ma ci sono anche belle guide rapide in Italiano tipo [questa](#)**



Caratteristiche principali git

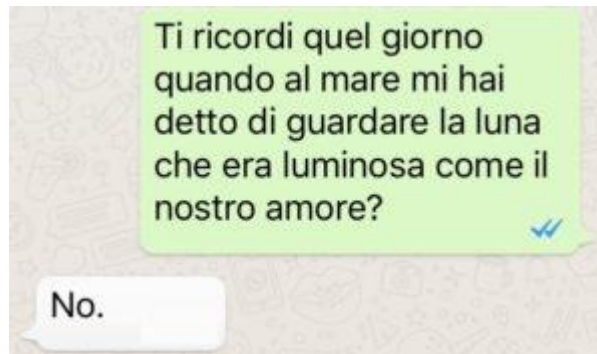
- **Permette la creazione di versioni interne del codice**
- **Gestisce la sincronizzazione del codice fra più utenti**
- **Rileva conflitti in fase di sincronizzazione e offre strumenti per gestirli**
- **Ci da la possibilità di creare versioni parallele del software (chiamate branch)**
- **Permette di tornare indietro ad una vecchia versione del codice.**



Perché vogliamo poter tornare indietro?



Perché vogliamo poter tornare indietro? 2

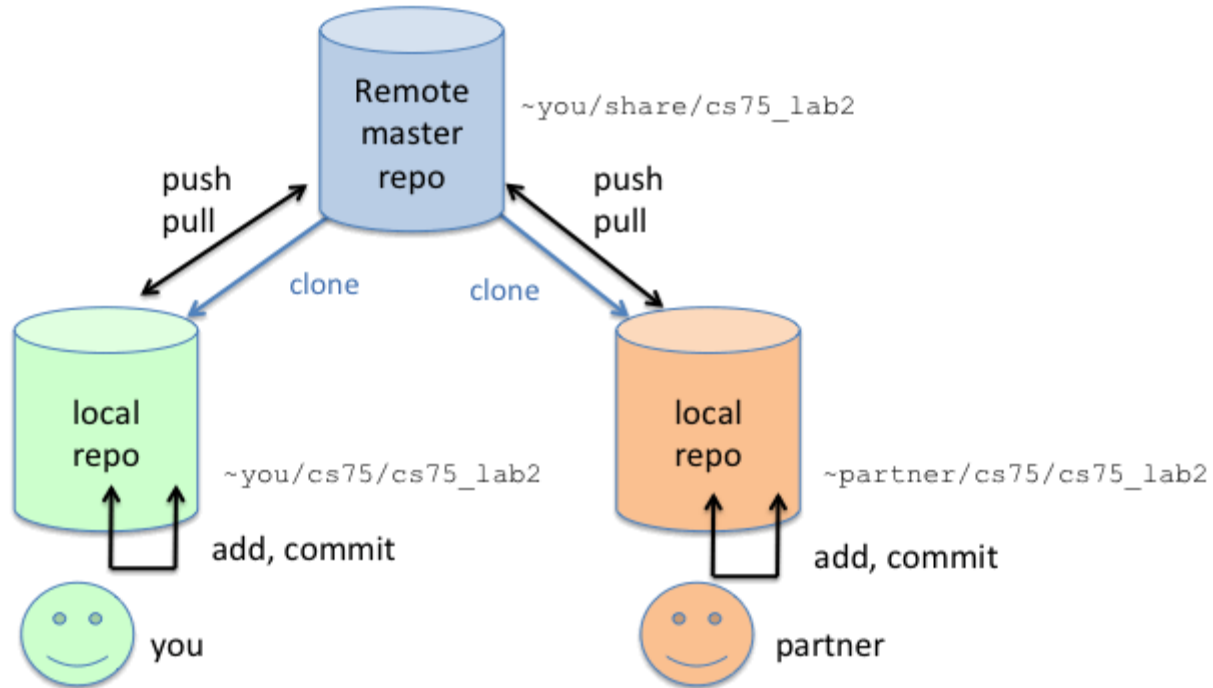


Bonus: Chi dei due vuole tornare indietro?

Glossario git

| Termine | Significato |
|-------------------|---|
| Repository | Dove risiede il codice e le sue versioni, può essere locale o online («remote»). |
| Init | Comando per inizializzare una repo |
| Pull | È il comando che aggiorna il vostro codice alla versione più recente |
| Commit | Il comando con cui create una nuova versione del codice |
| Push | Il comando con cui inviare un commit alla repository online |
| Conflict | Avviene quando due utenti modificano le stesse linee di un file, creando un conflitto, e git non sa come risolvere la questione senza intervento umano |
| Merge | L'azione di unire due diverse versioni della stessa repository |
| Branch | Una versione parallela del codice che in futuro può essere reintegrata in quella principale |
| Fork | Inizia una nuova repository partendo da un'altra, a differenza del branch questa non sarà una versione parallela e non farà parte della repository ufficiale |

Repository



Git commit push



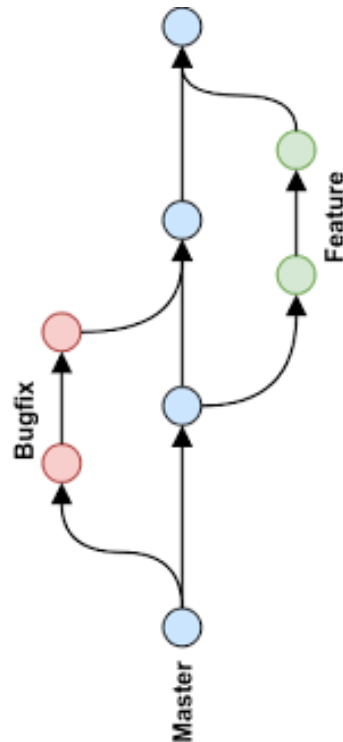
Cos'è un branch?



Sicuramente non questo che è un brunch!

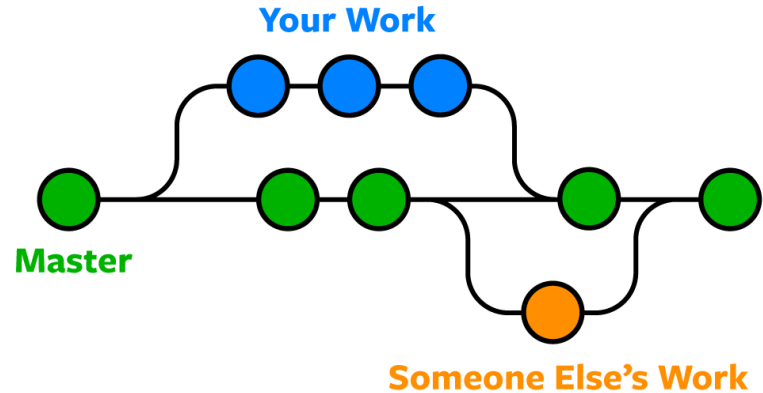
Cos'è un branch?

- Tramite il meccanismo del branch Git ci permette di continuare a sviluppare sul progetto su un «ramo» di sviluppo parallelo.
- Questo ci consente di lavorare senza doverci sincronizzare con i nostri colleghi.
- Un branch per definizione (e buona regola) deve più prima che poi confluire nel ramo principale di sviluppo.
- Più passa il tempo e più sarà doloroso far confluire il branch nel lavoro principale.



Perché creare un branch?

- **Può capitare che per implementare una feature importante ci sia bisogno di lavorarci per più di qualche giorno lavorativo.**
- **Alcune feature e ottimizzazioni durante la loro implementazione possono dare fastidio se non rompere altre funzionalità del progetto.**
- **In questo caso è molto utile fare un diverso branch, portare avanti lo sviluppo della feature su quel branch e infine «mergiare» il branch su quello principale (solitamente master o main).**



Se non volessi includere alcuni file nel repository?

- Per questo genere di evenienza ci basta scrivere l'indirizzo del file dentro il file `git.ignore`
- Dentro il file `git.ignore` possiamo dire a git di «ignorare» un singolo file, una cartella o addirittura tutti i file con una certa estensione
- Solitamente si ignorano file troppo pesanti (tipo assets grafici), file di compilazione e file che contengono chiavi di configurazione che non vogliamo assolutamente condividere.

```
# Operating System Files
# =====

# OSX
# =====

.DS_Store
.AppleDouble
.LSOverride

# Thumbnails
.*_

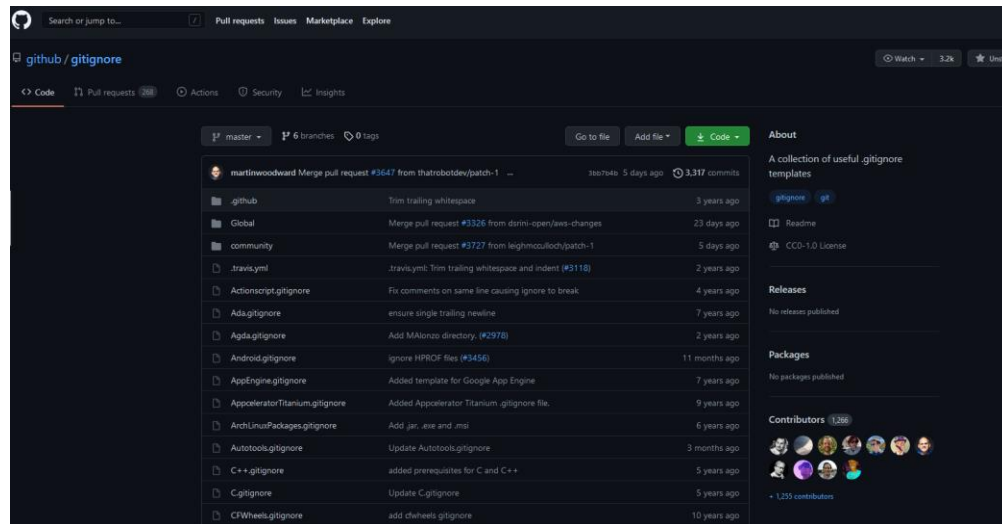
# Files that might appear on external disk
.Spotlight-V100
.Trashes

# Directories potentially created on remote AFP share
.AppleDB
.AppleDesktop
Network Trash Folder
Temporary Items
.apdisk

# Untracked files:
Codice che retriava/ratings.csv
Codice che retriava/ratingsFiltered.csv
Codice che retriava/ratingsDiscarded.csv
```

Un buon inizio per il git ignore

- Su diversi siti si trovano dei file utilizzabili come gitignore a seconda del tipo di progetto che state iniziando.
- A [questo link](#) potete trovare un'ottima collezione di file regolarmente aggiornata.
- Nei vostri progetti Java usate sempre il gitignore base per Java



Come usare git

- È possibile installare git sul proprio computer e utilizzarlo da riga di comando
- Si può utilizzare direttamente all'interno di IntelliJ, [qui](#) una guida pratica per settarlo
- Ci sono diversi programmi che offrono una comoda interfaccia grafica per utilizzarlo, ne vedremo uno in seguito.

```
david@DavMon-N551JM: ~/ProjectTest
File Modifica Visualizza Cerca Terminale Aiuto
david@DavMon-N551JM:~/ProjectTest$ git init
Initialized empty Git repository in /home/david/ProjectTest/.git/
david@DavMon-N551JM:~/ProjectTest$ git add FileTestA.log
david@DavMon-N551JM:~/ProjectTest$ git commit -m 'Test File'
[master (root-commit) 3848b34] Test File
 1 file changed, 128 insertions(+)
 create mode 100644 FileTestA.log
david@DavMon-N551JM:~/ProjectTest$ git status
Sul branch master
nothing to commit, working tree clean

david@DavMon-N551JM:~/ProjectTest$ git status
Sul branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   FileTestA.log

no changes added to commit (use "git add" and/or "git commit -a")
david@DavMon-N551JM:~/ProjectTest$
```


Consigli salvavita (e tempo)

- **Prima di lavorare al vostro codice fate sempre un pull per aggiornarvi all'ultima versione**
- **Date nomi sensati ai vostri commit che vi aiutino a capire cosa cambia dalla versione precedente**
- **Evitate come la morte di committare versioni di codice che non compilano**
- **Se ci sono bug noti importanti nel vostro codice, scrivetelo nel commit**
- **Se state lavorando ad un'importante modifica per cui ci vuole molto tempo, valutate di aprire un nuovo branch ad hoc.**
- **Più tempo passa per un branch, più sarà difficile reintegrarlo nel branch principale**
- **Prima di committare qualcosa, assicuratevi con gli altri membri del team che nessuno abbia lavorato sugli stessi vostri file, in caso contrario preparatevi ad un merge.**



ADVICE

Quando e perché creare un commit?

- **Se avete completato una feature o un task può avere senso creare un commit**
- **A fine giornata lavorativa, se il vostro codice è in uno status safe (compila e non ha bug importanti) ha senso creare un commit**
- **Se un membro del team vi avverte che sta per lavorare su un file su cui avete lavorato voi, committate (e pushate) e fategli scaricare quindi l'ultima versione.**



GitHub



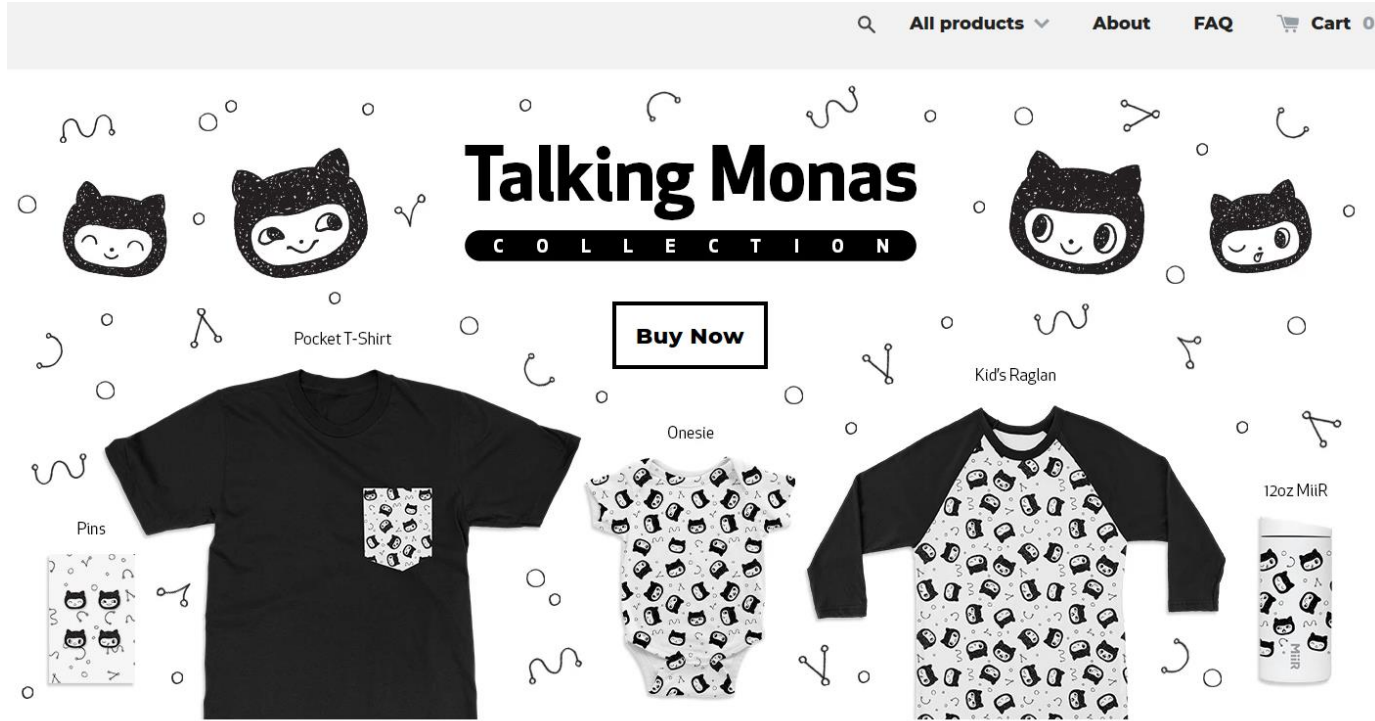


GITHUB NON È GIT

Cos'è GitHub?

- **GitHub è un sito che offre la possibilità di poter hostare in remoto le proprie repository**
- **L'account base è gratuito e permette di avere infinite repository sia pubbliche che private e infiniti collaboratori su un unico progetto**
- **Ci sono altre due opzioni per account a pagamento che offrono la possibilità di poter fare più azioni al mese e avere più spazio su GitHub. Queste opzioni hanno senso per aziende medie/grandi**
- **GitHub non è solo uno spazio per hostare progetti ma anche una grande community di sviluppatori**
- **Ci sono poi diverse funzionalità offerte in particolare per i progetti open source**
- **E poi c'è lui...**





Shirts

Alternative a GitHub



Bitbucket



GitLab



SOURCEFORGE

Qualche esempio virtuoso

- <https://github.com/spring-projects/spring-framework>
- <https://github.com/Jaeger87/Unknow-pleasure-processing>
- <https://github.com/godotengine/godot>
- <https://github.com/TheAlgorithms/Python>
- <https://github.com/mifi/lossless-cut>
- <https://github.com/EbookFoundation/free-programming-books>
- <https://github.com/freeCodeCamp/freeCodeCamp>
- <https://imdbpy.github.io/>
- <https://github.com/jwasham/coding-interview-university>
- <https://github.com/xelatihy/fondamentibook>
- https://github.com/adafruit/Adafruit_NeoMatrix

Sapersi presentare

- **Se la vostra repo è privata potete tranquillamente fregarvene della sua presentazione.**
- **Per una repo pubblica invece può aver senso preoccuparsi di come si presenti**
- **Presentare significa descrivere bene il progetto nella home così chi apre può farsi un'idea del progetto.**
- **Molti sviluppatori indipendenti curano le repo dei propri progetti personali per utilizzarle come biglietti da visita.**



ReadMe

- Lo strumento più rapido per dare un tocco di stile alla vostra repository è il readme
- Modificando questo file potrete modificare la home del vostro repository.
- Questo file può essere scritto in linguaggio markdown, un linguaggio di markup simile all'html ma con una sintassi molto più semplice
- Ci sono molte guide a riguardo sul markdown, [questa](#) scritta da GitHub ne è un buon esempio

README.md



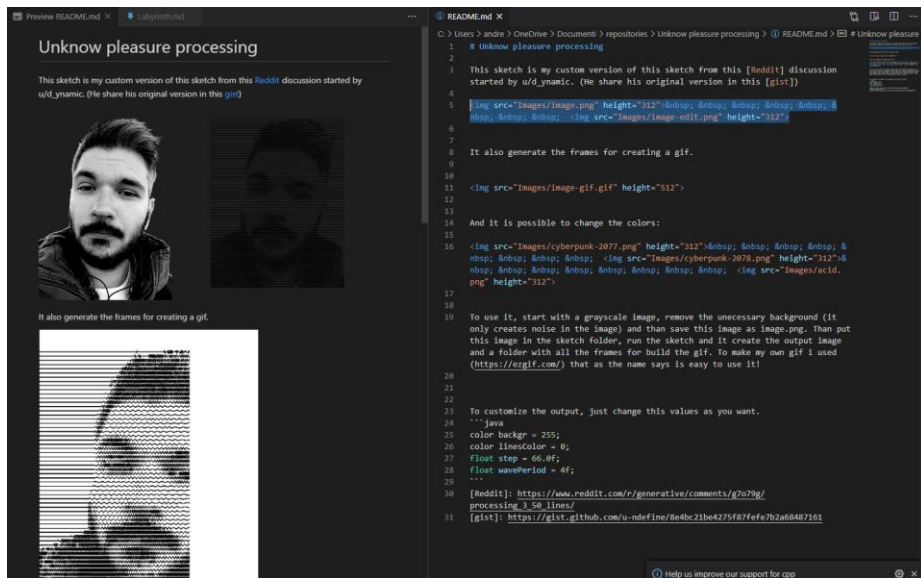
Spring Framework build passing

This is the home of the Spring Framework: the foundation for all [Spring projects](#). Collectively the Spring Framework and the family of Spring projects are often referred to simply as "Spring".

Spring provides everything required beyond the Java programming language for creating enterprise applications for a wide range of scenarios and architectures. Please read the [Overview](#) section as reference for a more complete introduction.

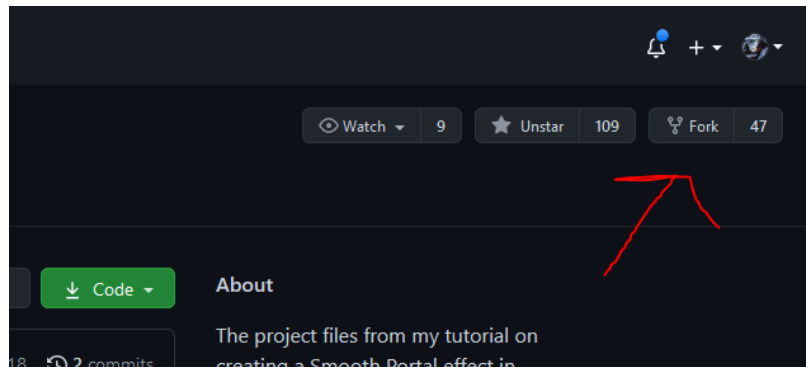
Visual studio code

- Un ottimo strumento per scrivere markdown è visual studio code, un editor di codice creato da Microsoft disponibile gratuitamente e su ogni piattaforma
- Con visual studio code non soltanto avete le evidenziazioni sul codice ma, tramite un plugin, anche un anteprima grafica di quello che state facendo.



Piccolo consiglio ninja...

- **Spesso per risolvere un problema cercate del codice di qualcuno che l'ha già risolto (pratica sempre consigliata) e lo importate nel vostro progetto.**
- **Capita però che quel codice funzioni ma presenti qualche bug molto difficile da risolvere.**
- **Un buon trucco ninja è quello di vedere su GitHub quali sono le repo che hanno forkato quel progetto e vedere se hanno risolto quel bug.**
- **Non ci crederete, ma ogni tanto funziona!**



GitKraken



electron

boards

timelines

+

repository

electron

branch

master

Undo

Redo

Fetch

Push

Branch

Stash

Pop

Search

Menu

Profile

justino@gitkraken.com

Viewing 1181/1181

Show All

Filter (⌘ + Option + f)

LOCAL

1/1

REMOTE

131/131

PULL REQUESTS

93

ISSUES

Select an issue tracker for this repo

GitHub

GitHub Enterprise

GitKraken Boards

GitLab

GitLab Self-Managed

Jira Cloud

Jira Server

Trello

None

TAGS

1049/1049

SUBMODULES

2

GITHUB ACTIONS

0

master

roller/chromium/master

trop/10-x-y-bp-fix-provi...

11-x-y

10-x-y

trop/11-x-y-bp-fix-deco...

trop/10-x-y-bp-fix-deco...

honor-printing-11-x-y

trop/10-x-y-bp-fix-...

printing-ranges-honor-...

roller/chromium/11-x-y

v12.0.0-nightly.20200923

roller/chromium/10-x-y

9-x-y

roller/node/master

SM

AL

AL

SV

T

T

JR

T

T

JR

JR

SV

SA

JK

EB

EB

T

T

EB

EB

feat(extensions): expose ExtensionRegistryObserv...

update patches

update printing.patch given print settings mojoific...

fix: provide asynchronous cleanup hooks in n-api

fix: update node certdata to NSS 3.56 (#25362)

fix: update node certdata to NSS 3.56 (#25361)

fix: check for destroyed webcontents in converter (...)

docs: remove unused StreamProtocolResponse / S...

docs: remove unused StreamProtocolResponse / S...

fix: order menu items before filtering excess separ...

fix: decompress devtools discovery html

fix: decompress devtools discovery html

fix: decompress devtools discovery html (#25576)

fix: honor pageRanges when printing

Address extra null checks & update test 1 hour ago

fix: order menu items before filtering excess separ...

2418471: PDF Viewer update: Add missing aria-lab...

fix: honor pageRanges when printing

2387901: Accessing C++ enums in Java ... 3 hours ago

update patches

update patches

Bump v12.0.0-nightly.20200923

update patches

fix: unsubscribe from observers when window is cl...

fix: unsubscribe from observers when window is cl...

fix: unsubscribe from observers when window is cl...

chore: bump chromium in DEPS to ddb5b6db5e35...

chore: bump node in DEPS to v14.12.0

commit: 9d0d9a

feat(extensions): expose ExtensionRegistryObserver events (#25385)

* feat(extensions): expose ExtensionRegistryObserver events

SM

Samuel Maddock

parent: 881ac9

authored 9/23/2020 @ 12:29 PM

G

GitHub

committed 9/23/2020 @ 12:29 PM

4 modified

Path

Tree

View all files

docs/api/session.md

shell/browser/api/electron_api_session.cc

shell/browser/api/electron_api_session.h

spec-main/extensions-spec.ts

Feedback

PRO

7.3.2



GitKraken

- **GitKraken è un software che vi permette di gestire facilmente tramite un'interfaccia grafica le vostre repository**
- **Potete agganciarlo al vostro account GitHub (o bitbucket o gitlab se li usate) e da quel momento potete clonare localmente le vostre repo e lavorarci.**
- **GitKraken vi da anche la possibilità di iniziare una nuove repo localmente e poi inviarla a GitHub**
- **È disponibile anche una board condivisa per tenere traccia dei task dei vostri progetti.**

Dev Backlog



see 1 more



Filter cards on this board... (example: "infinity stone @thanos")



Backlog

6 cards

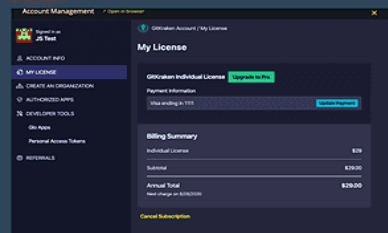
GitHub sync is next

Feature v2.4

DB-20

Show what Individual User sees when they upgrade

Bug



DB-21

Update certificate

Feature

DB-15

Additional layout setting in preferences

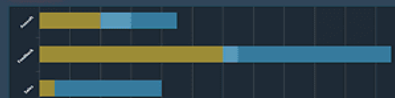
Add a card

In Progress

2 cards

Unable to connect widget over http

Bug



DB-14 12/26



Cursor should not skip when accessing description in Firefox

Bug On-Hold

Description

Jonathan created on Apr 24, 2018, 1:28 PM

User Story:

DB-17 5/15

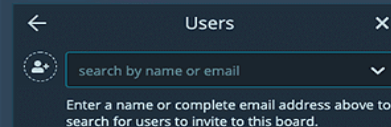
Add a card

PR is Ready

2 cards

Expand search results in the user dropdown menu

Feature v2.4



DB-8 5/25

Fix typo in connect button

Bug v2.3

DB-2 2/3

Add a card

Ready for Testing

7 cards

Edit path when installing

Feature v2.4

DB-3 1/3

Fix invalid character when access diff

Bug v2.4

DB-4

Update the copyright year

Feature

DB-7

Ability to delete items from edit window

Feature

DB-9 6/17

Link to item should not lead to 404

Bug

Check out our [Support Site](#) for documentation and tutorials!
The GitKraken team looks forward to your feedback, but please note emails.

Add a card

Alternative

- <https://gitahead.github.io/gitahead.com/>
- <https://www.sourcetreeapp.com/>
- <https://www.syntevo.com/smartgit/>



Time to make practice

A puppet dog with a surprised expression, suspended by strings, with a person's legs visible in the background.

C'È UNA NOVITÀ DA OGGI

Niente più esercizi nelle slide

- Da oggi in poi caricheremo gli esercizi direttamente in una repo su GitHub:
<https://github.com/Backend-Developer-School-Tree/Corso-Java-Developer-2023-03>
- Create un account GitHub e iniziate a seguire la repo
- Clonate la repo nel vostro PC
- Da adesso in poi per avere i nuovi esercizi vi basterà «pullare» la repo!
- Non dimenticatevi di lasciare una star alla repo!