



# Github

Il drive degli sviluppatori

# Riepilogo comandi (1)

Clonare una repository	<code>git clone &lt;repository_url&gt;</code>
Aggiungere tutti i file modificati nella cartella corrente (e sottocartelle) in un commit	<code>git add .</code> <code>git commit -m "messaggio"</code>
Visualizzare lo stato dei file	<code>git status</code>
Visualizzare i cambiamenti non committati	<code>git diff</code>
Visualizzare i commit precedenti	<code>git log</code>
Visualizzare le modifiche in un commit	<code>git show &lt;commit id&gt;</code>

## Riepilogo comandi (2)

Mandare online i commit locali	<code>git push</code>
Scaricare in locale i commit remoti	<code>git pull</code>
Creare un branch	<code>git checkout -b &lt;nome branch&gt;</code>
Posizionarsi su un branch	<code>git checkout &lt;nome branch&gt;</code>
Cancellare le modifiche correnti non ancora salvate su un commit	<code>git stash</code>
Fare un merge di un altro branch	<code>git merge &lt;nome branch&gt;</code>

# Preparazione

Assicuratevi di:

- Avere a disposizione un account Github
- Aver installato sul proprio dispositivo l'applicativo Git
- Riuscire a creare una repository su Github
- Riuscire a clonare la repository in locale
- Inviare un commit sulla repository

# Preparazione - esempio comandi

- [GITHUB] creazione repository repo-prova
- [GITHUB] copiare url della repository (es. <https://github.com/allemonta/repo-prova> )
- Aprire un terminale e spostarsi sul Desktop

`cd Desktop`

`git clone https://github.com/allemonta/repo-prova`

`cd repo-prova`

- CREARE FILE NELLA CARTELLA “repo-prova” che sarà stata creata sul proprio desktop

`git add .`

`git commit -m “primo commit”`

`git push`

```
→ ~ cd Desktop
→ Desktop git clone https://github.com/allemonta/repo-prova.git
Cloning into 'repo-prova'...
warning: You appear to have cloned an empty repository.
→ Desktop cd repo-prova
→ repo-prova git:(main) git add .
→ repo-prova git:(main) x git commit -m "Primo commit"
[main (root-commit) 522dcab] Primo commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.html
→ repo-prova git:(main) git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 229 bytes | 229.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/allemonta/repo-prova.git
* [new branch]      main -> main
```



# Github Pages

# Github Pages

Link riferimento: <https://pages.github.com/>

- 1) Creare una repository **pubblica** `<username>.github.io` (dove dovrete sostituire `<username>` con il vostro username di github)
- 2) Eseguire nuovamente i passaggi della “preparazione” inserendo all’interno della repository un file “index.html”

Controllare ora il sito <https://<username>.github.io>





# Github Organizations

# Github Organizations

- 1) Creare una nuova organizzazione
- 2) Inserire tutti i componenti del gruppo nell'organizzazione con il ruolo di OWNER
- 3) Creare una repository privata all'interno della organizzazione
- 4) Il compagno A esegue le solite operazioni della "preparazione", inviando in un commit un file con un codice che contiene un errore
- 5) Il compagno A crea una issue su github che segnala che c'è un problema XXXX nel file YYYY. La issue deve essere assegnata al compagno B
- 6) Il compagno B clona la repository sul proprio computer
- 7) Il compagno B esegue le solite operazioni della "preparazione" inviando online un commit che risolve il problema del compagno A
- 8) Il compagno B chiude la issue a lui associato
- 9) Il compagno A recupera le modifiche fatte dal compagno B (`git pull`)

# Valutazione PCTO

<https://forms.gle/ZNQbafVys21inzaG8>



# Interesse Workshop

<https://forms.gle/bKP2fL1QkCjeyLVt8>



# Links

Github Copilot: <https://github.com/features/copilot>

Github for students: <https://education.github.com/students>

Valutazione PCTO: <https://forms.gle/ZNQbafVYs21inzaG8>

Workshop in azienda: <https://www.soluzionifutura.it/sf-code-lab>

[ChatGPT che gioca a scacchi](#) :)