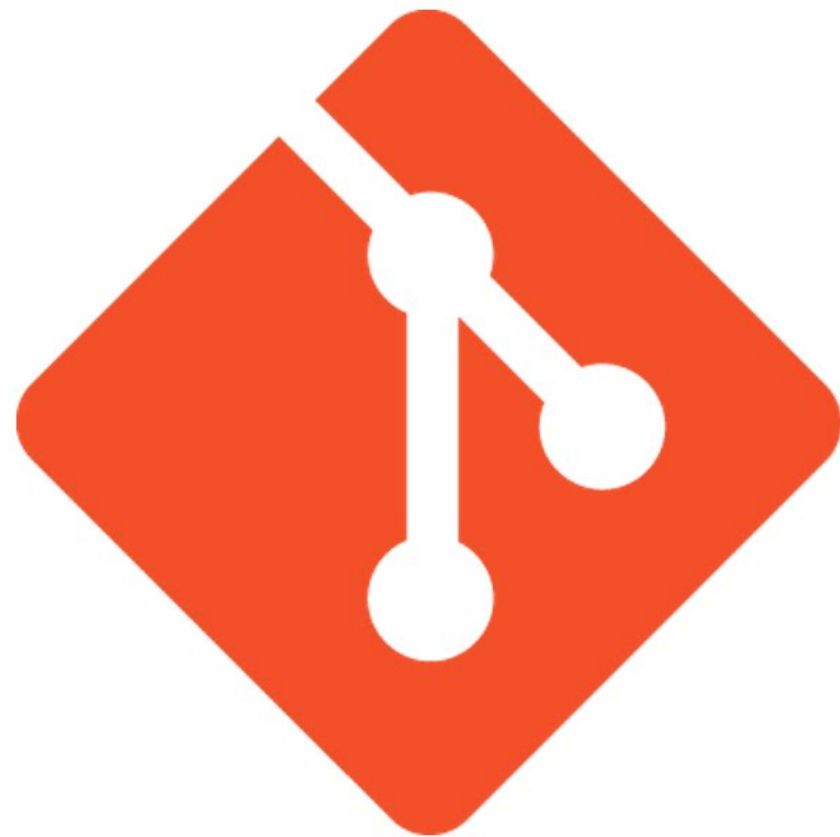


Git and GitHub

- The beginning -



SUMMARY

1. CONFIGURIAMO L'AMBIENTE

2. GIT E GitHub

3. COME FUNZIONA GIT



SUMMARY

1. CONFIGURIAMO L'AMBIENTE

2. GIT E GitHub

3. COME FUNZIONA GIT



LINUX (Fedora) **Command:** `sudo yum install git`

LINUX (Debian) **Command:** `sudo apt-get install git`

WINDOWS <http://git-scm.com/download/win>

MAC OS X <http://git-scm.com/download/mac>



Your Identity

```
git config --global user.name "Your Name"
```

```
git config --global user.email example@ncsu.edu
```

Your editor

```
git config --global core.editor <editor like vim, emacs etc>
```

Enable color in git

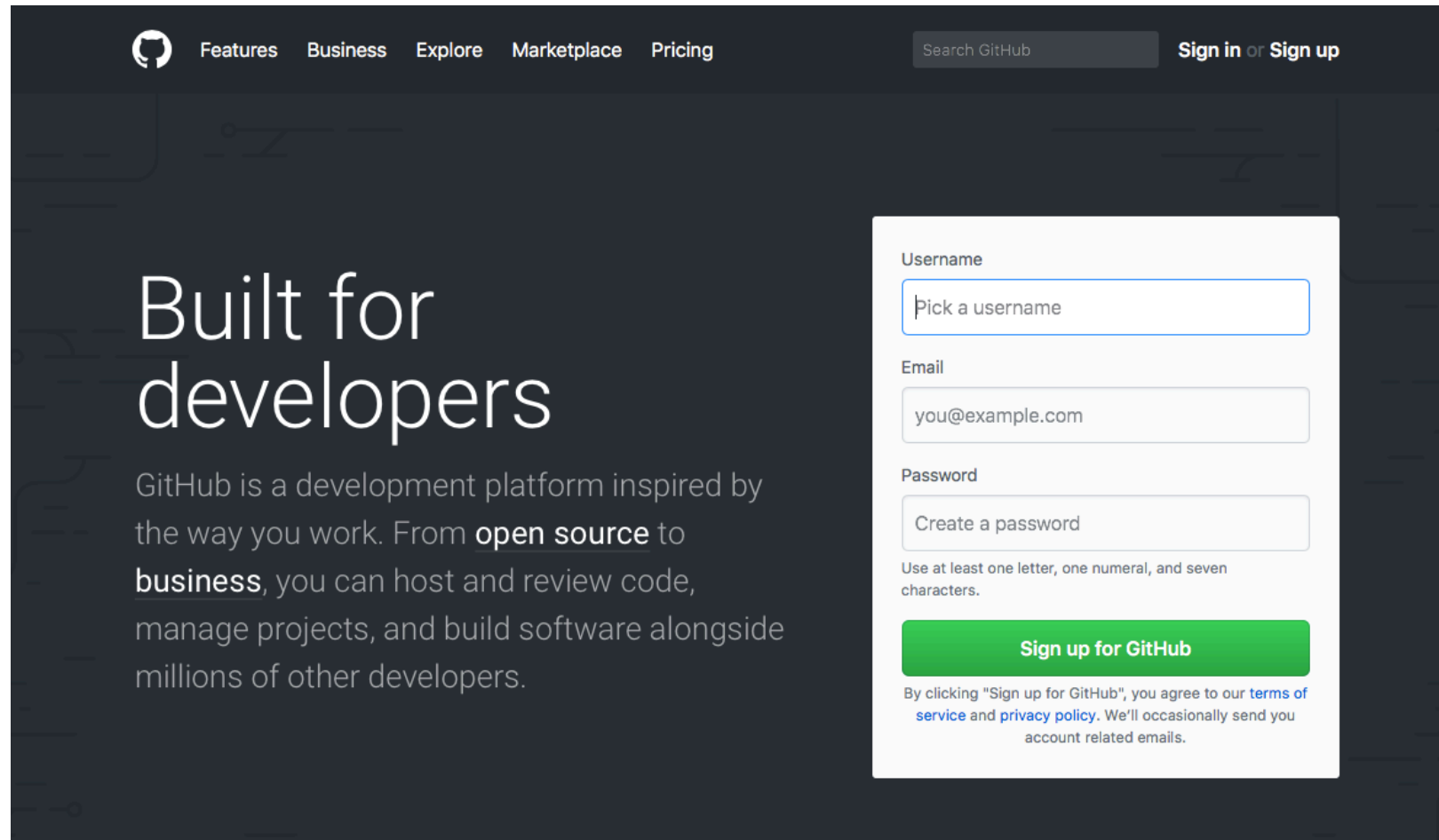
```
git config --global color.ui auto
```

Checking your settings

```
git config --list
```



www.github.com



The screenshot shows the GitHub homepage with a dark background. At the top, there is a navigation bar with links for Features, Business, Explore, Marketplace, and Pricing. A search bar labeled "Search GitHub" and links for "Sign in" or "Sign up" are also present. The main heading reads "Built for developers". Below this, a paragraph describes GitHub as a development platform inspired by the way you work, from open source to business, where you can host and review code, manage projects, and build software alongside millions of other developers. On the right side, a white sign-up form is overlaid. It contains fields for Username (with placeholder "Pick a username"), Email (with placeholder "you@example.com"), and Password (with placeholder "Create a password"). Below the password field, there is a note: "Use at least one letter, one numeral, and seven characters." A green button labeled "Sign up for GitHub" is at the bottom of the form. Below the button, a small disclaimer states: "By clicking 'Sign up for GitHub', you agree to our [terms of service](#) and [privacy policy](#). We'll occasionally send you account related emails."

Features Business Explore Marketplace Pricing Search GitHub Sign in or Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside millions of other developers.

Username
Pick a username

Email
you@example.com

Password
Create a password

Use at least one letter, one numeral, and seven characters.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We'll occasionally send you account related emails.



SUMMARY

1. CONFIGURIAMO L'AMBIENTE

2. GIT E GitHub

3. COME FUNZIONA GIT



VERSION CONTROL

VERSION CONTROL

UN SISTEMA CHE TIENE TRACCIA DELLE MODIFICHE APPORTATE

VERSION CONTROL

**CONSENTE AGLI SVILUPPATORI DI COLLABORARE
E CAPIRE CHI E QUANDO EFFETTUA CAMBIAMENTI**

VERSION CONTROL

**CONSENTE AGLI SVILUPPATORI DI COLLABORARE
E CAPIRE CHI E QUANDO EFFETTUA CAMBIAMENTI**

**PERMETTE DI RIPRISTINARE LE MODIFICHE
TORNANDO ALLO STATO PRECEDENTE!!!**

VERSION CONTROL

**CONSENTE AGLI SVILUPPATORI DI COLLABORARE
E CAPIRE CHI E QUANDO EFFETTUA CAMBIAMENTI**

**PERMETTE DI RIPRISTINARE LE MODIFICHE
TORNANDO ALLO STATO PRECEDENTE!!!**

**GLI SVILUPPATORI POSSONO MANTENERE IL CODICE E
LA SUA STORIA SULLE PROPRIE MACCHINE, EFFETTUANDO MODIFICHE**

VERSION CONTROL



SISTEMA DI CONTROLLO VERSIONE DISTRIBUITO

VERSION CONTROL



ALTRI SISTEMI DI CONTROLLO VERSIONE



VERSION CONTROL



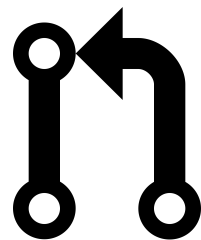
PIATTAFORMA DI CONTROLLO DI VERSIONE E
COLLABORAZIONE WEB PER SVILUPPATORI BASATA SU GIT

VERSION CONTROL



PIATTAFORMA DI CONTROLLO DI VERSIONE E
COLLABORAZIONE WEB PER SVILUPPATORI BASATA SU GIT

PULL REQUEST



BUG TRACKING

SUMMARY

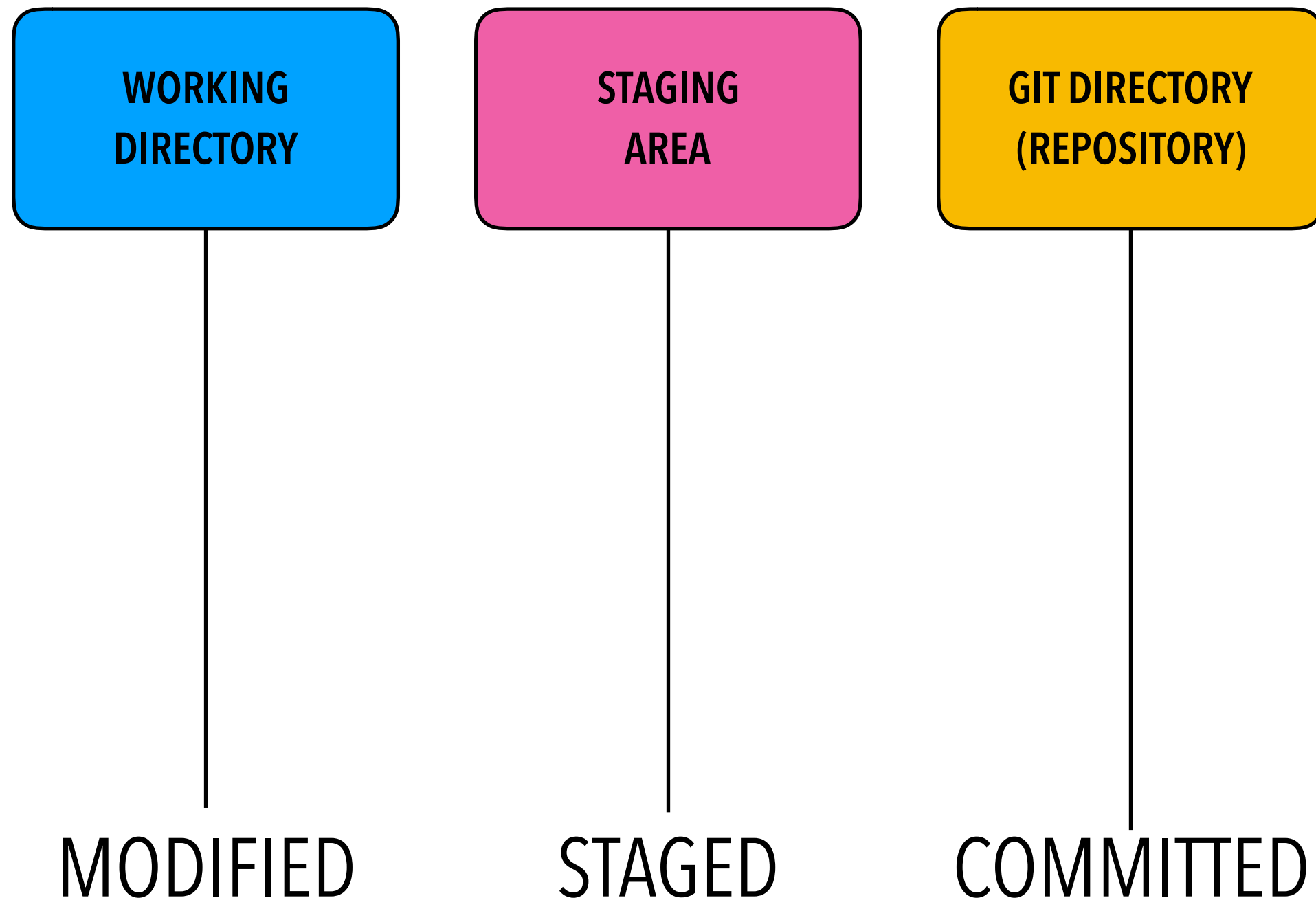
1. CONFIGURIAMO L'AMBIENTE

2. GIT E GitHub

3. COME FUNZIONA GIT



GIT WORKFLOW



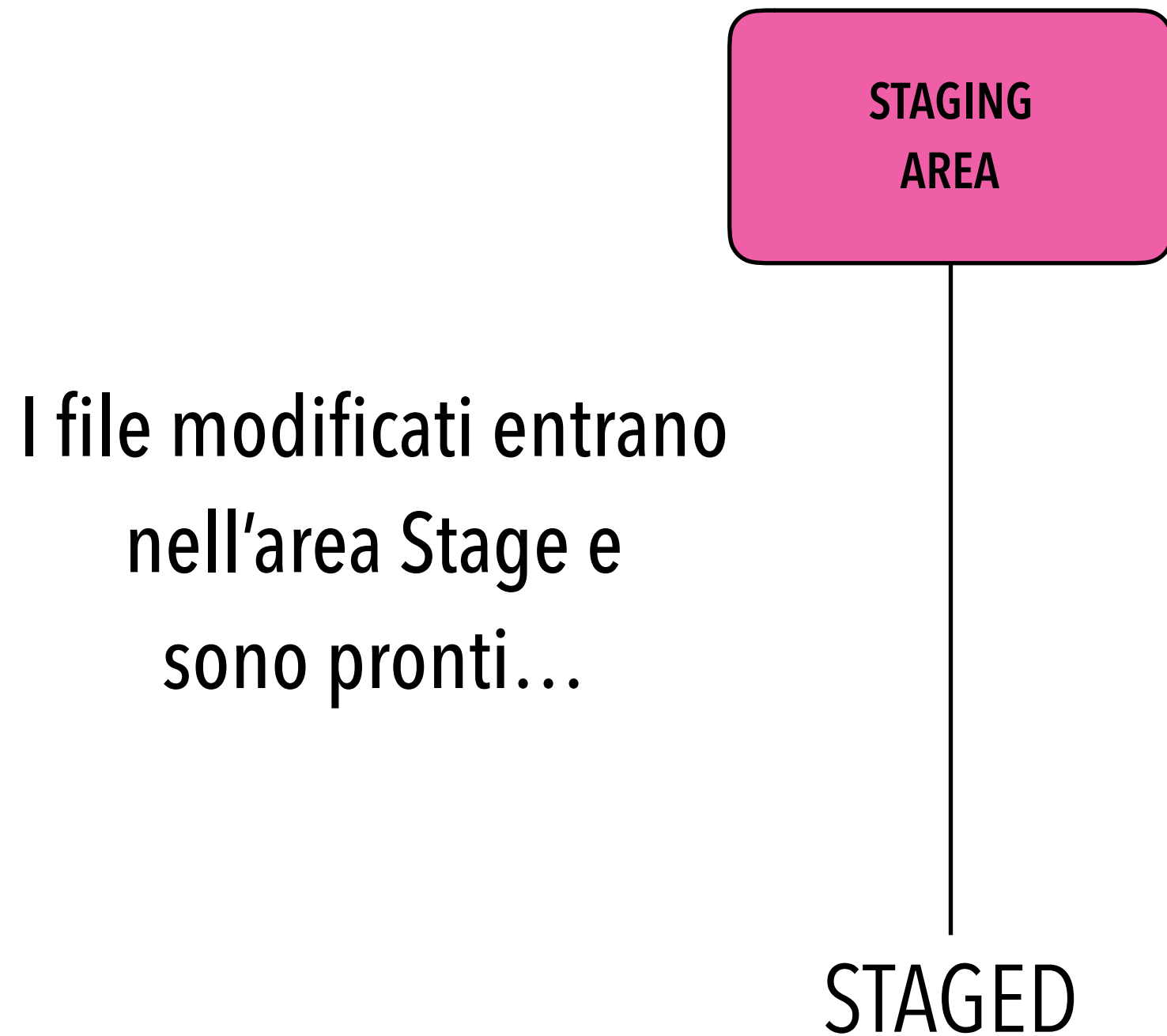
GIT WORKFLOW



Lo sviluppatore modifica
uno o più file nella propria
Working Directory

MODIFIED

GIT WORKFLOW



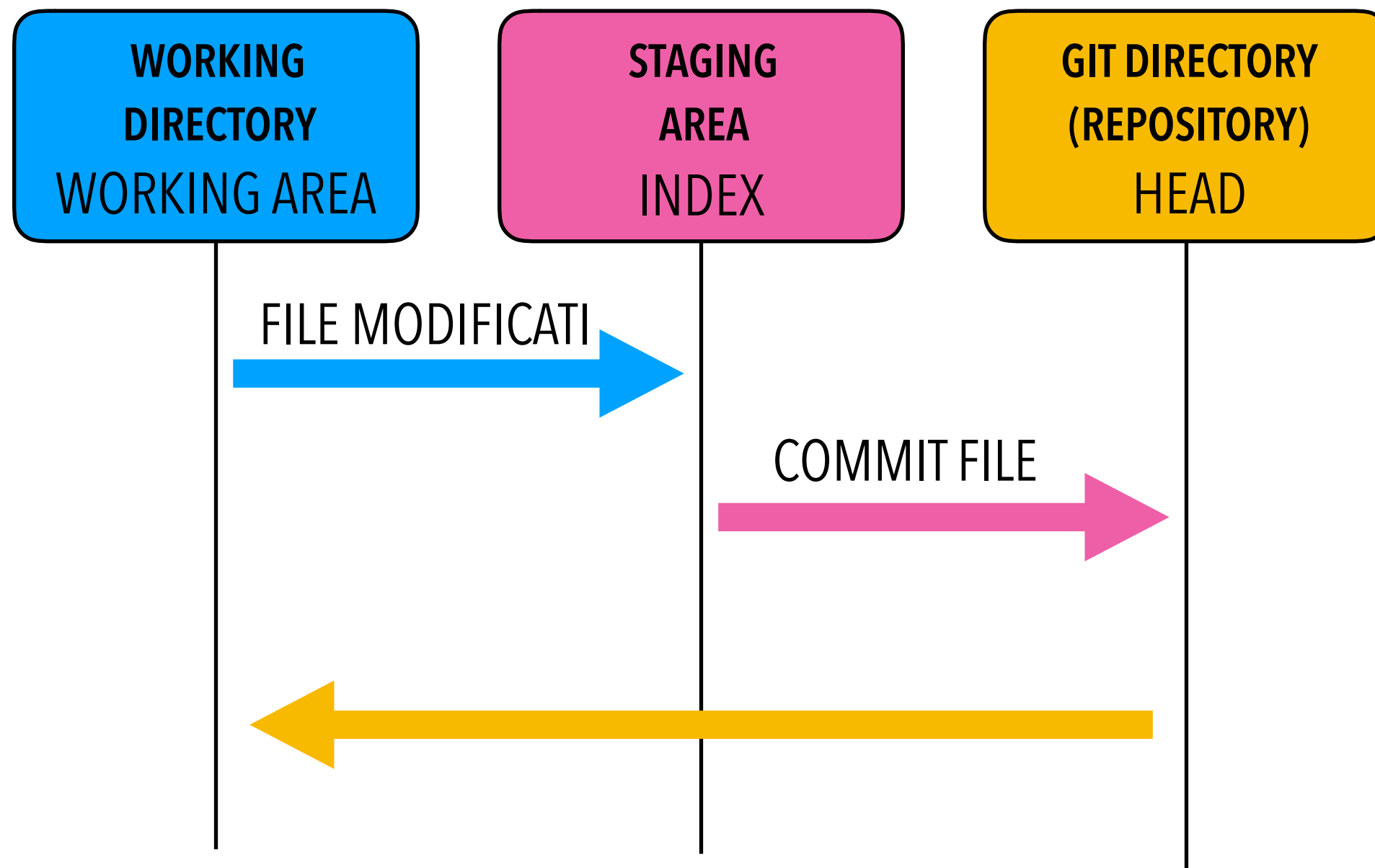
GIT WORKFLOW

per essere memorizzati
in maniera permanente
nella nostra Git Directory!



COMMITTED AND PUSH

GIT WORKFLOW



LO SVILUPPATORE POTRA' RIAGGIORNARE LA SUA VERSIONE LOCALE
E FARE MODIFICHE, QUINDI FAR RICOMINCIARE L'ITER!

PASSIAMO ALLA PRATICA!

CREIAMO LA NOSTRA WORKING AREA - REPOSITORIES



Create la vostra cartella

Command: `git init`

IN BASE AL TIPO DI PROGETTO,
E AL LINGUAGGIO DI PROGRAMMAZIONE
GITHUB VI CONSENTE DI COSTRUIRE
IN AUTOMATICO IL DOCUMENTO GITIGNORE



INIZIALIZZATE IL VOSTRO REPOSITORY UTILIZZANDO LA PIATTAFORMA GITHUB!!!!

CREIAMO LA NOSTRA WORKING AREA - REPOSITORIES

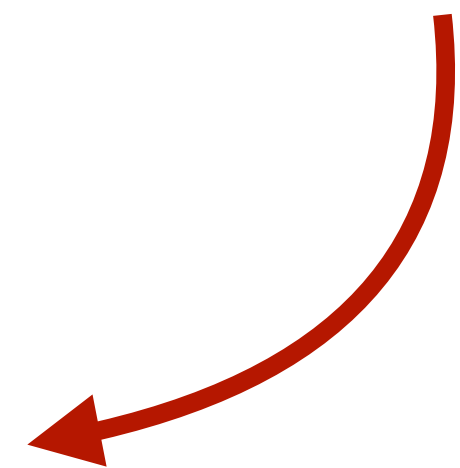


Create la vostra cartella

Command: `git init`

IN BASE AL TIPO DI PROGETTO,
E AL LINGUAGGIO DI PROGRAMMAZIONE
GITHUB VI CONSENTE DI COSTRUIRE
IN AUTOMATICO IL DOCUMENTO GITIGNORE

IN QUESTO MODO IL VCS CAPISCE AUTOMATICAMENTE
QUALI FILE (DI CONFIGURAZIONE) NON DEVONO ESSERE TOCCATI
DURANTE LE MODIFICHE DI SVILUPPO



CHECKOUT REPOSITORIES



Command

```
git clone /percorso/del/repository
```

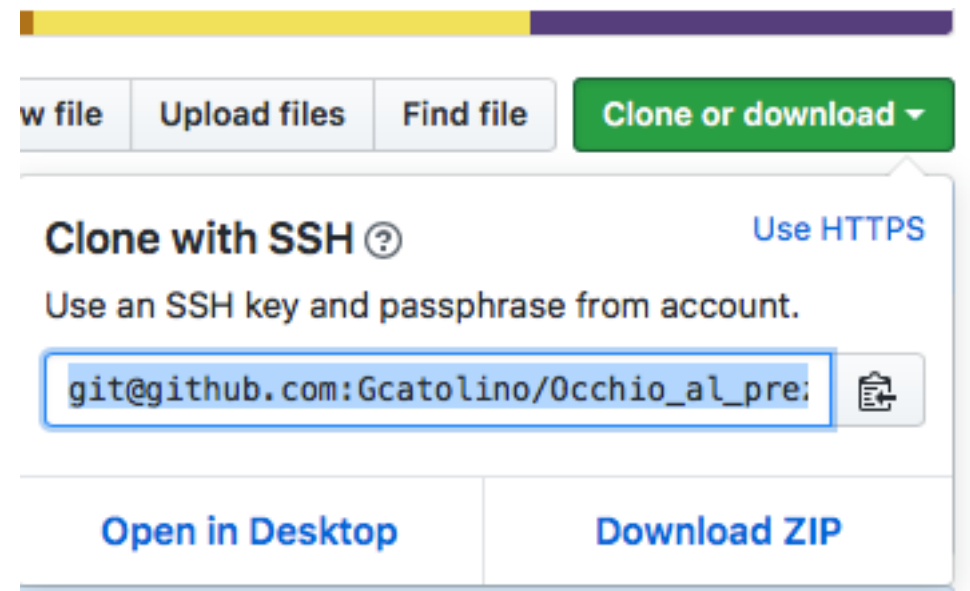
CHECKOUT REPOSITORIES



Command

```
git clone /percorso/del/repository
```

ANDANDO SUL VOSTRO REPOSITORY TROVERETE
CLONE COPIATE IL LINK
E POI DA COMANDO TERMINALE
git clone "link di github"



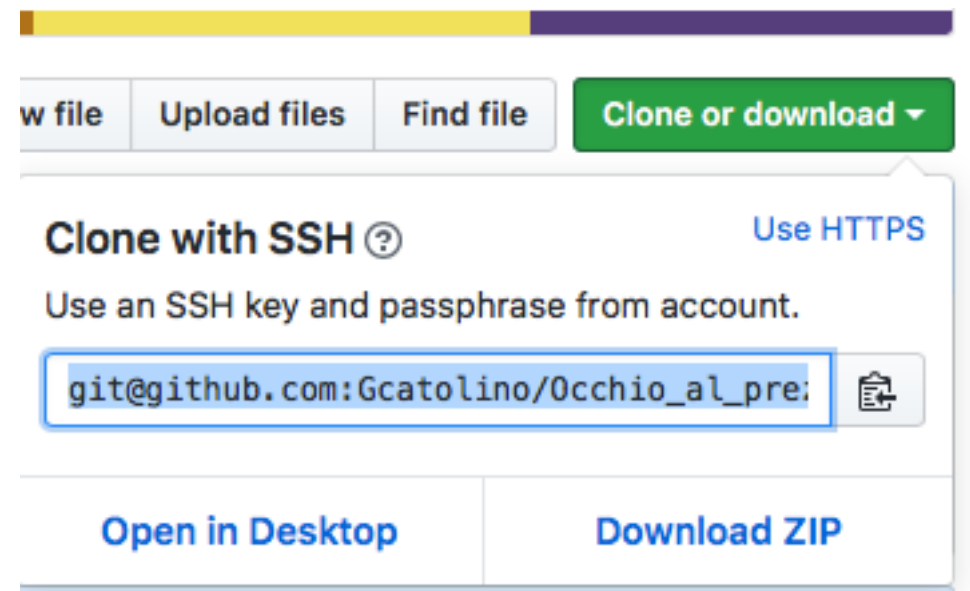
CHECKOUT REPOSITORIES



Command

```
git clone /percorso/del/repository
```

ANDANDO SUL VOSTRO REPOSITORY TROVERETE
CLONE COPIATE IL LINK
E POI DA COMANDO TERMINALE
git clone "link di github"



AGGIUNTA E VALIDAZIONE FILE



Aggiunta

`git add <nome del file>`

`git add *` (ovvero tutti i file)

AGGIUNTA E VALIDAZIONE FILE



Command

git status

Controllate lo stato del vostro repository locale rispetto al repository su GitHub

AGGIUNTA E VALIDAZIONE FILE



Aggiunta

`git add <nome del file>`

`git add *` (ovvero tutti i file)

UNA VOLTA AGGIUNTI I FILE NELLA VOSTRA CARTELLA, BISOGNA
VALIDARE QUESTA MODIFICA UTILIZZANDO L'OPERAZIONE DI **COMMIT**

`git commit -m "Messaggio per la commit"`

AGGIUNTA E VALIDAZIONE FILE



Aggiunta

```
git add <nome del file>
```

```
git add * (ovvero tutti i file)
```

DA QUEL MOMENTO IN POI OGNI VOLTA CHE MODIFICATE I FILE VI BASTERÀ FARE SOLO
OPERAZIONE DI **COMMIT**

```
git commit -m "Messaggio per la commit"
```


INVIO DELLE MODIFICHE



Command: `git push origin master`

FINO AL MOMENTO DEL GIT PUSH LE VOSTRE MODIFICHE RIMANGONO
NELL' HEAD DELLA NOSTRA WORKING AREA.

DOPO L'OPERAZIONE DI PUSH LE MODIFICHE ANDRANNO SUL REPOSITORY REMOTO

AGGIORNAMENTO, MERGE, SOSTITUZIONE



Command

git pull

Aggiornare il repository locale al commit più recente

AGGIORNAMENTO, MERGE, SOSTITUZIONE



Command

`git pull`

Aggiornare il repository locale al commit più recente

`git fetch origin`

`git checkout -- <nomefile>`

Recuperare le modifiche fatte sul server remoto di tutto o di un solo file

AGGIORNAMENTO, MERGE, SOSTITUZIONE



Command

`git pull`

Aggiornare il repository locale al commit più recente

`git fetch origin`

`git checkout -- <nomefile>`

Recuperare le modifiche fatte sul server remoto di tutto o di un solo file

`git merge`

Incorporare le modifiche fatte sul server remoto, **non è una procedura automatizzata!**

BRANCH



UTILIZZATI PER SVILUPPARE FEATURES CHE SONO ISOLATE L'UNA DALL'ALTRA.
IL BRANCH MASTER E' DI DEFAULT IL PRIMO CREATO

Command

```
git brach new_branch_name
```

Per creare un nuovo Branch

BRANCH



UTILIZZATI PER SVILUPPARE FEATURES CHE SONO ISOLATE L'UNA DALL'ALTRA.
IL BRANCH MASTER E' DI DEFAULT IL PRIMO CREATO

Command

```
git brach new_branch_name
```

Per creare un nuovo Branch

```
git checkout other_branch_name
```

Per passare da un brach ad un altro

BRANCH



UTILIZZATI PER SVILUPPARE FEATURES CHE SONO ISOLATE L'UNA DALL'ALTRA.
IL BRANCH MASTER E' DI DEFAULT IL PRIMO CREATO

Command

```
git brach new_branch_name
```

Per creare un nuovo Branch

```
git checkout other_branch_name
```

Per passare da un brach ad un altro

```
git branch -d branch_name
```

Cancellare un branch

BRANCH



PER INCORPORARE I BRANCH NEL MASTER PUOI UTILIZZARE IL COMANDO **MERGE**

Command

```
git brach new_branch_name
```

Per creare un nuovo Branch

```
git checkout other_branch_name
```

Per passare da un brach ad un altro

```
git branch -d branch_name
```

Cancellare un branch



SCARICANDO ECLIPSE GIT TEAM PROVIDER

<https://git-scm.com/book/en/v2/Getting-Started-Git-Basics>

<http://rogerdudler.github.io/git-guide/index.it.html>

<https://www.slideshare.net/HubSpot/git-101-git-and-github-for-beginners>

<https://www.slideshare.net/akrish/introduction-to-gitgithub-a-beginners-guide>

MATERIALE DI AIUTO

Git and GitHub

- The beginning -

Bisogno di aiuto?

gcatolino@unisa.it

