# **پاورپوینت آموزشی Git**

[پاورپوینت آموزشی Git](#_d0c4s91ukam2)

[مقدمه](#_d592i3w8zuio)

[انواع سیستم‌های کنترل نسخه](#_dk2x4i6qjum2)

[آشنایی با Git](#_13sp6kh1wdwu)

[دستورات اولیه Git](#_kgffz3nu771)

[مفاهیم شاخه‌ها (Branches)](#_s039ffh87gi0)

[دستورات مهم Git](#_susmmyr3rmwl)

[مراحل کار عملی با Git](#_7i2m56ok90qp)

[مفاهیم ریپازیتوری](#_surttv4lhsmy)

[ایجاد اکانت GitHub و ریپازیتوری](#_u763cwfr97fu)

[ارسال فایل پاورپوینت به GitHub](#_x6yqq0p1t7e2)

[نتیجه‌گیری](#_hbkp7e4d7s7e)

# 

## **مقدمه**

Git یکی از سیستم‌های کنترل نسخه (VCS) محبوب و پرکاربرد است که برای مدیریت کدها و پروژه‌های نرم‌افزاری استفاده می‌شود. این سیستم به برنامه‌نویسان کمک می‌کند تا تغییرات کدها را پیگیری کرده و با دیگر اعضای تیم به‌صورت هماهنگ کار کنند.

## **انواع سیستم‌های کنترل نسخه**

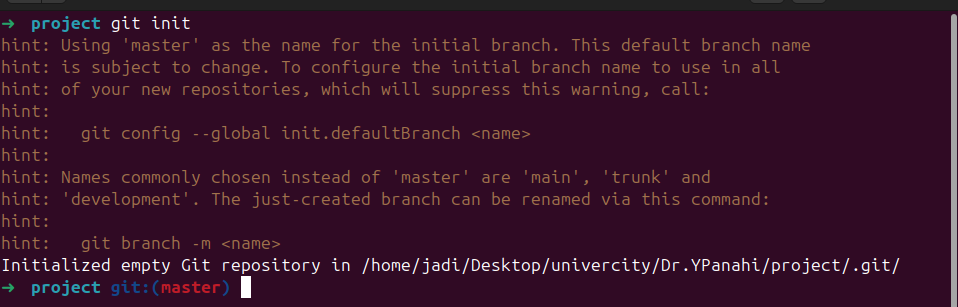
1. **سیستم‌های محلی (Local Version Control Systems)**
   * نگهداری تغییرات روی یک سیستم محلی.
   * ابزارهایی مانند RCS.
2. **سیستم‌های متمرکز (Centralized Version Control Systems)**
   * یک سرور مرکزی که تمام تغییرات روی آن ذخیره می‌شود.
   * مثال: Subversion (SVN).
3. **سیستم‌های توزیع‌شده (Distributed Version Control Systems)**
   * نسخه کامل پروژه در هر کامپیوتر نگهداری می‌شود.
   * مثال: Git.

## **آشنایی با Git**

* **Git**: سیستم کنترل نسخه توزیع‌شده.
* **ویژگی‌ها**:
  + سرعت بالا.
  + قابلیت کار به صورت آفلاین.
  + ادغام راحت شاخه‌ها (Branches).

## **دستورات اولیه Git**

**ایجاد ریپازیتوری محلی**:  
1. git init



**پیکربندی کاربر**:

➜ project git:(master) git config --global user.name mehdi kazemi

➜ project git:(master) git config --global user.email [souityab@gmail.com](mailto:souityab@gmail.com)"

**مشاهده تنظیمات**:  
git config --list

#### **3. ایجاد یک فایل جدید (touch)**

* touch todo\_list.py

4 . مشاهده تاریخچه تغییرات (git log)



## **مفاهیم شاخه‌ها (Branches)**

* **شاخه (Branch)**: مسیر جداگانه‌ای برای توسعه.

### **چرا به شاخه نیاز داریم؟**

**وقتی روی یک پروژه کار می‌کنیم، ممکن است بخواهیم:**

1. **ویژگی جدیدی (feature) به پروژه اضافه کنیم.**
2. **خطایی (bug) را اصلاح کنیم.**
3. **تغییرات بزرگ یا آزمایشی را بدون تأثیر بر پروژه اصلی (main branch) انجام دهیم.**

**ایجاد شاخه به شما این امکان را می‌دهد که تغییرات را به‌صورت جداگانه و مستقل از پروژه اصلی توسعه دهید. وقتی تغییرات کامل شد و مطمئن بودید که مشکلی ایجاد نمی‌کند، می‌توانید شاخه جدید را به شاخه اصلی ادغام (merge) کنید.**

* **دستورات**:

ایجاد شاخه جدید :

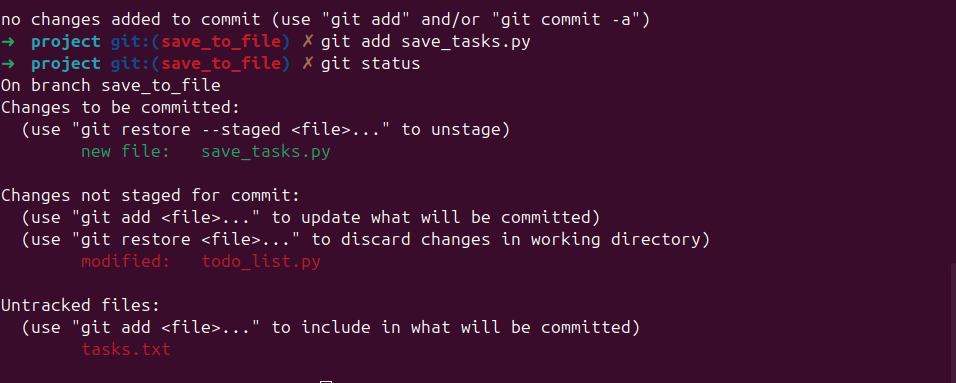
git branch save\_to\_file

تغییر شاخه:

git checkout save\_to\_file:

OR

Git switch save\_to\_file:



## 

## 

## 

## 

## **دستورات مهم Git**

**اضافه کردن فایل‌ها به مرحله آماده‌سازی**

**(Staging Area)**

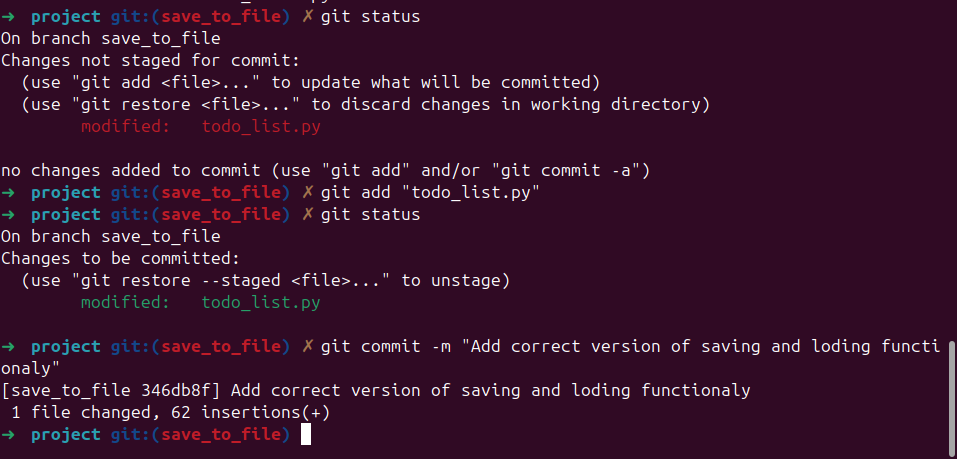
git add to\_do\_list.py

**کامیت کردن تغییرات**:

git commit -m "Add correct version of saving and loding functionaly"

**بررسی وضعیت ریپازیتوری**:

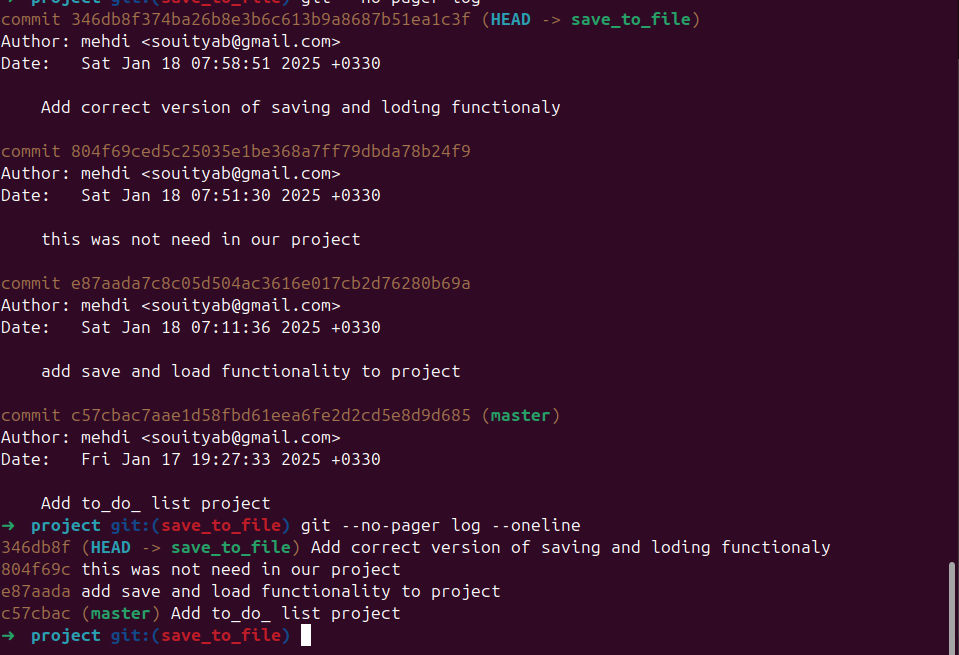
git status



**مشاهده لاگ تغییرات**:

git log

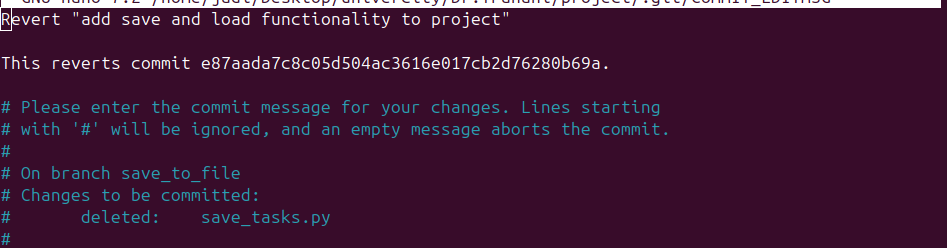
git log --oneline



#### **بازگرداندن یک کامیت خاص (git revert):**

1. **اجرای git log --oneline و یافتن شناسه کامیت (hash).**
2. **بازگرداندن تغییرات کامیت خاص:**

**git revert e87aada7c8c05d504ac3616e017cb2d76280b69a**

****

### **بازگرداندن تغییرات (git revert و git reset)**

git restore --staged file\_name :

git reset --soft HEAD~1 :

بازگرداندن تغییرات به حالت **soft** (فقط بازگشت به مرحله آماده‌سازی):

git reset --mixed HEAD~1 :

بازگرداندن تغییرات به حالت **mixed** (خروج از مرحله آماده‌سازی):

git reset --hard HEAD~1 :

بازگرداندن کامل تغییرات (**hard reset**):

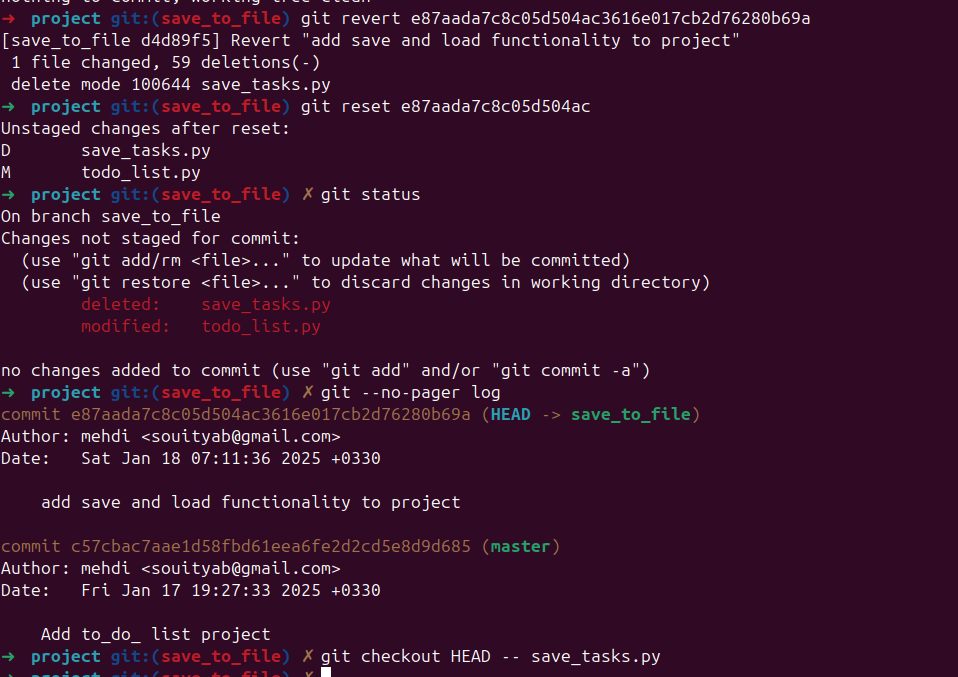
git revert

ابتدا کامیت مربوط به اضافه کردن قابلیت ذخیره‌سازی , معکوس شد.

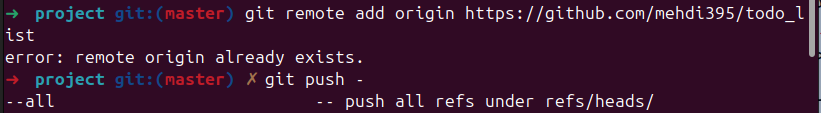
git reset

سپس پروژه به وضعیت قبلی بازگردانده شد و فایل‌های تغییر یافته بررسی شدند.

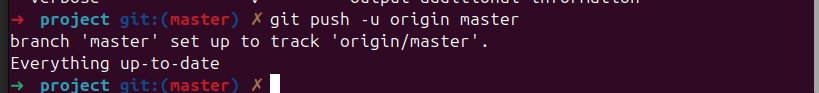
git checkout

در نهایت، فایل حذف‌شده بازگردانده شد.

**آپلود روی GitHub**:

  
  
 git remote add origin https://github.com/mehdi395/todo\_list

git push -u origin master

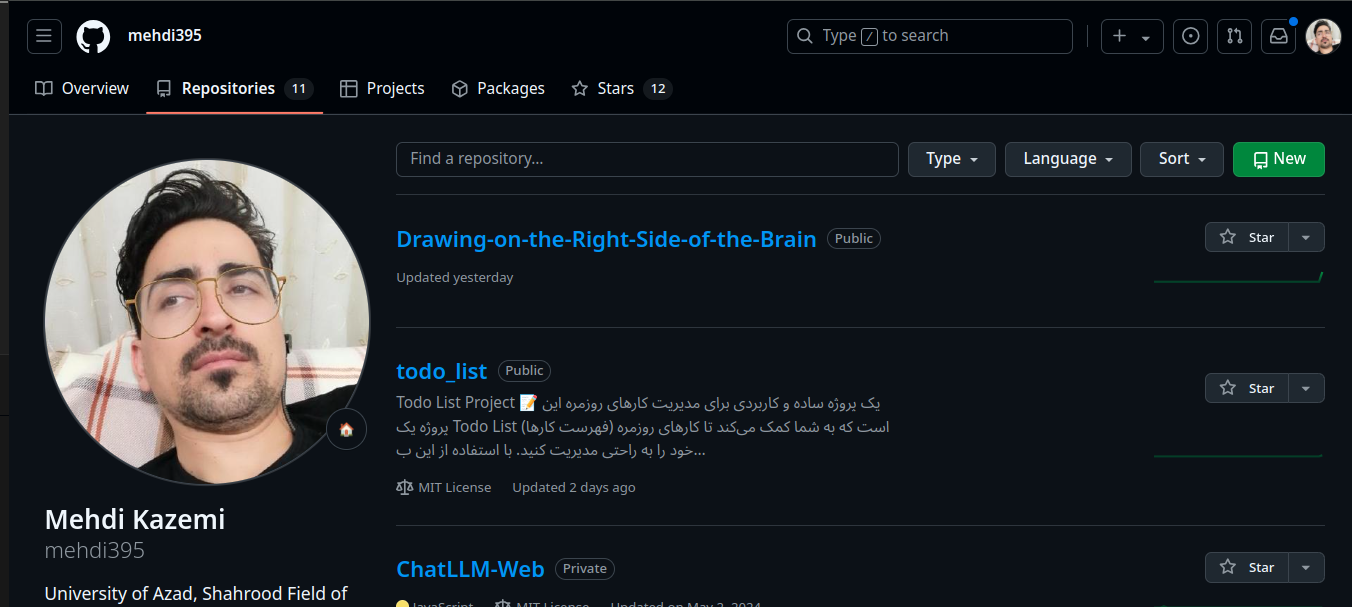
.

## **مفاهیم ریپازیتوری**

* **ریپازیتوری محلی (Local Repository)**: مخزن روی سیستم کاربر.
* **ریپازیتوری راه دور (Remote Repository)**: مخزن روی سروری مانند GitHub.

## **ایجاد اکانت GitHub و ریپازیتوری**

1. **ایجاد اکانت**: به آدرس [GitHub](https://github.com/) بروید و ثبت‌نام کنید.
2. **ایجاد ریپازیتوری**:
   * وارد حساب شوید.
   * روی دکمه "New" کلیک کنید.
   * نام ریپازیتوری را وارد کنید.
   * تنظیمات را ذخیره کنید.



## **ارسال فایل پاورپوینت به GitHub**

1. فایل پاورپوینت آماده‌شده را در پوشه پروژه قرار دهید.

مراحل زیر را در خط فرمان اجرا کنید:  
 git add .

git commit -m "Add presentation"

git push

1. لینک ریپازیتوری را کپی کرده و برای استاد ارسال کنید.

## **نتیجه‌گیری**

با استفاده از Git و GitHub، شما می‌توانید به‌راحتی پروژه‌های خود را مدیریت کرده و با دیگران همکاری کنید. استفاده از این ابزارها برای هر برنامه‌نویس یا دانشجو ضروری است.