أساسيات Gitلمهندسي البرمجيات

دليل شامل للطلاب لفهم واستخدام Gitبكفاءة في مشاريع هندسة البرمجيات.

تهيئة Gitوتكوينه

تهيئة المستودع

لبدء تتبع مشروعك باستخدام Git، تحتاج إلى تهيئة مستودع جديد في مجلد المشروع.

```
git init
```

هذا ينشئ مجلد `.git` مخفي لتخزين سجل المشروع.

ضبط إعدادات المستخدم

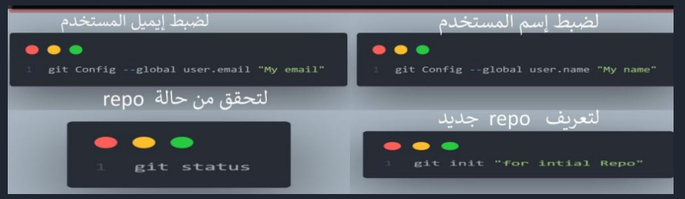
الخطوة الأولى هي تعريف نفسك لـ Git. هذا يضمن أن تكون جميع التغييرات التي تجريها مرتبطة بهويتك.

```
git config --global user.email
"your.email@example.com"git config --global
user.name "Your Name"
```

تتبع الملفات والالتزام بالتغييرات

فحص حالة المستودع

`git status` يعرض حالة ملفات العمل والمؤقتة والمستودع، موضحًا أي تغييرات لم يتم تتبعها أو التزام بها.



إضافة الملفات للمنطقة المؤقتة

`git add <file>` يضيف الملفات إلى المنطقة المؤقتة (Staging Area)، مما يعني أن هذه التغييرات جاهزة ليتم تضمينها في الالتزام التالي.

```
git add FirstAssignment.txt
```

الالتزام بالتغييرات

`git commit -m "رسالة"` يحفظ التغييرات المؤقتة بشكل دائم في سجل المستودع.

```
git commit -m
"FirstAssignment"
```

عرض سجل الالتزامات

هذا السجل ضروري لتتبع تقدم المشروع والعودة إلى إصدارات سابقة إذا لزم الأمر.

يعرض الأمر `git log` سجل الالتزامات (Commits) كاملاً للمستودع، موضحًا كل التزام مع معرفه الفريد (Commit ID)، والمؤلف، والتاريخ، ورسالة الالتزام.

```
git log
```

Python

لإضافة الملف الى Stage Area

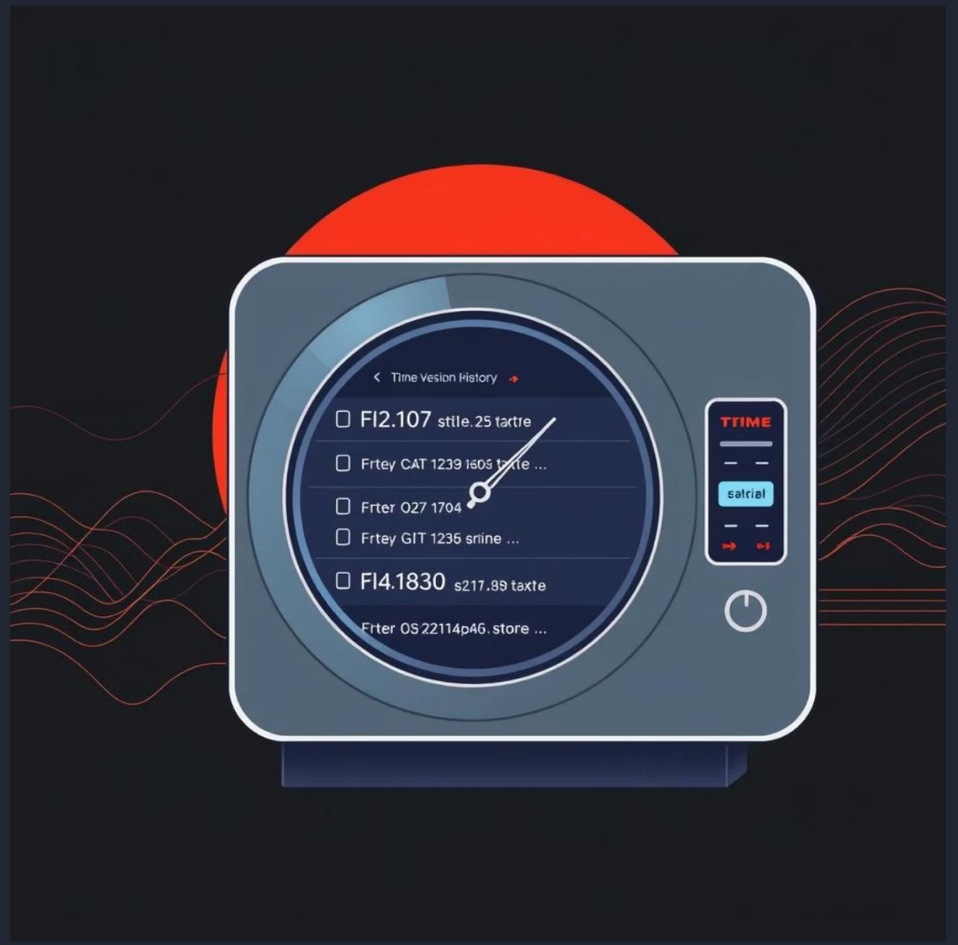
```
git add 'filename'
```

لحفظ التغيرات وعمل snapshot

```
git commit -m 'message'
```

لإضهار كل السجل الخاص ل repo

```
git log
```



التعامل مع الفروع (Branches)

git

عرض الفروع

`git branch` يعرض جميع الفروع الموجودة في مستودعك، ويشير إلى الفرع الحالي.

التحويل بين الفروع

`git checkout <فـرع>` للتبديل إلى فرع موجود.

لرجوع الى commit محدد باستخدام Id

```
git checkout 'commitID'
```

إنشاء فرع جديد والتحويل إليه

`git checkout -b <اسم_الـفـرع>` ينشئ فرعًا جديدًا ويتحول إليه مباشرةً.

لتحويل الى فرع اخر

دمج الفروع (Merging Branches)

`git merge <اسم_الفرع>` يدمج التغييرات من فرع آخر إلى فرعك الحالي.

```
git merge testing_branch
```

هذه الخطوة أساسية لتوحيد عمل المطورين المختلفين في المشروع.

أوامر Gitالمتقدمة

عرض الملفات المتعقبة

`git ls-files`: يسرد جميع الملفات التي يتتبعها Git حاليًا في مستودعك.

تنظيف الملفات غير المتعقبة

`git clean -dn`: يعرض الملفات والدلائل التي سيتم حذفها ولم يتم تتبعها بعد.

حذف الملفات غير المتعقبة

`git clean -df`: يحذف فعليًا الملفات والدلائل التي لم يتم تتبعها بعد.

التراجع عن التغييرات المحلية

`git restore <file>`: للتراجع عن التغييرات غير الملتزم بها في ملف معين.

حذف ملف ملتزم

`git rm <file>`: يحذف ملفًا من المستودع ويضع علامة عليه ليتم حذفه في الالتزام التالي.

لسحب الملفات من Stage Area

الخلاصة والخطوات التالية

لقد غطيت الآن أساسيات Git، من التهيئة إلى إدارة الفروع والالتزامات.

المفاهيم الرئيسية

- المستودعات (Repositories)
- الالتزامات (Commits)
- الفروع (Branches) والدمج (Merging)
- حالة العمل والمنطقة المؤقتة

الممارسة الموصى بها

- الالتزام بانتظام بتغييرات صغيرة.
- استخدام رسائل التزام وصفية.
- العمل على فروع منفصلة للميزات.
- التعاون بفعالية مع أعضاء الفريق.

مصادر إضافية

- توثيقات Git الرسمية
- Learn Git Branching

شكرًا لكم!