



git + GitHub

اعداد :

م. أحمد حاج محمد سلامة

الجامعة الدولية للعلوم والنهضة



ماذا سوف نتعلم في هذا المقرر



اساسيات Git

- 🐱 مرحلة الادراج Staging في Git
- 🐱 إيداع المساهمات Committing في Git
- 🐱 إيداع نفس الملف مرة اخري
- 🐱 الاطلاع على سجل الايداعات
- 🐱 التراجع عن عملية إيداع في Git

اساسيات Git

- 🐱 ما هي Git
- 🐱 كيف نشأت Git ولماذا
- 🐱 تثبيت Git
- 🐱 تعريف حساب Git محلي
- 🐱 البدا باستخدام Git
- 🐱 حالات الملفات في Git



ماذا سوف نتعلم في هذا المقرر

التفرعات Branches في Git

دمج فروع في Git

حذف فرع في Git

تعارض عمليات الدمج Merge conflicts

حل التعارض Conflict Resolve

تثبيت أداة gitk

التفرعات Branches في Git

ما هي التفرعات Branches في Git

إنشاء فرع جديد في Git

الانتقال الى فرع اخر في Git

فرع الإصلاحات الطارئة Hot fixes



ماذا سوف نتعلم في هذا المقرر

مدخل GitHub

- 🐙 إنشاء الفروع Branches في GitHub
- 🐙 مخطط عمل GitHub
- 🐙 خدمة GitHub Pages لاستضافة صفحات الويب
- 🐙 الاشتقاق في GitHub
- 🐙 استنساخ المستودعات Repositories
- 🐙 Clonning في GitHub

مدخل GitHub


- 🐙 Git أداة عمل موزع Distributed
- 🐙 تسجيل حساب جديد في GitHub
- 🐙 إنشاء مستودع في GitHub
- 🐙 تعريف المستودع بعيد محليا
- 🐙 ارسال التعديلات المحلية الى المستودع البعيد push
- 🐙 تعديل الملفات في GitHub مباشرة
- 🐙 جلب التعديلات المستودع البعيد الى المستودع المحلي pull




ماذا سوف نتعلم في هذا المقرر

مواضيع متقدمة في Git

ملف التجاهل GitIgnore 

السفر في الزمن Git Checkout 

الامر Reset 

استخدام Commit –amend 

git blame 

git bisect 



اساسيات Git



Git هي أداة تستخدمها فرق العمل بشكل عام والفرق البرمجية بشكل خاص لإدارة إصدارات المشروع الذي يتعاون أعضاء الفريق على إنجازه ونعني بـ "إدارة الإصدارات" إمكانية تتبع التغيرات التي أجراها أعضاء الفريق على ملفات المشروع خلال مراحل تنفيذه المختلفة بحيث يمكن تحديد وقت وتاريخ كل تغير إضافة لعضو الفريق المسؤول المتسبب بهذا التغير

- ما هي Git
- كيف نشأت Git ولماذا
- تثبيت Git
- تعريف حساب Git محلي
- البدء باستخدام Git
- حالات الملفات في Git



اساسيات Git



يعتبر لينوس تورفالدز (منشئ نظام لينكس الأب الروحي لـ git و تعود أسباب نشأة git إلى حاجة الفريق العامل على تطوير نواة لينكس حينها إلى أداة مناسبة لإدارة الإصدارات و صديقة لفرق عمل البرمجيات مفتوحة المصدر، خصوصاً بعد النزاع الذي نشأ بينهم و بين الشركة القائمة على تطوير نظام BitKeeper الذي كانوا يستعملونه وقتها (أبريل) ٢٠٠٥ حيث أن الأخير مرخص تجارياً و بالتالي لاستخدامه من قبل مجتمع كبير من المطورين تبعات قانونية كانت سبباً في إشعال فتيل النزاع.

- ما هي Git
- كيف نشأت Git ولماذا
- تثبيت Git
- تعريف حساب Git محلي
- البدء باستخدام Git
- حالات الملفات في Git



اساسيات Git



كما ذكرنا في التعريف، فإن الأهداف الرئيسية لاستخدام git تتلخص في:

1. تسجيل التغييرات التي يجريها المطورون في شيفرات المشاريع البرمجية.
2. تسجيل المتسبب بكلّ من تلك التغييرات و تاريخ كل منها.
3. إتاحة إمكانية التعاون في المشاريع البرمجية بحيث يمكن لأكثر من عضو في الفريق العمل على نفس الملف/الملفات في نفس الوقت دون ضياع أية مساهمات.

ما هي Git

كيف نشأت Git ولماذا

تثبيت Git

تعريف حساب Git محلي

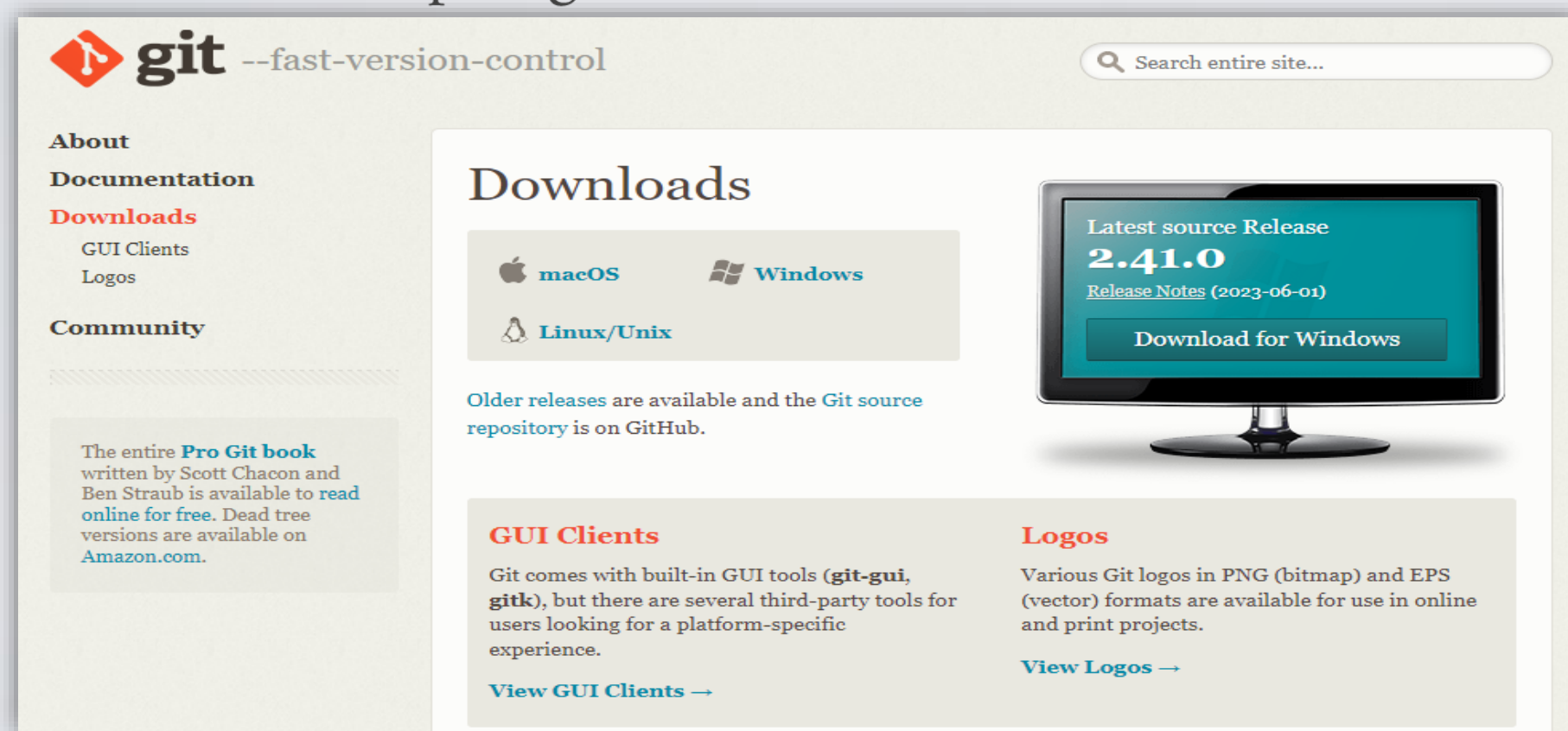
البدء باستخدام Git

حالات الملفات في Git



اساسيات Git

تختلف آلية تثبيت git باختلاف نظام التشغيل المستخدم :
بالنسبة لنظام Windows تقوم بتحميله من الموقع الرسمي عبر الرابط التالي:
<https://git-scm.com/downloads>



- ما هي Git
- كيف نشأت Git ولماذا
- تثبيت Git
- تعريف حساب Git محلي
- البدء باستخدام Git
- حالات الملفات في Git



اساسيات Git



قبل أن نبدأ باستخدام git يفضل أن نقوم بمعرفة اصدار git و بتعريف الحساب المحلي الخاص بنا على git، و ذلك من خلال تنفيذ هذه الاوامر:

```
git -v  
git config --global user.name "ahmadSalameh1995"  
git config --global user.email "a.salama998877@gmail.com"
```

للتحقق من أن المعلومات الخاصة بالمستخدم تم تعيينها بنجاح، يمكنك استخدام الأمر:

```
git config --global user.name  
git config --global user.email
```

ما هي Git

كيف نشأت Git ولماذا

تثبيت Git

تعريف حساب Git محلي

البدء باستخدام Git

حالات الملفات في Git



اساسيات Git



ملحوظة أولى: لا تنس تغيير اسم المستخدم وعنوان البريد في المثال أعلاه إلى اسم المستخدم الخاص بك و عنوان بريدك الإلكتروني.

ملحوظة ثانية : كوننا استعملها الوسيط العام global فهذا يعني أن الحساب الذي قمنا بتعريفه سيكون هو الحساب المستخدم في جميع مستودعات git على هذا الحاسوب، ما لم نقوم بتخصيص بعض المستودعات بغير ذلك صراحةً و سنأتي على ذكر معنى المستودع بعد قليل.

```
MINGW64:/  
A.Salama@DESKTOP-409U06R MINGW64 /  
$ git config --global user.name "ahmadSalameh1995"  
A.Salama@DESKTOP-409U06R MINGW64 /  
$ git config --global user.email "a.salama998877@gmail.com"  
A.Salama@DESKTOP-409U06R MINGW64 /  
$ git config --global user.name  
ahmadSalameh1995  
A.Salama@DESKTOP-409U06R MINGW64 /  
$ git config --global user.email  
a.salama998877@gmail.com
```

- ما هي Git
- كيف نشأت Git ولماذا
- تثبيت Git
- تعريف حساب Git محلي
- البدء باستخدام Git
- حالات الملفات في Git



اساسيات Git



و الآن، يمكننا البدء باستخدام git لإدارة إصدارات المشروع الذي نعمل عليها حالياً، و ذلك من خلال الانتقال إلى مجلد المشروع، ثم إدخال الأمر التالي كما تبين الصورة أدناه، و ذلك لتحويله إلى مستودع git، حيث أننا نستخدم مصطلح "مستودع" Repository للإشارة إلى أي مشروع تدار إصداراته باستخدام git، و أمر التهيئة هو :

git init

و بهذا نكون قد قمنا بالتو بإنشاء أول مستودع git بنجاح، مبارك مبارك

- ما هي Git
- كيف نشأت Git ولماذا
- تثبيت Git
- تعريف حساب Git محلي
- البدء باستخدام Git
- حالات الملفات في Git



اساسيات Git



```
MINGW64:/c/users/A.salama/Desktop/test
A. Salama@DESKTOP-409U06R MINGW64 /d/university/nahda
$ cd /

A. Salama@DESKTOP-409U06R MINGW64 /
$ cd c

A. Salama@DESKTOP-409U06R MINGW64 /c
$ cd users

A. Salama@DESKTOP-409U06R MINGW64 /c/users
$ cd A.salama

A. Salama@DESKTOP-409U06R MINGW64 /c/users/A.salama
$ cd Desktop

A. Salama@DESKTOP-409U06R MINGW64 /c/users/A.salama/Desktop
$ cd test

A. Salama@DESKTOP-409U06R MINGW64 /c/users/A.salama/Desktop/test
$ git init
Initialized empty Git repository in C:/Users/A.Salama/Desktop/test/.git/
```

او نختار المجلد بالزر الماوس الأيمن ثم نفتح به باستخدام git bash
ثم نكتب الامر git init مباشرة

- ما هي Git
- كيف نشأت Git ولماذا
- تثبيت Git
- تعريف حساب Git محلي
- البدء باستخدام Git
- حالات الملفات في Git



اساسيات Git



لقد أنشأنا مستودع git محلياً في الفقرة السابقة، غير أنه فارغ و لا يحوي أية ملفات، لذا سنقوم بإنشاء ملف جديد و ليكن ملف index.html على سبيل التجربة، ثم نقوم بحفظ ذلك الملف و إضافته

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1>Hello Git</h1>
</body>
</html>
```

ما هي Git

كيف نشأت Git ولماذا

تثبيت Git

تعريف حساب Git محلي

البدء باستخدام Git

حالات الملفات في Git



اساسيات Git



و الآن إذا عدنا إلى سطر الأوامر ، و استعرضنا الملفات الموجودة في مجلدنا الجديد فيجب أن نرى ملفنا index.html فقط كما تبينه الصورة التالية:
لاستعراض الملفات نكتب الامر التالي

dir

ملحوظة: في لينكس و ماك أو إس، يستخدم الأمر ls لاستعراض الملفات الموجودة في المجلد، و يقابله الأمر dir في سطر أوامر ويندوز.

```
A.Salama@DESKTOP-409U06R MINGW64 /c/users/A.salama/Desktop/test (master)
$ dir
index.html
```

- ما هي Git
- كيف نشأت Git ولماذا
- تثبيت Git
- تعريف حساب Git محلي
- البدء باستخدام Git
- حالات الملفات في Git



اساسيات Git



الآن يمكننا أن نتحقق من حالة تتبع الملفات في git باستخدام نكتب الامر التالي

git status

و الذي سنستعمله كثيراً ضمن صفحات هذا الكتاب، حيث يستعمل هذا الأمر لرؤية الوضع الحالي لملفات المشروع بالنسبة لـ git، و في مثالنا الحالي بما أنّ الملف جديد فسيظهره لنا git على أنه ملف جديد لم يتم تتبعه بعد Untracked و لم نقم بالاحتفاظ بأي إصدار" منه كما تبين لقطة الشاشة أدناه.

```
A.Salama@DESKTOP-409U06R MINGW64 /c/users/A.salama/Desktop/test (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  index.html
```

- ما هي Git
- كيف نشأت Git ولماذا
- تثبيت Git
- تعريف حساب Git محلي
- البدء باستخدام Git
- حالات الملفات في Git



اساسيات Git



فبالنسبة لـ git تكون الملفات الموجودة في مستودع ما على إحدى حالتين:

Tracked: و تشير إلى الملفات التي يعرفها git والمضافة إلى المستودع من قبل.

Untracked: و تشير إلى الملفات الجديدة التي لم تُضف إلى المستودع بعد.

سنتعلم بعد قليل الأمر المستعمل لنقل الملفات من حالتها غير المتبعة إلى حالة التتبع مروراً
بمرحلة الإدراج ، و هي ما سنناقشه في الفقرة التالية.

ما هي Git

كيف نشأت Git ولماذا

تثبيت Git

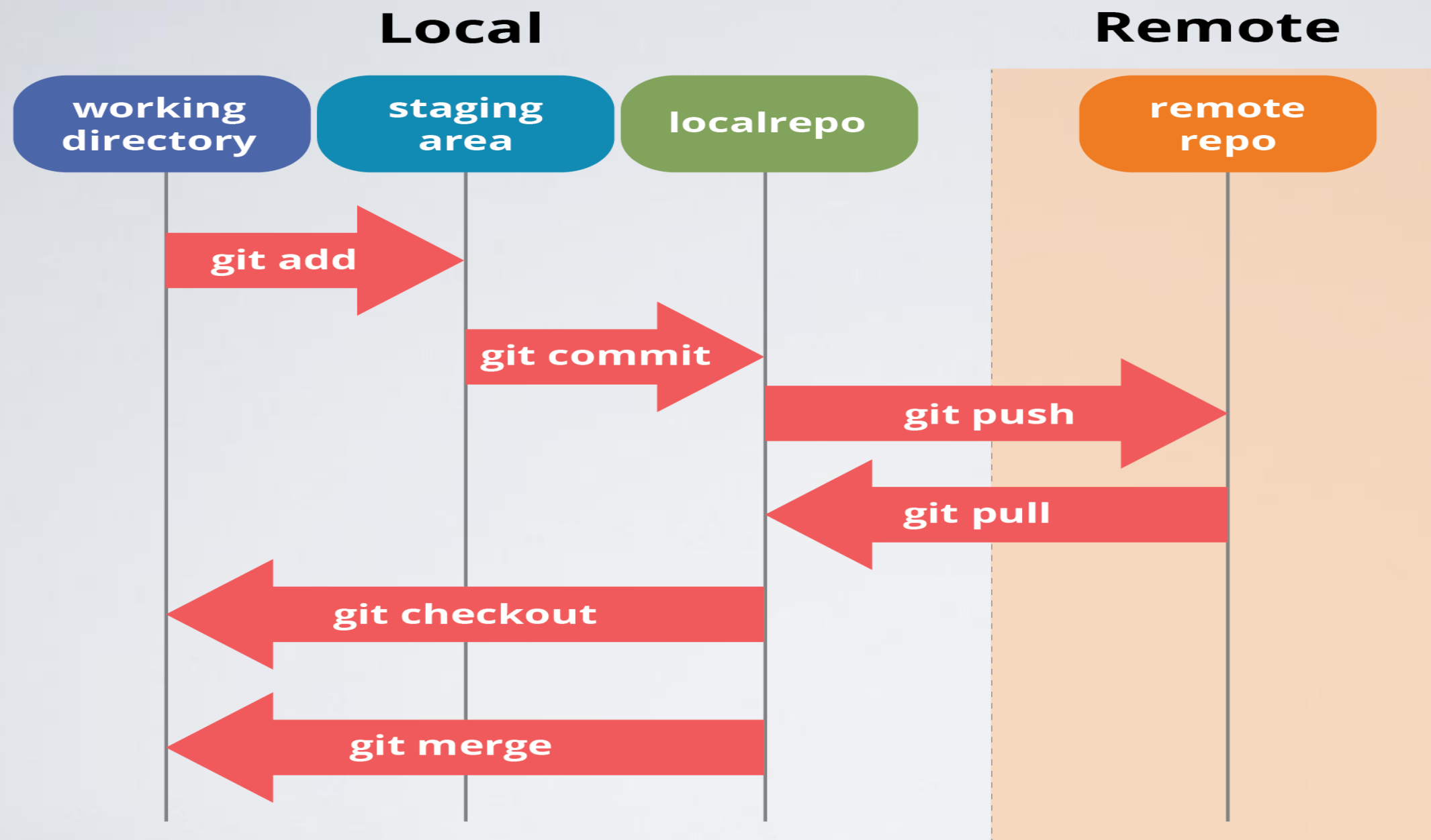
تعريف حساب Git محلي

البدء باستخدام Git

حالات الملفات في Git



اساسيات Git



- مرحلة الادراج في Git
- إيداع المساهمات في Git
- إيداع نفس الملف مرة أخرى
- الاطلاع على سجل الايداعات
- التراجع عن عملية إيداع في Git



اساسيات Git



تمثل مرحلة الإدراج Staging الحالة التي تسبق مرحلة التتبع مباشرة للملف الذي ننوي تتبعه لأول مرة، أو تتبع التغييرات التي تمت عليه لاحقاً، بما فيها حذفه، حيث لا يمكننا نقل الملف أو تغييراته من حالة عدم التتبع إلى حالة التتبع مباشرة دون المرور بمرحلة الإدراج.

و الملفات المدرجة هي ملفات جاهزة للإيداع كمساهمة commit في المستودع الذي تعمل عليه، وهو ما سنتعرض له بالتفصيل أثناء شرح عملية إيداع المساهمة commit في الفقرة اللاحقة،

مرحلة الادراج في Git

إيداع المساهمات في Git

إيداع نفس الملف مرة أخرى

الاطلاع على سجل الايداعات

التراجع عن عملية إيداع في Git



اساسيات Git



أما الآن و كوننا أتممنا العمل على الملف index.html، فلنقم بإضافته إلى مرحلة الإدراج باستخدام الأمر add بالشكل التالي:

```
git add index.html
```

```
A.Salama@DESKTOP-409U06R MINGW64 /c/users/A.salama/Desktop/test (master)
$ git add index.html
```

مرحلة الادراج في Git

إيداع المساهمات في Git

إيداع نفس الملف مرة أخرى

الاطلاع على سجل الايداعات

التراجع عن عملية إيداع في Git



اساسيات Git



و الآن إذا قمنا بتنفيذ أمر التحقق من حالة git الأمر git status فسنرى انتقال الملف من حالة عدم التتبع بشكل كلي، إلى حالة الإدراج staging كما تبينه الصورة التالية:

```
A.Salama@DESKTOP-409U06R MINGW64 /c/users/A.salama/Desktop/test (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   index.html
```

ملحوظة : يمكننا إضافة أكثر من ملف واحد إلى حالة الإدراج دفعة واحدة من خلال الأمر git add --all حيث أنّ الوسيط all - يعني جميع الملفات. كما يمكننا إضافة مجلد كامل إلى حالة الإدراج من خلال كتابة اسم المجلد بدلاً من اسم الملف عند تنفيذ تعليمة add

- مرحلة الادراج في Git
- إيداع المساهمات في Git
- إيداع نفس الملف مرة أخرى
- الاطلاع على سجل الايداعات
- التراجع عن عملية إيداع في Git



اساسيات Git



يعتمد git في إدارة الإصدارات على مفهوم "المساهمات" المساهمة commit و هي مجموعة من التعديلات التي تتم على الملفات و التي تحفظ ككتلة واحدة في شجرة تاريخ git، بحيث تعتبر كل محطات سير المشروع و يتيح git العودة إلى أية محطة مستقبلاً كما مساهمة commit محطة سنرى.

على أية حالة، لإيداع مساهمة يجب أولاً إدراج جميع التغييرات المطلوب إيداعها، ثم و بعد الإدراج Staging يتم الإيداع في المستودع من خلال الأمر commit مع مراعاة ترك رسالة مناسبة ترافق عملية الإيداع و تبين ما تم فيها بشكل مختصر عادة لسهولة المراجعة من قبل المساهم نفسه أو زملائه في فريق العمل.

- ١٠ مرحلة الادراج في Git
- ١٠ إيداع المساهمات في Git
- ١٠ إيداع نفس الملف مرة اخرى
- ١٠ الاطلاع على سجل الايداعات
- ١٠ التراجع عن عملية إيداع في Git



اساسيات Git



لإيداع العمل الذي قمنا به في الفقرة السابقة مع رسالة تفيد أنّه أول إيداع نقوم بتنفيذ الأمر التالي:

```
git commit -m "c-one"
```

ينفذ الأمر commit عملية الإيداع ، وتضيف "message" -m رسالة و هكذا تكون بيئة الإدراج قد أودعت في مستودعنا مع رسالة "c-one"، الإيداع الأول" كما يظهر في الصورة أدناه و يلاحظ أن git يقوم تلقائياً بمنح ترقيم فريد لكل عملية إيداع، و يسمّى هذا الرقم معرف الإيداع. Commit ID.

```
A.Salama@DESKTOP-409U06R MINGW64 /c/users/A.salama/Desktop/test (master)
$ git commit -m "c-one"
[master (root-commit) 7c3eb50] c-one
1 file changed, 11 insertions(+)
create mode 100644 index.html
```

- ❖ مرحلة الادراج في Git
- ❖ إيداع المساهمات في Git
- ❖ إيداع نفس الملف مرة أخرى
- ❖ الاطلاع على سجل الايداعات
- ❖ التراجع عن عملية إيداع في Git



اساسيات Git



الآن بعد أن قمنا بإيداع الملف للمرة الأولى، فلنقم بعمل أي تغيير ضمن الملف index.html كإضافة سطر مثلاً أو تغيير محتوى الجملة النصية و ذلك بهدف رؤية كيف سيقوم git بالتعرف على التغيير الحاصل عند تنفيذ الأمر git status كما توضحه الصورة أدناه،
و يرجى ملاحظة كلمة modified و التي تعني أن الملف مدرج سابقاً لكنه الآن خضع لتعديل جعل نسخته الحالية معدلة modified بمعنى أنها مختلفة عن آخر إدراج تم لها في git

```
A.Salama@DESKTOP-409U06R MINGW64 /c/users/A.salama/Desktop/test (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html
```

- ١٠ مرحلة الادراج في Git
- ١٠ إيداع المساهمات في Git
- ١٠ إيداع نفس الملف مرة أخرى
- ١٠ الاطلاع على سجل الايداعات
- ١٠ التراجع عن عملية إيداع في Git



اساسيات Git



يمكننا الآن تكرار نفس خطوات عملية الإيداع السابقة، و إجراء إيداع ثان لنفس الملف مع رسالة تفيد بسبب الإيداع كما تبينه الصورة التالية

```
A.Salama@DESKTOP-409U06R MINGW64 /c/users/A.salama/Desktop/test (master)
$ git add index.html

A.Salama@DESKTOP-409U06R MINGW64 /c/users/A.salama/Desktop/test (master)
$ git commit -m "c-tow"
[master 42b4f41] c-tow
1 file changed, 1 insertion(+)
```

- مرحلة الادراج في Git
- إيداع المساهمات في Git
- إيداع نفس الملف مرة أخرى
- الاطلاع على سجل الايداعات
- التراجع عن عملية إيداع في Git



اساسيات Git



الآن و كوننا قمنا بعمل إيداعين في مستودعنا، ربما يكون الوقت مناسباً لتجربة أمر الاطلاع على الإيداعات التي تمت بالفعل في المستودع من خلال تنفيذ الأمر:

git log

و الذي يعرض معرّفات عمليات الإيداع السابقة و تاريخ كل منها و الرسالة التي تبين سبب الإيداع كما أدخلها المستخدم الذي قام بعملية الإيداع كما يظهر في الصورة أدناه مع ملاحظة أن ترتيب ظهور عمليات الإيداع عكسي من الأحدث للأقدم

```
A.Salama@DESKTOP-409U06R MINGW64 /c/users/A.salama/Desktop/test (master)
$ git log
commit 42b4f4178088c6751416365f3e9650c46ba2fa8e (HEAD -> master)
Author: ahmadSalameh1995 <a.salama998877@gmail.com>
Date:   Mon Jul 31 18:53:38 2023 +0300

    c-tow

commit 7c3eb50cce49ff1202077b94cba1cb4c09111fca
Author: ahmadSalameh1995 <a.salama998877@gmail.com>
Date:   Mon Jul 31 18:39:39 2023 +0300

    c-one
```

- ❖ مرحلة الادراج في Git
- ❖ إيداع المساهمات في Git
- ❖ إيداع نفس الملف مرة أخرى
- ❖ الاطلاع على سجل الايداعات
- ❖ التراجع عن عملية إيداع في Git



اساسيات Git



التراجع عن عملية إيداع في git يحدث أحياناً أن نقوم بعملية إيداع ما أو أكثر، ثم و لسبب أو لآخر نكتشف أننا بحاجة للتراجع عن عملية الإيداع تلك، تتيح git التراجع عن عمليات الإيداع بأسلوبين اثنين أحدهما بسيط و سنناقشه في هذه الفقرة، والآخر متقدم سنناقشه في الفصل الرابع بإذن الله تعالى

أما الأمر البسيط فهو أمر git revert و الذي يقوم بالتراجع عن عملية إيداع معينة من خلال القيام بعملية إيداع أخرى معاكسة تماماً بحيث تكون النتيجة العودة بالملفات إلى حالتها التي كانت عليها قبل عملية الإيداع الأساسية التي نرغب بالتراجع عنها

- ١٠ مرحلة الادراج في Git
- ١٠ إيداع المساهمات في Git
- ١٠ إيداع نفس الملف مرة أخرى
- ١٠ الاطلاع على سجل الايداعات
- ١٠ التراجع عن عملية إيداع في Git



اساسيات Git

لكي نفهم الأمر بشكل عملي ، لنقم بالتراجع عن عملية الإيداع الثانية التي قمنا بها في الفقرة السابقة و التي قمنا بموجها بإضافة بعض الكلمات إلى ملف html الذي أنشأناه، و للقيام بهذا الغرض نحتاج لمعرفة معرّف عملية الإيداع Commit ID المطلوب التراجع عنها و الذي يمكننا ببساطة نسخه مما يولده الأمر git log كما تظهر الصورة أدناه

```
A.Salama@DESKTOP-409U06R MINGW64 /c/users/A.salama/Desktop/test (master)
$ git log
commit 42b4f4178088c6751416365f3e9650c46ba2fa8e (HEAD -> master)
Author: ahmadSalameh1995 <a.salama998877@gmail.com>
Date: Mon Jul 31 18:53:38 2023 +0300

    c-tow

commit 7c3eb50cce49ff1202077b94cba1cb4c09111fca
Author: ahmadSalameh1995 <a.salama998877@gmail.com>
Date: Mon Jul 31 18:39:39 2023 +0300

    c-one
```

- ❖ مرحلة الادراج في Git
- ❖ إيداع المساهمات في Git
- ❖ إيداع نفس الملف مرة أخرى
- ❖ الاطلاع على سجل الايداعات
- ❖ التراجع عن عملية إيداع في Git



اساسيات Git

الملف قبل عملية التراجع عن الإيداع كما هو موضح بالصورة ادناه

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport">
  <title>Document</title>
</head>
<body>
  <h1>Hello Git</h1>
  <h1>Hello Github</h1>
  <h1>Hello Github</h1>
</body>
</html>
```

You, 3 minutes ago (August 1st, 2023 5:03 PM)

Uncommitted changes

Working Tree < ? ?

-

+ <h1>Hello Github</h1>

Changes ? dc8c29d ↔ Working Tree | ?

You, 52 seconds ago • Uncommitted changes

ثم تنفيذ الأمر التالي مع مراعاة استبدال < commit-id > بمعرف عملية الإيداع :

git revert <commit-id>

git log

- مرحلة الادراج في Git
- إيداع المساهمات في Git
- إيداع نفس الملف مرة أخرى
- الاطلاع على سجل الايداعات
- التراجع عن عملية إيداع في Git



اساسيات Git



الملف قبل بعد التراجع عن الإيداع كما هو موضح بالصورة ادناه

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device
  <title>Document</title>
</head>
<body>
  <h1>Hello Git</h1>
  <h1>Hello GitHub</h1>
```

- مرحلة الادراج في Git
- إيداع المساهمات في Git
- إيداع نفس الملف مرة أخرى
- الاطلاع على سجل الايداعات
- التراجع عن عملية إيداع في Git



اساسيات Git



فائدة : git ذكي كفاية لمعرفة عملية الإيداع المقصودة بمجرد إدخال أوّل 8 أحرف من معرف العملية إذا كنت لا ترغب بنسخ كامل المعرّف. وهذا ينطبق على جميع الأوامر التي تستقبل commit ID و منها الأمر revert

- ❖ مرحلة الادراج في Git
- ❖ إيداع المساهمات في Git
- ❖ إيداع نفس الملف مرة أخرى
- ❖ الاطلاع على سجل الايداعات
- ❖ التراجع عن عملية إيداع في Git