



# Git + GitHub

اعداد :

م. أحمد حاج محمد سلامة

الجامعة الدولية للعلوم والنهضة



# التفرعات Branches في Git



نستخدم في العمل ضمن الفرق التقنية التفرعات (Branches) في نظام Git لتمكين كل عضو أو مجموعة من أعضاء الفريق من تطوير جزء منفصل من المشروع بشكل مستقل، دون التأثير على باقي الفريق. يعمل كل مطور على نسخة شبه مستقلة (فرع) من المشروع، حيث يمكنه أخذ الوقت الكافي للتطوير بدون التأثير على الفرع الرئيسي. عند اكتمال التطوير في الفرع الجديد، يتم دمج مع المشروع الرئيسي. يُمكن التبديل بين الفروع والعمل على مشاريع مختلفة دون التأثير على بعضها البعض، مما يجعل استخدام التفرع طريقة سريعة وفعالة لإجراء التعديلات على المشروع.

ما هي التفرعات في Git

إنشاء فرع جديد في Git

الانتقال الى فرع اخر في Git

فرع الإصلاحات الطارئة

دمج فروع في Git

حذف فرع في Git

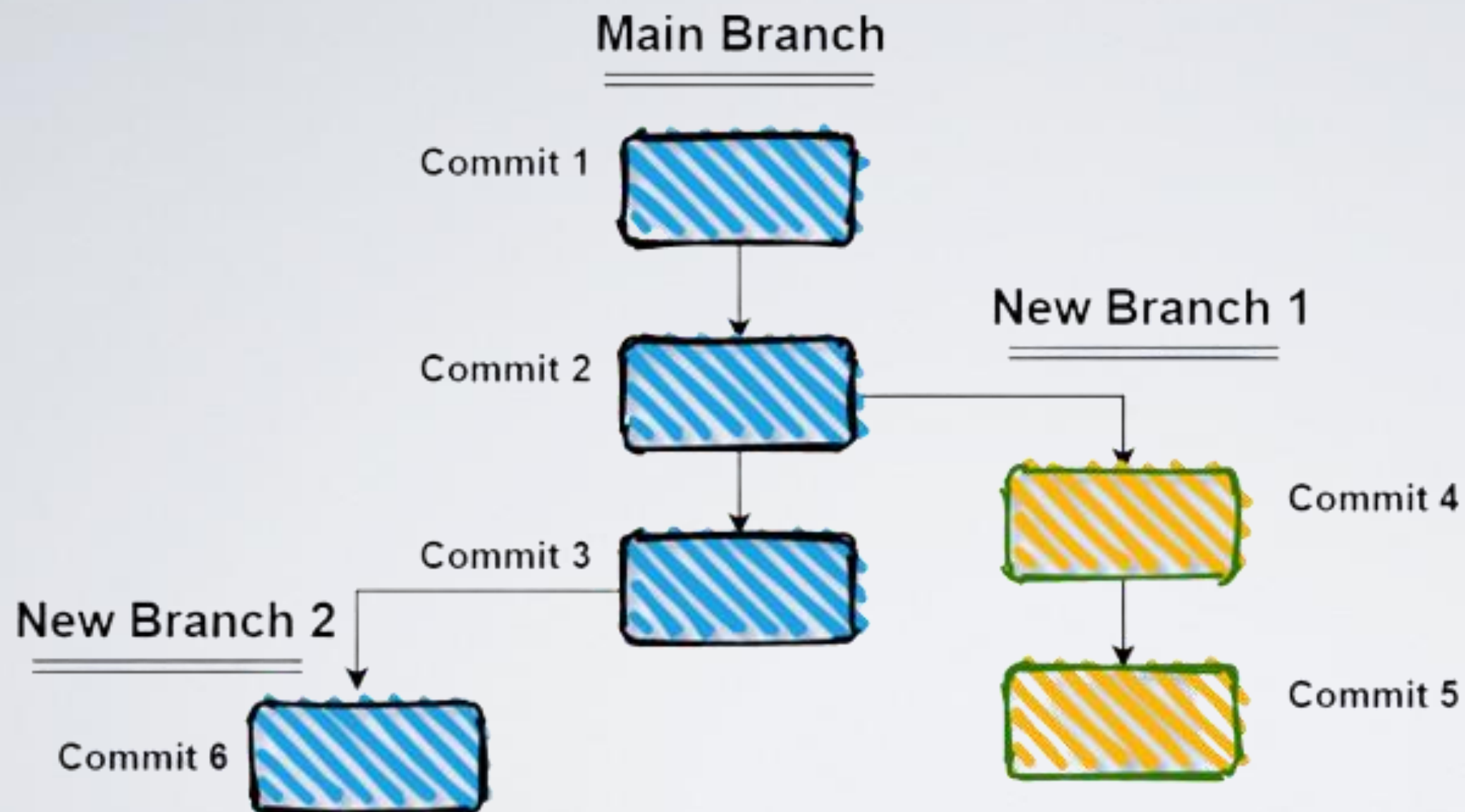
Merge conflicts

Conflict Resolve

تثبيت أداة gitk



# التفرعات Branches في Git



ما هي التفرعات في Git

إنشاء فرع جديد في Git

الانتقال الى فرع اخر في Git

فرع الإصلاحات الطارئة

دمج فروع في Git

حذف فرع في Git

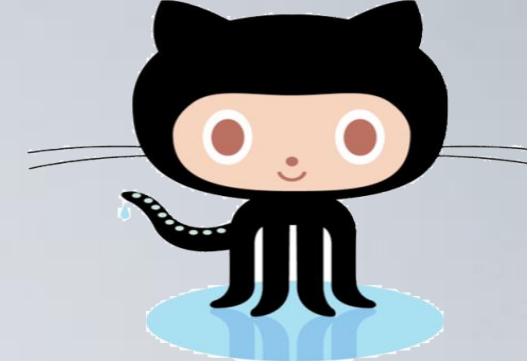
Merge conflicts

Conflict Resolve

تثبيت أداة gitk



# التفرعات Branches في Git



لنصف الآن بعض المزايا الجديدة في صفحة index.html التي أنشأناها في الفصل السابق، و كوننا نرغب أن نقوم بهذا في مستودعنا المحلي دون التأثير على المشروع الرئيسي أو تعطيله سنقوم بإنشاء فرع جديد باستخدام الأمر

```
git branch hello-world-images
```

قد أنشأنا الآن فرعاً جديداً باسم branch hello-world-again، و يمكننا أن نتحقق من وجوده من خلال تنفيذ الأمر branch دون تمرير اسم أي فرع مرة أخرى

```
A. Salama@DESKTOP-409U06R MINGW64 ~/Desktop/test2 (master)
$ git branch hello-word-images

A. Salama@DESKTOP-409U06R MINGW64 ~/Desktop/test2 (master)
$ git branch
hello-word-images
* master
```

ما هي التفرعات في Git

إنشاء فرع جديد في Git

الانتقال الى فرع اخر في Git

فرع الإصلاحات الطارئة

دمج فروع في Git

حذف فرع في Git

Merge conflicts

Conflict Resolve

تثبيت أداة gitk





# التفرعات Branches في Git



وكما نرى فيوجد الآن فرع جديد باسم hello-world-images و فرع آخر اسمه master بجانبه علامة نجمة \* تعني أننا الآن في هذا الفرع في الفرع master و ربما تتساءل الآن من أين جاء الفرع master و أنا أتعلم التفرعات للتو ؟ والجواب أن git تنشئ الفرع master بشكل آلي في كل مستودع جديد.

**فائدة:** بعض توزيعات git تستخدم الاسم main بدلاً من master لتلافي الإيحاء التاريخي الذي يعنيه لفظ السيد master في بعض الثقافات الغربية.

**تنبيه:** يجب الانتباه إلى الفرع الذي تتواجد فيه لحظة إنشاء فرع جديد، لأن الفرع الجديد سيكون في لحظة إنشائه نسخة مطابقة تماماً للفرع الذي تم إنشاؤه منه.

ما هي التفرعات في Git

إنشاء فرع جديد في Git

الانتقال إلى فرع آخر في Git

فرع الإصلاحات الطارئة

دمج فروع في Git

حذف فرع في Git

Merge conflicts

Conflict Resolve

تثبيت أداة gitk



# التفرعات Branches في Git



يُستخدم الأمر checkout للانتقال من الفرع الحالي إلى فرع آخر نمّر اسمه مع الأمر، و لتجربة ذلك يمكننا الانتقال إلى الفرع الجديد الذي أنشأناه توا كما يلي:

```
A.Salama@DESKTOP-409U06R MINGW64 ~/Desktop/test2 (master)
$ git checkout hello-word-images
Switched to branch 'hello-word-images'
```

و هكذا نكون قد نقلنا مساحة العمل الحالية من الفرع الرئيسي إلى الفرع الجديد، و الآن لنفتح محرراً نصياً لنجري بعض التعديلات.

ما هي التفرعات في Git

إنشاء فرع جديد في Git

الانتقال الى فرع اخر في Git

فرع الإصلاحات الطارئة

دمج فروع في Git

حذف فرع في Git

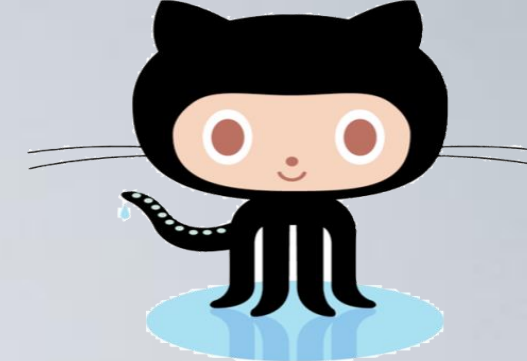
Merge conflicts

Conflict Resolve

تثبيت أداة gitk



# التفرعات Branches في Git



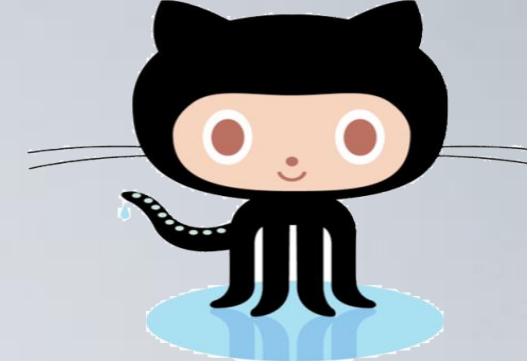
**فائدة:** يمكننا استخدام الخيار -b مع الأمر checkout لنطلب من git أن ينتقل إلى الفرع الجديد في حال كان موجوداً، وإلا أن ينشئه ثم ينتقل إليه مباشرة.  
في هذا المثال سنقوم بإضافة صورة الى مجلد العمل ثم نضمها ضمن صفحة index.html بإضافة السطر التالي:

```
<body>
  <h1>Hello Git</h1>
  <h1>Hello Github</h1>
  <div></div>
</body>
```

- ما هي التفرعات في Git
- إنشاء فرع جديد في Git
- الانتقال الى فرع اخر في Git
- فرع الإصلاحات الطارئة
- دمج فروع في Git
- حذف فرع في Git
- Merge conflicts
- Conflict Resolve
- تثبيت أداة gitk



# التفرعات Branches في Git



وبهذا نكون قد أجرينا تغييرات على ملف index.html و أضفنا الملف الجديد إلى مجلد العمل،  
و إذا نفذنا git status سنجد ما يلي:

```
A.Salama@DESKTOP-409U06R MINGW64 ~/Desktop/test2 (hello-word-images)
$ git status
On branch hello-word-images
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

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

وكما تلاحظ فالملف index.html يحوي بعض التعديلات (modified) غير أنه لم يُدرج بعد  
Not staged، و الصورة فحالته Untracked كونه يضاف للمرة الأولى و لم يودع من قبل.

ما هي التفرعات في Git

إنشاء فرع جديد في Git

الانتقال الى فرع اخر في Git

فرع الإصلاحات الطارئة

دمج فروع في Git

حذف فرع في Git

Merge conflicts

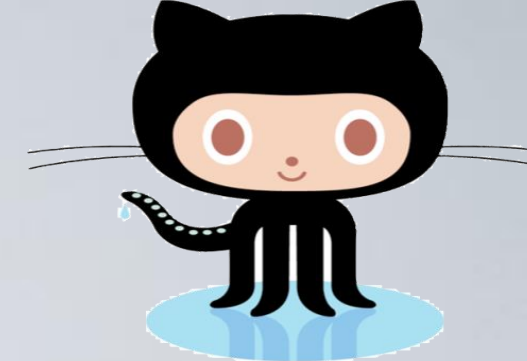
Conflict Resolve

تثبيت أداة gitk





# التفرعات Branches في Git



و للقيام بإيداع كلا التغييرين سنحتاج إلى إضافة كلا الملفين إلى بيئة الإدراج ننفذ الأمر التالي:  
`git add --all`

```
A.Salama@DESKTOP-409U06R MINGW64 ~/Desktop/test2 (hello-word-images)
$ git add --all

A.Salama@DESKTOP-409U06R MINGW64 ~/Desktop/test2 (hello-word-images)
$ git status
On branch hello-word-images
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   image.jpg
        modified:   index.html
```

ما هي التفرعات في Git

إنشاء فرع جديد في Git

الانتقال إلى فرع آخر في Git

فرع الإصلاحات الطارئة

دمج فروع في Git

حذف فرع في Git

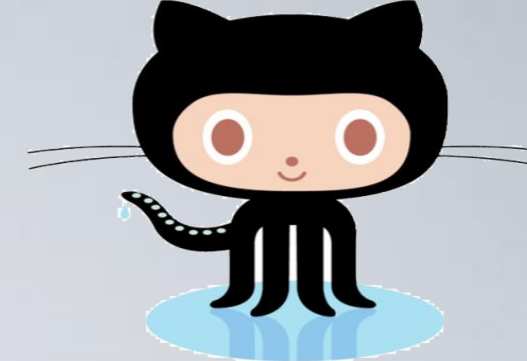
Merge conflicts

Conflict Resolve

تثبيت أداة gitk



# التفرعات Branches في Git



إيداع هذه التعديلات يمكننا إيداعها باستخدام الأمر git commit الذي تعلمناه سابقاً مع ترك رسالة مناسبة توضح الهدف من عملية الإيداع:

```
git commit -m "Added image to index.html"
```

```
A.Salama@DESKTOP-409U06R MINGW64 ~/Desktop/test2 (hello-word-images)
$ git commit -m "Added image to index.html"
[hello-word-images bf334b1] Added image to index.html
3 files changed, 2 insertions(+), 23 deletions(-)
create mode 100644 image.jpg
delete mode 100644 ssd
```

**ملاحظة:** لإعادة تسمية الفرع باسم جديد نستخدم الأمر التالي:

```
git branch -m newbranchName
```

ما هي التفرعات في Git

إنشاء فرع جديد في Git

الانتقال الى فرع اخر في Git

فرع الإصلاحات الطارئة

دمج فروع في Git

حذف فرع في Git

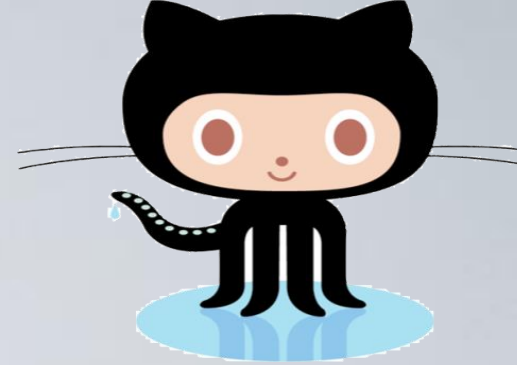
Merge conflicts

Conflict Resolve

تثبيت أداة gitk



# التفرعات Branches في Git



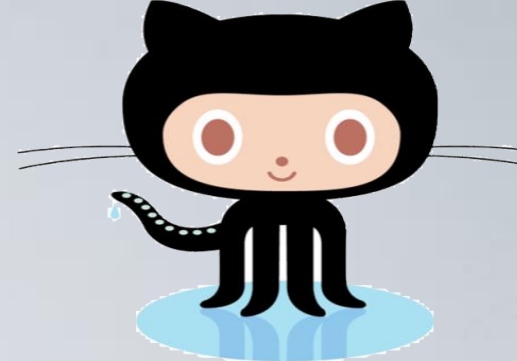
ما يهمنا الآن و نريد التركيز عليه هو وجود فرع branch جديد يختلف عن الفرع الرئيسي master و الاختلاف يمكن في عملية الإيداع الأخيرة، و لنرى الآن مدى سهولة العمل مع الفروع المختلفة ومدى كفاءتها في العمل. نحن الآن في الفرع hello-world-images، وأضافنا صورة إلى ذلك الفرع، و إذا استعرضنا الملفات التي في المجلد الحالي سنجد الملفين index.html و imges.jpg، بمعنى أننا نستطيع الآن رؤية الملف الجديد imges.jpg ، وكذلك نستطيع رؤية التغييرات الحادثة في الشيفرة إذا فتحنا ملف index.html، وهو ما نتوقعه تماماً. والآن، لنرى ما يحدث عند تغيير الفرع إلى master

**git checkout master**

- ما هي التفرعات في Git
- إنشاء فرع جديد في Git
- الانتقال الى فرع اخر في Git
- فرع الإصلاحات الطارئة
- دمج فروع في Git
- حذف فرع في Git
- Merge conflicts
- Conflict Resolve
- تثبيت أداة gitk



# التفرعات Branches في Git



لم تعد الصورة الجديدة جزءاً من هذا الفرع، و لو استعرضنا الملفات الموجودة في المجلد الحالي مرة أخرى لرأينا الآن أن الصورة الجديدة غير موجودة و كذلك الشيفرة التي أضفناها إلى index.html و هذا يرينا سهولة العمل مع الفروع وكيف أنها تسمح بالعمل على أشياء مختلفة في نفس الوقت!

ما هي التفرعات في Git

إنشاء فرع جديد في Git

الانتقال الى فرع اخر في Git

فرع الإصلاحات الطارئة

دمج فروع في Git

حذف فرع في Git

Merge conflicts

Conflict Resolve

تثبيت أداة gitk



# التفرعات Branches في Git



لنتخيل الآن أننا لم ننه العمل على الفرع hello-world-images، لكننا و لسبب هام نريد إصلاح خطأ طارئ في الفرع الرئيسي master دون المساس بذلك الفرع الرئيسي ولا فرع hello world-images بما أننا لم ننه العمل عليه بعد، في تلك الحالة يمكننا ببساطة إنشاء فرع جديد للتعامل مع هذا الإصلاح الطارئ و يمكننا تسميته hot-fix اصطلاحاً

## git checkout -b hot-fix

و نكون الآن قد أنشأنا فرعاً جديداً من الفرع الرئيسي و أجرينا تعديلات عليه، و نستطيع الآن إصلاح الخطأ دون التأثير على الفروع الأخرى، و كما تتوقع سيتم الأمر بإصلاح الأخطاء ثم إيداع هذا الإصلاح في الفرع الجديد، و لكن كيف نجلب هذا الإصلاح إلى الفرع الرئيسي؟ الجواب من خلال دمج الفرعين و هو ما سنناقشه في الفقرات التالية.

- ما هي التفرعات في Git
- إنشاء فرع جديد في Git
- الانتقال الى فرع اخر في Git
- فرع الإصلاحات الطارئة
- دمج فروع في Git
- حذف فرع في Git
- Merge conflicts
- Conflict Resolve
- تثبيت أداة gitk





# التفرعات Branches في Git



على فرض أن الخطأ الذي أصلحنه في الدرس السابق أصبح جاهزاً للدمج، يمكننا الآن أن نقوم بدمج الفرع hot-fix مع الفرع master باتباع الخطوات التاليتين:

. الانتقال إلى الفرع الهدف (master) الذي نرغب بدمج الفرع الآخر معه من خلال تعليمة  
**git checkout master**

دمج الفرع الآخر hot-fix مع الفرع الهدف من خلال الأمر

**git merge hot-fix**

في حال كون كل شيء على ما يرام سيتم الدمج وستظهر المساهمات التي قمنا بها في فرع الإصلاحات العاجلة ضمن الفرع الرئيسي كما تبين الصورة أدناه.

ما هي التفرعات في Git

إنشاء فرع جديد في Git

الانتقال إلى فرع آخر في Git

فرع الإصلاحات الطارئة

دمج فروع في Git

حذف فرع في Git

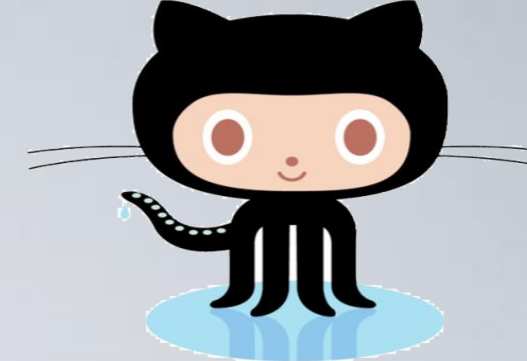
Merge conflicts

Conflict Resolve

تثبيت أداة gitk



# التفرعات Branches في Git



في حال لم نعد بحاجة إلى فرع ما لأي سبب كان فيمكننا حذفه من خلال الأمر

**git branch -D branch\_name**

حيث يمثل branch\_name اسم الفرع المطلوب حذفه، ويراعى أن يكون الحرف الوسيط -D بالحالة الكبيرة، كما تبين الصورة أدناه.

```
A.Salama@DESKTOP-409U06R MINGW64 ~/Desktop/test2 (master)
$ git branch -D hot-fix
Deleted branch hot-fix (was 8d81017).
```

**ملاحظة:** لحذف الفرع بشكل آمن، إذا تم دمج جميع التغييرات فيه نستخدم الامر.

`git branch -d branch_name``

ليحذف الفرع بشكل قوي دون التحقق من دمج التغييرات نستخدم الامر.

`git branch -D branch_name`` :

ما هي التفرعات في Git

إنشاء فرع جديد في Git

الانتقال الى فرع اخر في Git

فرع الإصلاحات الطارئة

دمج فروع في Git

حذف فرع في Git

Merge conflicts

Conflict Resolve

تثبيت أداة gitk



# التفرعات Branches في Git

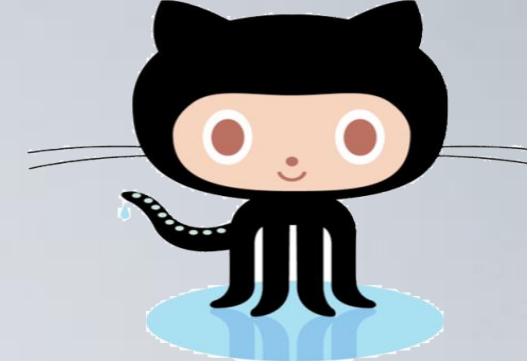


في الفقرة قبل السابقة، تمت عملية دمج الفرعين بشكل سلس و دون مشاكل و ذلك نظراً لعدم وجود أكثر من شخص يعملون على نفس الملفات في الوقت نفسه، لكن أحياناً و عند العمل ضمن فريق، لا تكون الأمور بهذه البساطة، و لا يستطيع git أحياناً معرفة الطريقة المثلى لدمج مساهمات المساهمين عندما يتصادف وجود تعديلين أو أكثر في نفس الملف و في نفس الأسطر، عندها يحدث ما يسمى بتعارض الدمج Merge conflict و الذي يتطلب التدخل اليدوي من أحد المساهمين للقيام بعملية حلّ التعارض Conflict resolvent بحيث يمكن إتمام الدمج بسلاسة. و لشرح هذه الحالة لا بدّ من اصطناع حالة تستوجب التعارض أولاً، ثم سنرى معاً كيف يحدث التعارض، و الأهم كيف يمكن حله!

- ما هي التفرعات في Git
- إنشاء فرع جديد في Git
- الانتقال الى فرع اخر في Git
- فرع الإصلاحات الطارئة
- دمج فروع في Git
- حذف فرع في Git
- Merge conflicts
- Conflict Resolve
- تثبيت أداة gitk



# التفرعات Branches في Git



حسناً، لاختلاق حالة التعارض قم بعمل فرع جديد ثم عدّل السطر رقم 5 مثلاً من ملف index.html في نفس الفرع الحالي، و قم بحفظ المساهمة، ثمّ انتقل إلى الفرع الجديد و قم بتعديل مختلف في نفس السطر و في نفس الملف السطر من الملف (index.html) ثم قم بحفظ المساهمة. الآن عندما تعود إلى الفرع الرئيسي و تحاول دمج الفرع الجديد سيظهر التعارض و سترى كلمة Conflict أمام الملفات المتعارضة كما تبينّه الصورة التالية:

```
A.Salama@DESKTOP-409U06R MINGW64 ~/Desktop/test2 (master)
$ git merge conf-y
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

- ما هي التفرعات في Git
- إنشاء فرع جديد في Git
- الانتقال الى فرع اخر في Git
- فرع الإصلاحات الطارئة
- دمج فروع في Git
- حذف فرع في Git
- Merge conflicts
- Conflict Resolve
- تثبيت أداة gitk



# التفرعات Branches في Git



حسناً كما رأينا في الفقرة السابقة فقد حصل تعارض عندما حاولنا دمج الفرعين معاً نتيجة لحقيقة أننا قمنا بإجراء تعديلات مختلفة في نفس الأماكن من نفس الملفات و بالتالي لن نستطيع git تحديد طريقة لدمج المساهمات معاً بشكل آلي في الملفات المتأثرة بالتعارض، يمكننا من خلال الأمر `git status` معرفة الملفات التي تحوي تعارضات حيث سيظهرها git و بجانبها وصف `both modified` بمعنى أنّ الملف خضع لتعديل في كلا الفرعين، كما يظهر في

```
A. Salama@DESKTOP-409U06R MINGW64 ~/Desktop/test2 (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

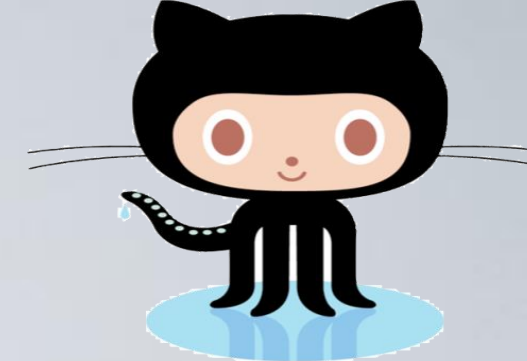
Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   index.html
```

- ما هي التفرعات في Git
- إنشاء فرع جديد في Git
- الانتقال الى فرع اخر في Git
- فرع الإصلاحات الطارئة
- دمج فروع في Git
- حذف فرع في Git
- Merge conflicts
- Conflict Resolve
- تثبيت أداة gitk





# التفرعات Branches في Git



لو قمنا الآن بفتح الملف المتأثر باستخدام المحرر النصي، سنجد أن git قام بوضع علامات تحدد مناطق التعارض ضمن الملف متوقعاً منا القيام بتعديلات يدوية على الملف لحل التعارضات، و من الجدير بالذكر أن بعض المحررات النصية المتقدمة تسهل عملية حل التعارضات البسيطة من خلال بعض الأدوات كما في حالة VS Code مثلاً كما يظهر في الصورة أدناه

```
<body>
|   <h1>hello git</h1>
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<< HEAD (Current Change)
|   <h1>hello github msa</h1>
=====
|   <h1>hello github asd</h1>
>>>>>> conf-y (Incoming Change)

</body>
```

- ما هي التفرعات في Git
- إنشاء فرع جديد في Git
- الانتقال الى فرع اخر في Git
- فرع الإصلاحات الطارئة
- دمج فروع في Git
- حذف فرع في Git
- Merge conflicts
- Conflict Resolve
- تثبيت أداة gitk



# التفرعات Branches في Git



يمكننا بوضوح رؤية أن git ترك التعديلين المتعارضين مع إشارة واضحة لاسم الفرع الذي جاء منه كل تعديل (head يعني الفرع الذي نحاول الدمج إليه master في مثالنا). والآن يمكننا إبقاء التعديل المطلوب و حذف التعديل غير المرغوب به ليعود الملف إلى حالة طبيعية، ثم نقوم بعد ذلك بعمل التعليمات **git add --all** للملفات التي تم تعديلها و حل التعارضات فيها، ثم التعليمة **git commit**

ما هي التفرعات في Git

إنشاء فرع جديد في Git

الانتقال الى فرع اخر في Git

فرع الإصلاحات الطارئة

دمج فروع في Git

حذف فرع في Git

Merge conflicts

Conflict Resolve

تثبيت أداة gitk



# التفرعات Branches في Git



بهذا نكون قد غطينا الأوامر الأساسية في git للتعامل مع المستودع محلياً على الحاسب الخاص بنا)، و سنقوم في الفصل القادم بشرح كيفية التعامل مع مستودع بعيد مستعرضين منصة GitHub كمثال على منصات git البعيدة، مع الإشارة إلى أنّ غالبية ما سنغطيه عند الحديث عن GitHub ينطبق على جميع المنصات الشهيرة مثل BitBucket أو Gitlab.



- ما هي التفرعات في Git
- إنشاء فرع جديد في Git
- الانتقال الى فرع اخر في Git
- فرع الإصلاحات الطارئة
- دمج فروع في Git
- حذف فرع في Git
- Merge conflicts
- Conflict Resolve
- أداة gitk