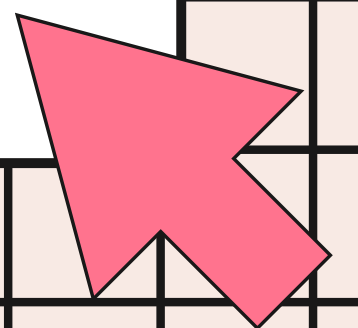
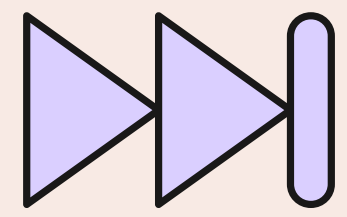


ما هي تقنية Git وما هو GitHub

تم إعداد هذا النسخة مجاناً ولا تباع
من @ALWALEEDALWABEL

الاصدار الاول - 2022 - كُتِبَ بكل فخر في المملكة العربية السعودية





— X

v2.6.0-beta.2 build: release 2.6.0-beta.2

build: build 2.6.0-beta.2

feat: dynamic directive arguments for v-on, v-bind and custom direct

origin/dynamic-directive-arguments feat: dynamic args for cust

perf: improve scoped slots change detection accuracy (#9371)

test: test cases for v-on/v-bind dynamic arguments

refactor: v-bind dynamic arguments use bind helper

test: fix tests, resolve helper conflict

fix: fix middle modifier

feat: handle dynamic argument for v-bind.sync

origin/slot-optimization perf: improve scoped slot

feat: dynamic directive arguments for v-bind and v-

refactor: extend dom-props update skip to more a

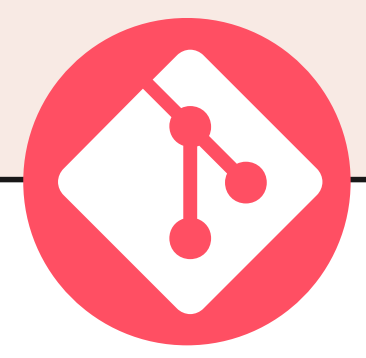
fix: fix checkbox event edge case in Firefox

test: fix tests in IE/Edge

S.md

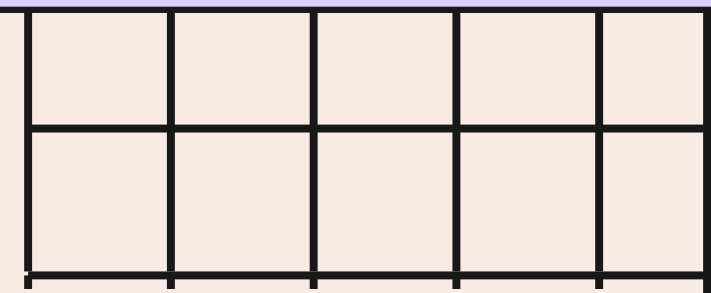
timestamp check

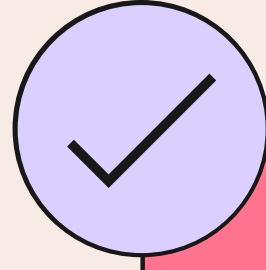
⏪ ⏩



ما هي تقنية Git؟

ما تاريخها؟

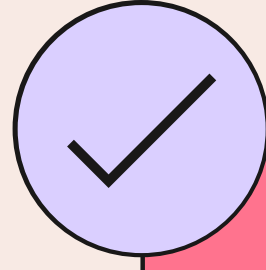




جيت / Git (برنامج)

جيت (بالإنجليزية: Git) هو برنامج حاسوب لإدارة الإصدارات (يُطلق على هذه العملية اسم "إدارة النُسخ" أيضا) وهو أحد برامج إدارة المراجعة الموزعة تمت برمجته ليكون سريعا في البداية قام لينوس تورفالدز ببرمجته ليستخدمه في تطوير نواة لينكس وليتجاوز عيوب ومشاكل برامج إدارة الإصدارات الأخرى، ثم انضم إليه لاحقا جونيو هامانو.

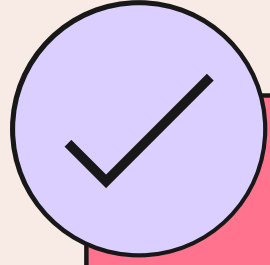
جيت برنامج حر ومفتوح المصدر يصدر تحت رخصة جنو العمومية الإصدار 2.



مثال وتاريخ

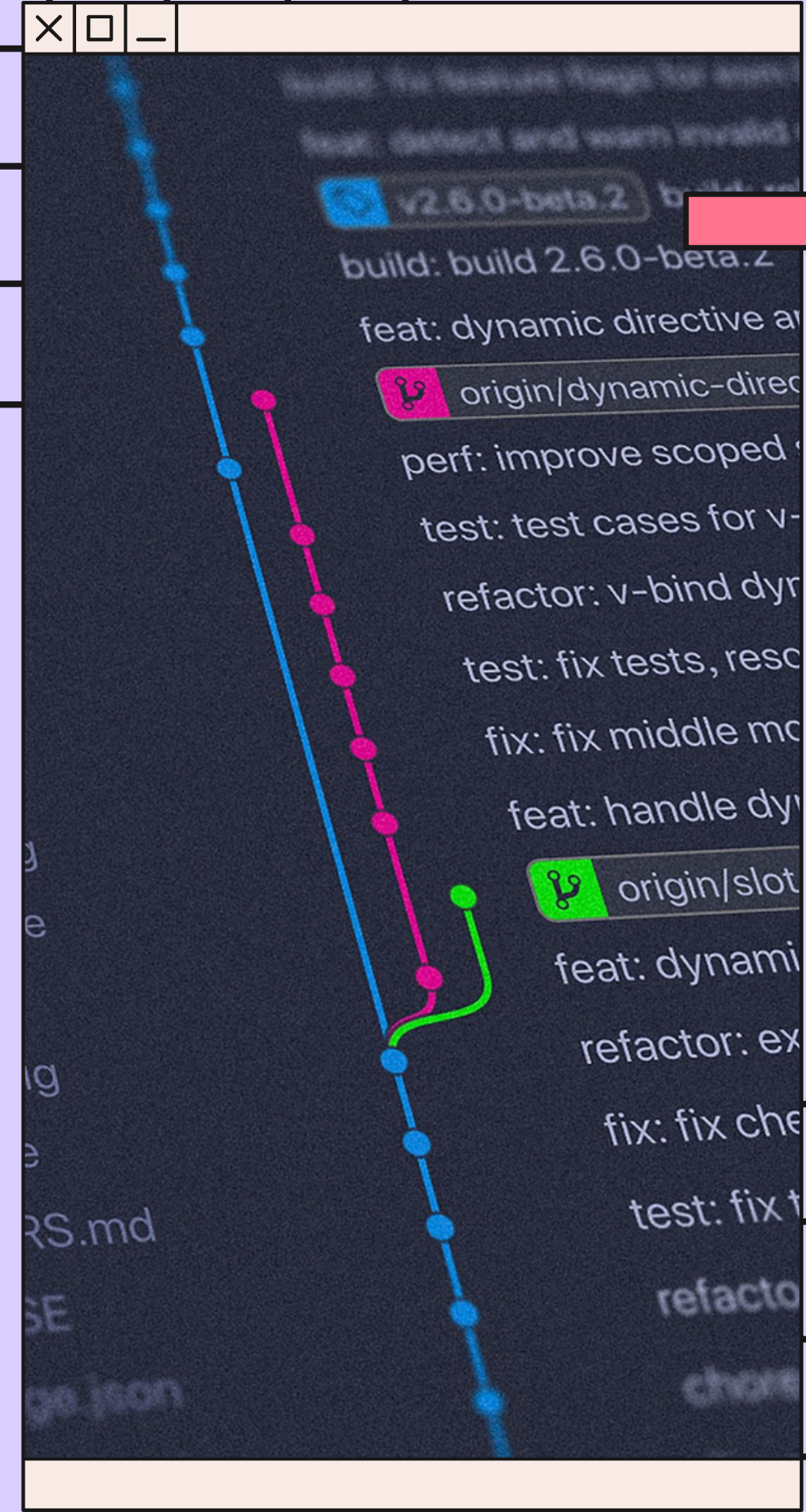
تخيل معي أنت قمت ببرمجة تطبيق او موقع وصديقك بجانبك او في منزلة وصديقك الآخر والذي يقع في الجهة المقابلة من الكره الأرضية ويود ايضا مساعدك والعمل على نفس الكود البرمجي لكنهم لا يمكنهم الإتصال بك كل يوم لكي يسألون هل قمت بعمل تعديل على الشيفرة المصدرية وإذا قمت بذلك فإنهم يودون أن تقوم بإرسال نسخة على البريد من نفس الكود الخاص بك

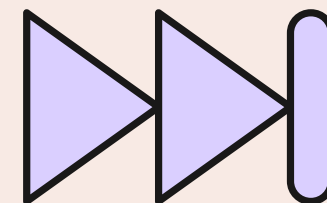
هذا الامر كان يحدث حتى عام 1990 عندما طُوّر ديك غرون نظام مركزي للتحكم في الإصدار وسماه نظام النسخ المتلاقية او Concurrent Versions System وإختصارها CVS



حقائق عن Git

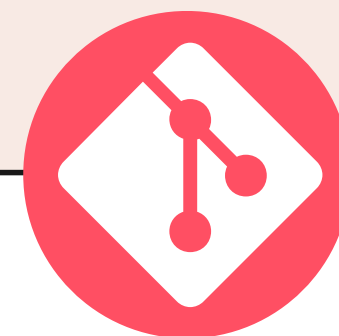
1. Git هو برنامج وليس لغة برمجة.
2. مفتوح المصدر ويطورة مبرمجين من جميع أنحاء العالم.
3. مطورين Git يتجاوز عددهم 1500 مطور حول العالم.
4. برنامج خفيف وسريع.





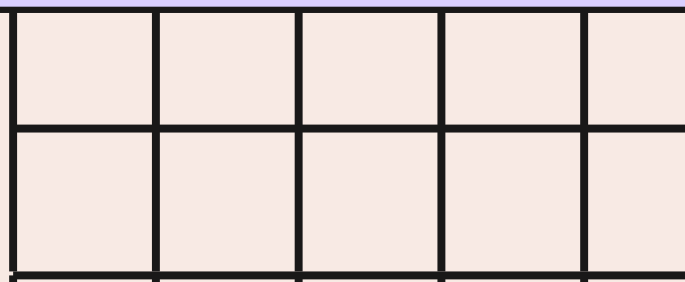
v2.6.0-beta.2 build: release 2.6.0-beta.2
build: build 2.6.0-beta.2
feat: dynamic directive arguments for v-on, v-bind and custom direct
origin/dynamic-directive-arguments feat: dynamic args for cust
perf: improve scoped slots change detection accuracy (#9371)
test: test cases for v-on/v-bind dynamic arguments
refactor: v-bind dynamic arguments use bind helper
test: fix tests, resolve helper conflict
fix: fix middle modifier
feat: handle dynamic argument for v-bind.sync
origin/slot-optimization perf: improve scoped slot
feat: dynamic directive arguments for v-bind and v-
refactor: extend dom-props update skip to more a
fix: fix checkbox event edge case in Firefox
test: fix tests in IE/Edge

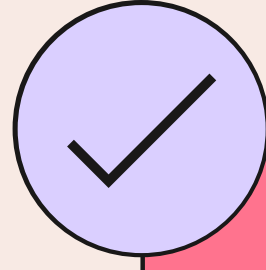
⏪ ⏩



كيف تثبيت Git؟

وكيف نبدأ بتعلمها؟

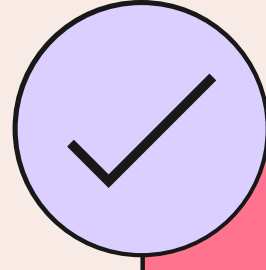




لكي تثبت Git تحتاج إلى

- تحديد نوع النظام الذي لديك (ويندوز، لينكس، ماك).
- زيارة موقع Git والذهاب إلى صفحة التحميل أنقر هنا للوصول السريع.
- النقر على زر التنزيل وإتباع باقي الخطوات مثل تثبيت أي برنامج عادي كل شيء Next.
- في ويندوز وقبل التحميل تأكد من تحميل نسخه (Standalone Installer).
- لتأكد من أنه تم تثبيت البرنامج أفتح CMD في عبر النقر على زر "إبدأ" و "ق" واكتب CMD هذا في الويندوز قم بتشغيل Git Bash.

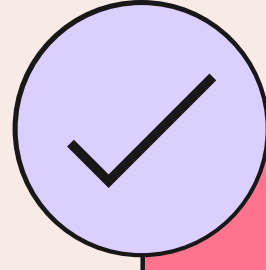




لكي تثبت Git تحتاج إلى

- بعد تشغيل Git Bash قم بكتابة الأمر `git --version` إذا ظهر لك كما في الصورة التالية فكل شيء على ما يرام وإذا ظهرت لديك مشكلة أخرى إبحث عنها في اليوتيوب وستجد شروحات كثيرة عنها.

```
MINGW64:/c/Users/الوليد
ALWALEED@MSI MINGW64 ~
$ git --version
git version 2.35.1.windows.2
ALWALEED@MSI MINGW64 ~
$ |
```

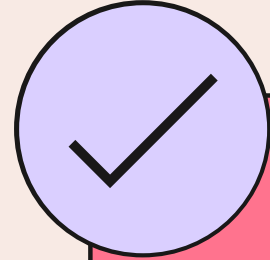


قبل أن تبدأ أكتب الأوامر التالية:

- تسمح الاوامر التاليه لك بتحديد من قام بالتعديل على أي ملف وكيف حدث التعديل سنشرحه بالتفاصيل لاحقاً

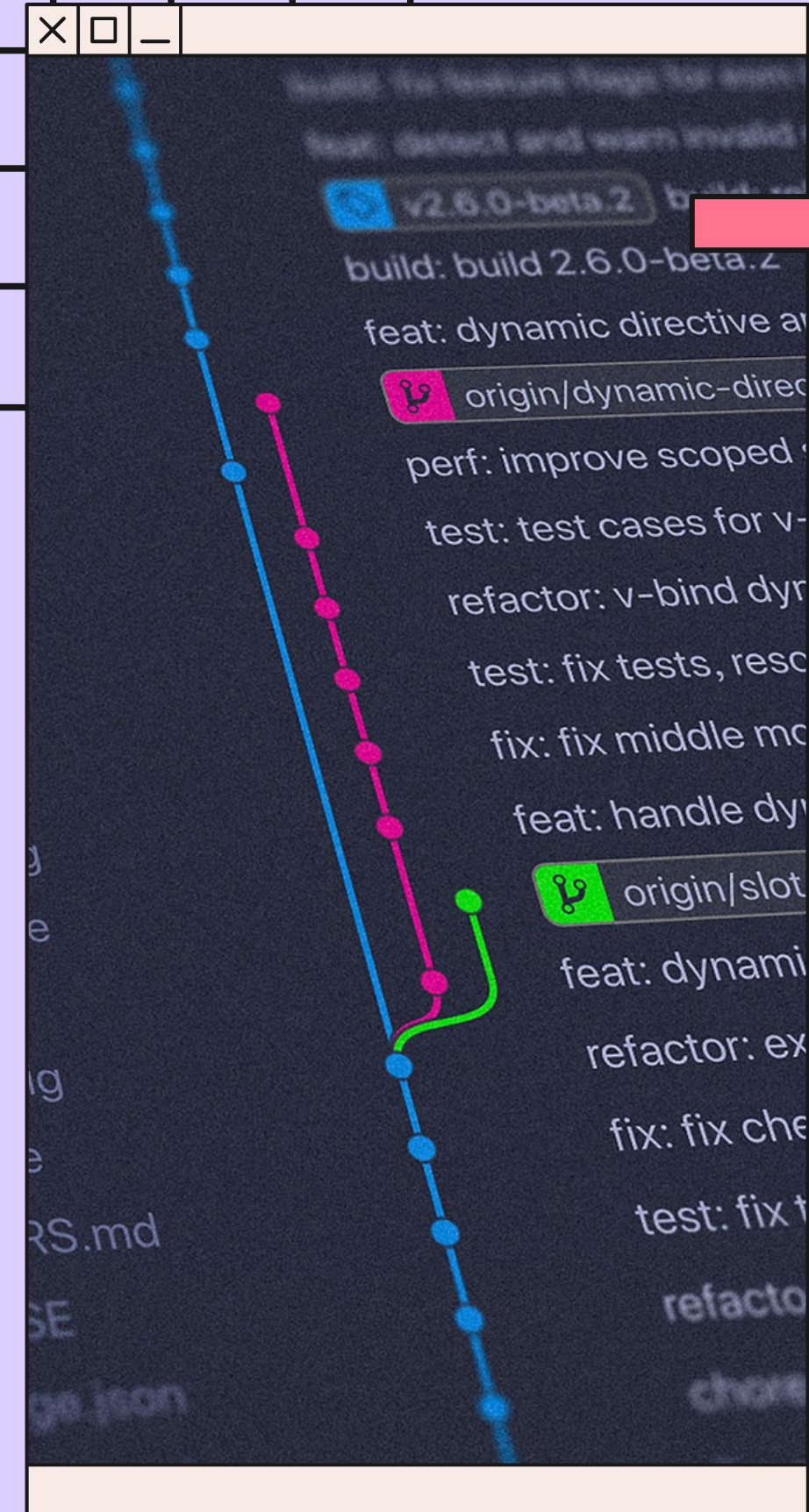
```
git config --global user.name "Your Name"  
git config --global user.email "your@email.com"
```

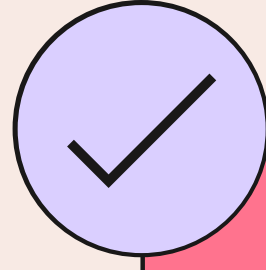
- قم فقط بتعديل الإسم وإزالة الأقواس واكتب اسمك ومن ثم إنتر ومن ثم قم بكتابة الأمر الذي يليه وعدل البريد إلى بريدك ايأً يكن أنت لن تتصل بأحد فقط لحفظ التغييرات بإسمك



المستودعات

- عند العمل مع Git ، من المهم أن تكون على دراية بالمصطلح ** repository ** وهو مستودع Git عبارة عن حاوية لمشروع يتم تعقبه بواسطة Git .
- يمكننا تحديد نوعين رئيسيين من مستودعات Git :
 - المستودع المحلي .
 - المستودع البعيد .





المزيد عن انواع المستودعات

يمكننا تحديد نوعين رئيسيين من مستودعات Git:

- المستودع المحلي - وهو مستودع معزول مخزن على جهاز الكمبيوتر الخاص بك ، حيث يمكنك العمل على الإصدار المحلي من مشروعك لوحده.
- المستودع البعيد - يتم تخزينه بشكل عام خارج نظامك المحلي المعزول ، عادةً على خادم بعيد. إنه مفيد بشكل خاص عند العمل في مجموعات - هذا هو المكان الذي يمكنك فيه مشاركة رمز مشروعك ، والاطلاع على كود الأشخاص الآخرين ودمجه في نسختك المحلية من المشروع ، وكذلك دفع تغييراتك إلى المستودع البعيد.

كيف نبدأ

قم بإنشاء مجلد على سطح المكتب او في اي مكان تفضله ومن ثم إنتقل إلى Git Bash واكتب الاوامر في اليمين بالترتيب للوصول للملف في حالتي إستخدمت الترمينل من اجل إنشاء ملف دون إنشاء ملف على سطح المكتب بالطريقة التقليدية وقمت بالتنقل للدخول إلى الملف بإستخدام أمر cd اما بالنسبة لأمر mkdir فهذا الامر ينشئ لك ملف جديد اكتب الامر متبوعاً بالإسم ومن ثم أنقر إنتر

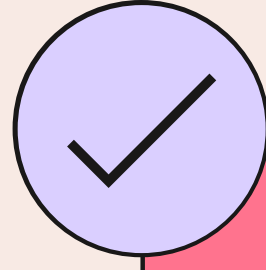
```
ALWALEED@MSI MINGW64 ~  
$ cd Desktop/  
  
ALWALEED@MSI MINGW64 ~/Desktop  
$ mkdir MyAppWithGit  
  
ALWALEED@MSI MINGW64 ~/Desktop  
$ cd MyAppWithGit  
  
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit  
$ |
```

تفعيل المستودع

لتفعيل المستودع المحلي على جهازك بعد الدخول للملف أكتب الامر التالي:
هذا `git init` سينشئ هذا الأمر دليلًا مخفيًا اسمه `git` لمشروعك ، حيث يخزن `Git` جميع بيانات التتبع الداخلية للمستودع الحالي.

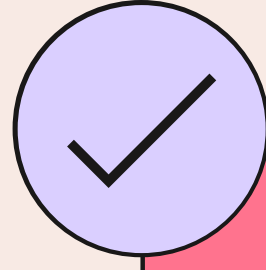
```
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit
$ git init
Initialized empty Git repository in C:/Users/      /Desktop/MyAppWithGit/.git/

ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$ |
```



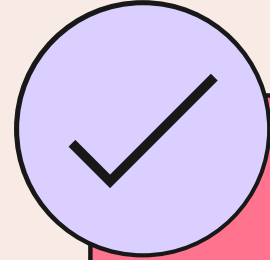
ماذا أستفيد من مدير الإصدارات هذا؟

- لنشرح الامر أكثر لنقل انك قمت ببرمجة مشروع وكان مشروعك يحتوي على سطر برمجي كتبته قبل أسبوع وكتبت تعليقات معه ولكنك نسيت ماذا كتبت حيث أنه قبل يوم قمت بحذف السطر البرمجي لكنك تود الوصول إلى الكود السابق فماذا تفعل مدير الإصدارات يسمح لك بتتبع التغييرات على كل ملف منذ إنشائه وحتى حذفه وما بعد ذلك وما بينهما لجعلك قادر على استعادته النسخ والفروقات بينها في اي وقت دون الحاجة للإبقاء على اكثر من ملف كل واحد منهم يحتوي على نسخه مختلفة او تعديلات بسيطة.
- لكن تذكر لا يمكنه تذكر تعديلاتك حتى تكتب أمر بعد كل تعديل.



إضافة إلى ماذا ستستفيد؟

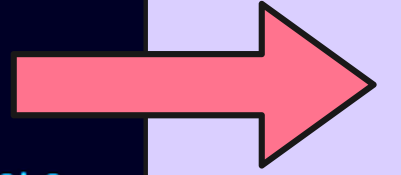
- إذا كنت غير مقتنع فقد تكون على حق حيث قد يكون الامر سهل أن تتبع ملف واحد لكن تخيل معي عشرات او مئات الملفات قمت بالتعديل والتغيير في كل واحد منهم سيصبح الامر كابوس بالنسبة لك هنا يتدخل برنامج مدير الإصدارات او في حالتنا git لكي يوفر عليك العناء والجهد للتتبع التغييرات في الملفات حيث يقوم بإنشاء ارشيف ضخمة من التعديلات البسيطة فمع كل تعديل بإضافة سطر او فاصلة ستلاحظ git ذلك على الفور.



أول commit

- سنقوم في هذا القسم سنقوم بأول commit لنا.

```
1  .box2 {
2    background-color: #2cd2b2;
3    height: 90px;
4    padding: 10px;
5    text-align: center;
6    border-radius: 19px 19px 0px
7    border: none;
8    width: 110%;
9    z-index: 2;
10 }
11 .box2 h1 {
12   font-size: 32px;
13   color: #000000;
14   font-family: Times New Roman;
15   font-weight: 400;
16 }
17 .box3 {
18   border: none 1px 1px 1px;
19   border-color: #cdcdcd;
20   border-style: solid;
21   background-color: #05152e;
22 }
23
```



قبل البدء

قم بالتعديل على المجلد الذي قمنا بإنشائه تحت إسم MyAppWithGit أو الإسم الذي إختارته سنقوم بإنشاء ملفين داخل هذا المجلد الاول بإسم index.html والثاني بإسم style.css وسنستخدم Git Bash وذلك عبر الامر touch ومتبوع بإسم الملف. صيغة الملف كما في الصورة

```
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$ touch index.html

ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$ touch style.css

ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$ .....
```

التأكد من حالة الملف

لتأكد من حالة الملف سنقوم
بكتابة الامر التالي:

```
git status
```

حيث سيسمح لنا بإستعراض
3 انواع حالات للملف وهي:

- ملف جديد.

- حذف ملف.

- تعديل على ملف سابق.

في حالتنا سيكون الخيار الاول
لأننا لم نقوم بإضافة هذا
الملف إلى git وكما تظهر
اسماء الملفات باللون الاحمر

```
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be commit
ted)
       index.html
       style.css

nothing added to commit but untracked files present (use "gi
t add" to track)

ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$ |
```

لنقم بتحديد الملفات

سنقوم أولاً بتحديد الملفات التي نود أن يتم إضافتها لكي نقوم بعمل commit ولعمل ذلك أكتب الامر التالي:
إسم الملف. الصيغة `git add` ويمكن أن تحدد أسماء الملفات مع وذلك عبر ترك مسافة بين كل ملف وآخر كما في الصورة أما إذا اردت تحديد الكل بدون كتابة إسم كل ملف `git add .`
النقطة تعني تحديد الكل

```
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$ git add index.html style.css
```

```
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$ |
```

التأكد من حالة الملف لمعرفة الحالة

بعد أن حددنا الملفات التي
سيتم إضافتها للـ `commit`
يمكن لنا مراجعتها عبر نفس
الامر السابق:

```
git status
```

وسنجد أن اللون تغير إلى
اللون الأخضر والحالة بأنه يوجد
تغييرات يجب عمل `commit`
لها ونوعها وفي حالتنا نوع
الحالة ملف جديد.

```
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$ git status
On branch master

No commits yet

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

ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$
```

اول commit

تذكر هذه اللحظة لانه سيكون شعور جميل أن تقوم بهذا الشيء لأول مره (: لنقم بذلك بالأمر التالي:

```
git commit -m "نص"
```

التعديل

الإم او -m ترمز إلى الرسالة
commit وهي النص الذي يلها
والذي بين علامتي التنصيص
فيه تكتب ماذا قمت بالتعديل
على الملفات

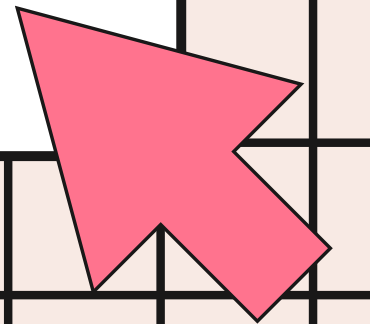
3 2 1
GO

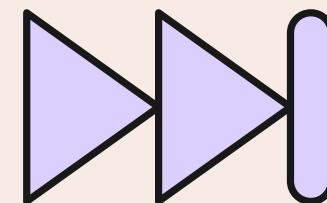
```
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$ git commit -m "first commit"
[master (root-commit) d3cc23b] first commit
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.html
create mode 100644 style.css

ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$
```

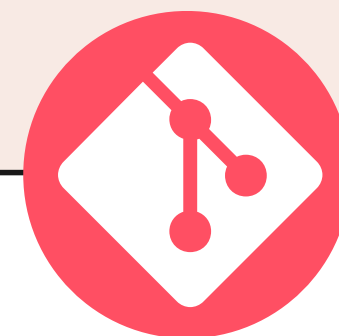
تهانينا

لقد قمت بعمل رائع لكن ينتظرك المزيد...
(فهل أنت مستعد؟)



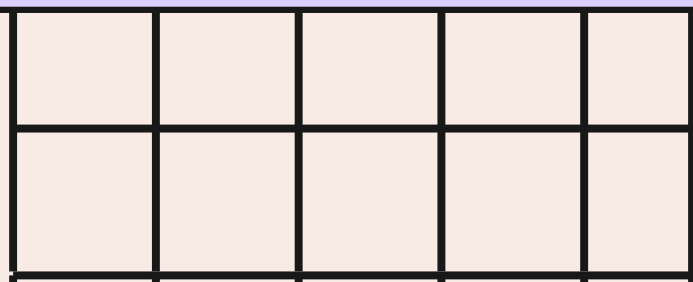


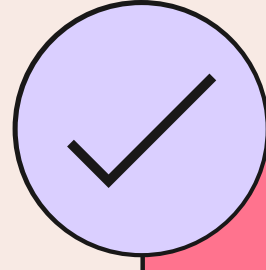
v2.6.0-beta.2 build: release 2.6.0-beta.2
build: build 2.6.0-beta.2
feat: dynamic directive arguments for v-on, v-bind and custom direct
origin/dynamic-directive-arguments feat: dynamic args for cust
perf: improve scoped slots change detection accuracy (#9371)
test: test cases for v-on/v-bind dynamic arguments
refactor: v-bind dynamic arguments use bind helper
test: fix tests, resolve helper conflict
fix: fix middle modifier
feat: handle dynamic argument for v-bind.sync
origin/slot-optimization perf: improve scoped slot
feat: dynamic directive arguments for v-bind and v-
refactor: extend dom-props update skip to more a
fix: fix checkbox event edge case in Firefox
test: fix tests in IE/Edge



العودة بالزمن

كيف تعود بالزمن إلى التعديلات السابقة؟





ماذا يعين العودة بالزمن؟

- خلال الفترة السابقة قمت بالتعديل على الملفات واضفت فيها الكثير من المعلومات وغيرها وقمت بأكثر من commit لكنك لأي سبب تود إستعادته أحد النسخ التي قمت بعمل commit لها خلال الفترة السابقة هذا ما يعنية العودة بالزمن ويمكن العودة بالزمن على عدة ملفات او ملف محدد او كل الملفات.

كيف تقوم بذلك؟

في البداية قم بعمل

```
git log
```

وحدد ما هي النسخة التي تود العودة لها في حالتها حد النسخة التي في وقت 23:23:32 ونسخت الهاش بجانب commit ثم كتبت الامر التالي:

```
git checkout <الهاش>
```

ما سيحدث او ان الكود البرمجي عاد إلى حالته السابقة والتي تم حفظها في الوقت المذكور لعكس العملية أكتب

```
git checkout master
```

```
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$ git log
commit d8a3daa6133a1025e08fb6b58a341282dfa73671 (HEAD -> master)
Author: [REDACTED]
Date: Thu Feb 17 23:24:32 2022 +0300

    add more style to css

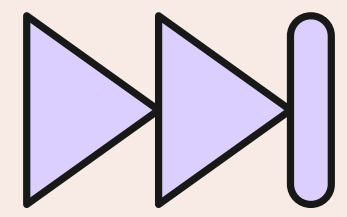
commit 69cbb9d35c6b86cd5ae9097bb364b8a4aba92c1c
Author: [REDACTED]
Date: Thu Feb 17 23:23:32 2022 +0300

    add text to the html page

commit 45aa706710e099fd545a8a391f84dbf8e100cbcd
Author: [REDACTED]
Date: Thu Feb 17 23:19:29 2022 +0300

    add html code and style using css
```

```
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$ git checkout 69cbb9d35c6b86cd5ae9097bb364b8a4aba92c1c
```

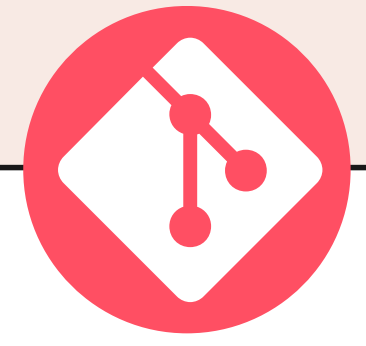


— X

v2.6.0-beta.2 build: release 2.6.0-beta.2
build: build 2.6.0-beta.2
feat: dynamic directive arguments for v-on, v-bind and custom direct
origin/dynamic-directive-arguments feat: dynamic args for cust
perf: improve scoped slots change detection accuracy (#9371)
test: test cases for v-on/v-bind dynamic arguments
refactor: v-bind dynamic arguments use bind helper
test: fix tests, resolve helper conflict
fix: fix middle modifier
feat: handle dynamic argument for v-bind.sync
origin/slot-optimization perf: improve scoped slot
feat: dynamic directive arguments for v-bind and v-
refactor: extend dom-props update skip to more a
fix: fix checkbox event edge case in Firefox
test: fix tests in IE/Edge

g
S.md

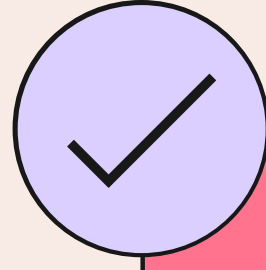
⏪ ⏩



الفروع او branches

ما هي؟ وما فائدتها؟



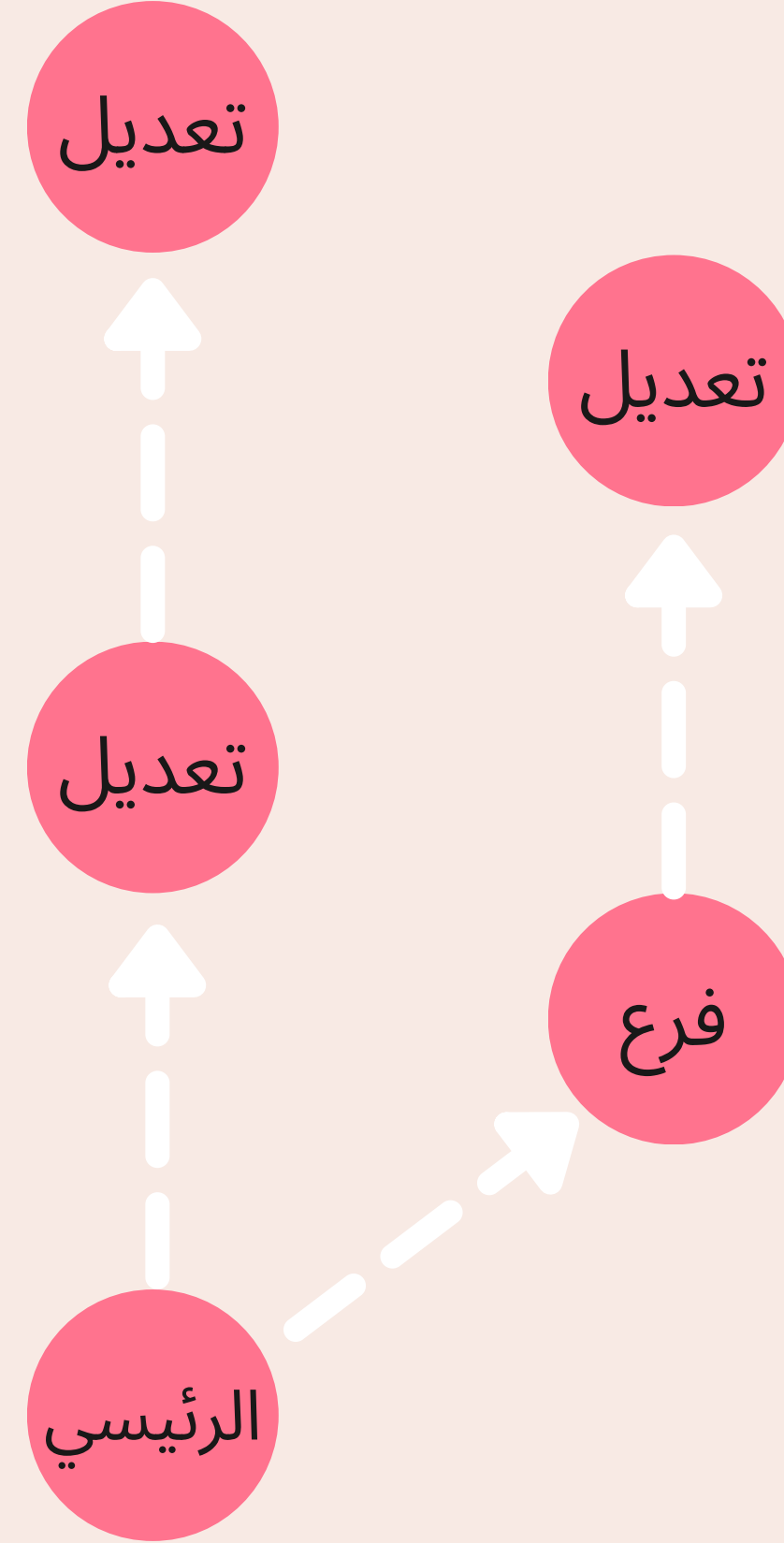


الفروع او branches وماهيتها؟

- الفروع هي أحد مميزات git ومتأكد أنك كنت ترى صورها في هذا الكتيب فكرتها بكل بساطة بأنه يمكن لك تطوير خاصية ولكن لا تدمجها مع كودك الرئيسي وبعد أن تتأكد منها يمكنك دمجها في اي وقت.
- تسمح الفروع بإنشاء نسخه كامله من المشروع الرئيسي، خلال فترة عملنا قمنا بإنشاء ملف رئيسي او فرع رئيسي فقط حيث أن الكود الرئيسي تتفرع منه الفروع ولكل فرع سيكون في البداية نسخه كاملة من المشروع ومع الوقت يمكن تطوير الرئيسي وفي نفس الوقت تطوير الفرعي وبدون اي تعديل على الرئيسي كما لو انه كود مختلف تماماً

مثال

في المثال التالي لاحظ
الرئيسي وهو الكود الرئيسي
الذي انشأناه حيث قمنا بعمل
نسخه منه وهنا ظهر فرع
نسخه طبق الأصل منفصلة عن
الرئيسي وقمنا بالتعديل عليها
كما وفي نفس الوقت قمنا
بعمل تعديلين على الرئيسي
هذا الامر جعل عملية التطوير
اكثر تقدماً وتعقيداً حيث
يمكنك الآن العمل على خاصية
في الفرع دون التعديل على
الرئيسي



كيف نقوم بذلك

خلال عملنا من الممكن أنك لاحظت كلمة master بجانب إسم المجلد هذا يعني أننا على الرئيسي لكي نتأكد من ذلك نقوم بكتابة الامر التالي:

git branch

وهذا الامر يسمح لنا بعرض كافة الفروع ومن ما يظهر لدينا فرع واحد فقط وإسمه master وهو الفرع الرئيسي

```
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$ git branch
* master

ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$
```

كيف نقوم بذلك

حسناً كيف نقوم بإنشاء فرع جديد بكل بساطة سنقوم بكتابة الامر التالي:

```
git branch <إسم الفرع>
```

وبعدها نكتب الامر التالي:

```
git checkout <إسم الفرع>  
>الجديد
```

لكي نتقل للتطوير عليه وتلاحظ بأنه قد تغير الإسم من master إلى develop وهو إسم الفرع الذي قمت بإنشائه الآن اي تطوير يتم على الفرع ولا يتم على الرئيسي او master ويمكن عمل commit كما شرحنا سابقاً

```
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
```

```
$ git branch develop
```

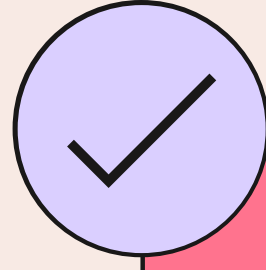
```
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
```

```
$ git checkout develop
```

```
Switched to branch 'develop'
```

```
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (develop)
```

```
$
```



ماذا يحدث عندما نعدل على الفرع؟

- بعد قيامك بالتحويل إلى الفرع والتعديل عليه وعمل commit سيذهب التعديل إلى الفرع وفي حالتي هو develop ويمكن التطوير إلى ما لا نهاية ويمكن العودة للفرع الرئيسي عبر كتابة الامر

```
git checkout master
```

ولإستعراض الفروع الأخرى أكتب

```
git checkout
```

ولتحويل إلى فرع والدخول عليه

```
git checkout <bransh-name>
```

مراجعة لأوامر

هذه القائمة على اليمين هي الأوامر التي تم شرحها ولكن لست جميعها ويمكن الرجوع إلى مرجع Git للمزيد من التفاصيل

init

checkout

branch

log

help

commit

add

status

--version

config

git

كيف ندمج الفروع

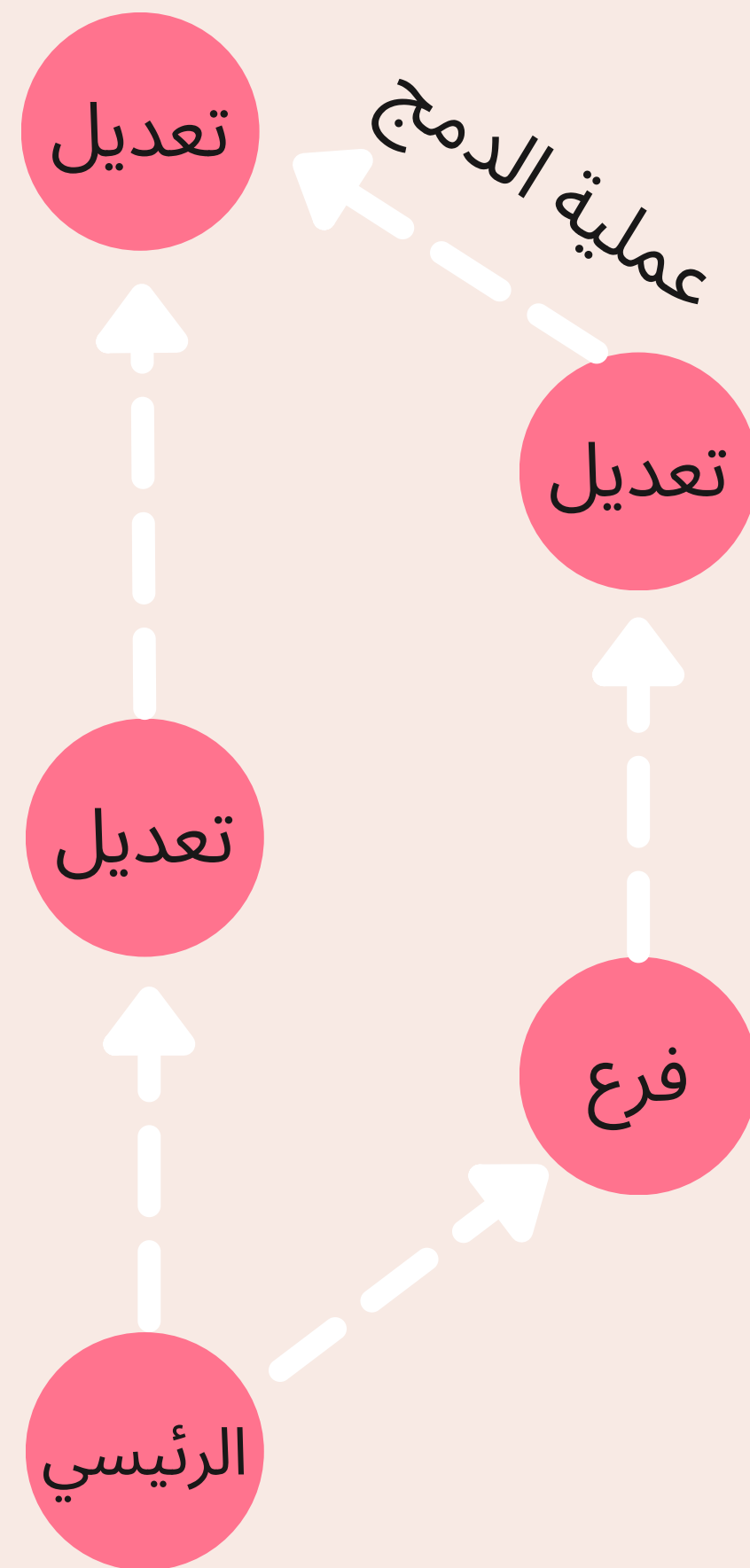
لكي نقوم بدمج تعدينا على الفرع الرئيسي نقوم أولاً بالتحويل إلى الفرع الرئيسي

```
git checkout master
```

ثم نستخدم الامر

```
git merge <branch-name>
```

ومتبوع بإسم الفرع وسيتم دمجها على الرئيسي مباشرة وعند مراجعتك للكود ستجد أنه تم دمج كل شيء



كيف ندمج الفروع

لاحظ كيف ظهر شريط يخبرك بأنه يوجد عدد من التعديلات حيث تم إضافة سطور وإزالة سطور برمجية آخر وتم الدمج وعند كتابة أمر

```
git branch
```

لازال فرع develop موجود وهذا لا يعني بأن الفرع يتوجب حذفه لكن توقف التطوير عليه ويمكن دائماً الرجوع إلى develop وإكمال التطوير ولحذف فرع نكتب

```
git branch -d
```

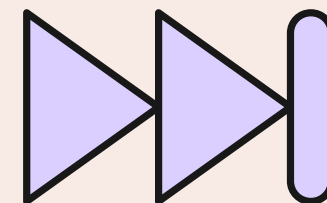
```
<branch-name>
```

وهذا سيحذف الفرع مباشرة

```
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$ git merge develop
Updating d8a3daa..603c0b6
Fast-forward
 index.html | 23 ++++++-----
 style.css  |  5 +++++
2 files changed, 19 insertions(+), 9 deletions(-)
```

```
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$ git branch
develop
* master
```

```
ALWALEED@MSI MINGW64 ~/Desktop/MyAppWithGit (master)
$
```

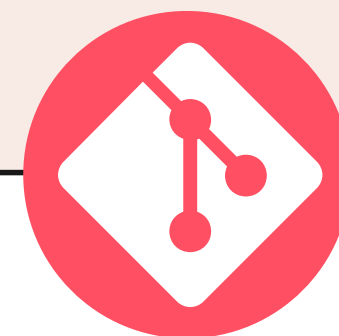


— X

v2.6.0-beta.2 build: release 2.6.0-beta.2
build: build 2.6.0-beta.2
feat: dynamic directive arguments for v-on, v-bind and custom direct
origin/dynamic-directive-arguments feat: dynamic args for cust
perf: improve scoped slots change detection accuracy (#9371)
test: test cases for v-on/v-bind dynamic arguments
refactor: v-bind dynamic arguments use bind helper
test: fix tests, resolve helper conflict
fix: fix middle modifier
feat: handle dynamic argument for v-bind.sync
origin/slot-optimization perf: improve scoped slot
feat: dynamic directive arguments for v-bind and v-
refactor: extend dom-props update skip to more a
fix: fix checkbox event edge case in Firefox
test: fix tests in IE/Edge

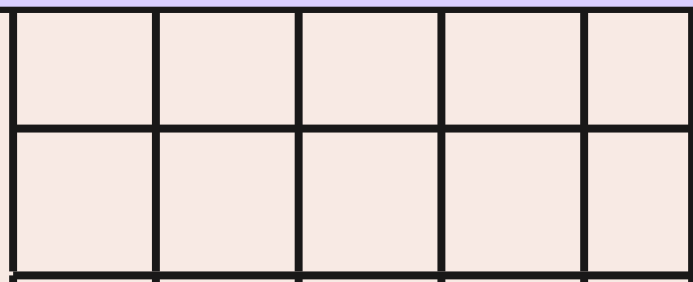
g
S.md

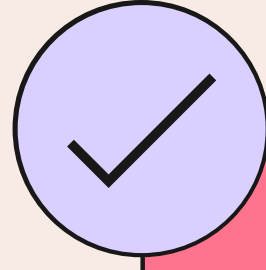
⏪ ⏩



ما هي تقنية GitHub؟

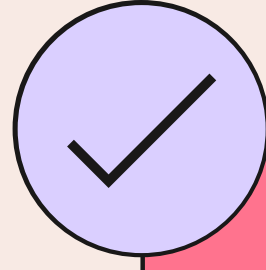
وما فائدتها؟





ما هي تقنية GitHub؟

- خلال الفصول السابقة كنا نقوم بعمل كل الفروع والـ commit محلياً فقط ولا نقوم بأي commit تسمح للأشخاص الآخرين بالعمل معنا في نفس المشروع هذا لا يعني بأن git لوحدها لا يمكن لها القيام بذلك بل على العكس يمكن لهذه التقنية القيام بذلك ولكن هذا الامر يتطلب مجهود اكبر بكثير وهنا تنجح تقنية GitHub حيث أنها خدمه سحابية فبدل من قيامك بكل ما قمت به في git محلياً يمكنك القيام بذلك سحابياً نعم سيتمكن الجميع من المشاركة في تطوير الشيفرة المصدرية الخاصة بك بعد موافقتك بالطبع وكل ذلك بإستخدام تقنية GitHub وبشكل مجاني



جد وثن هي؟

ما هي تقنية GitHub؟

- فبعد تعرفنا على تقنية Git ستجد ان مشاريعك التي قمت بمشاركتها باستخدام تقنية Git ستحتاج الى مكان لتبقى فيه , وهنا يأتي دور جيت هاب , GitHub هو موقع لتجميع المشاركات والمشاريع التي يتم عملها باستخدام تقنية Git , وفي واقع الأمر , لا يمكنك رفع مشروع مباشرة الى موقع GitHub بل عليك ان تكون قد قمت بإنشاء المشروع على جهازك الخاص ومن ثم رفعه - عبر سطر الأوامر او برامج مخصصة - الى حسابك في جيت هاب . ولنتقل الآن الى الخطوة الأولى والأهم , انشاء حساب على موقع GitHub

انشاء حساب على موقع جيتهاب Github

عند دخولك الى الصفحة
الرئيسية في موقع جيتهاب ,
سيظهر لك نموذج Form
لتسجيل الدخول او لتسجيل
حساب جديد تماما كأي موقع
آخر

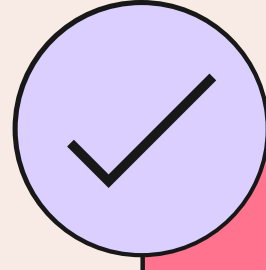
Welcome to GitHub!
Let's begin the adventure

Enter your email
✓ example@example.com

Create a password
→ ●●●●●●

Continue

Password is strong
Make sure it's at least 15 characters OR at least 8 characters including a



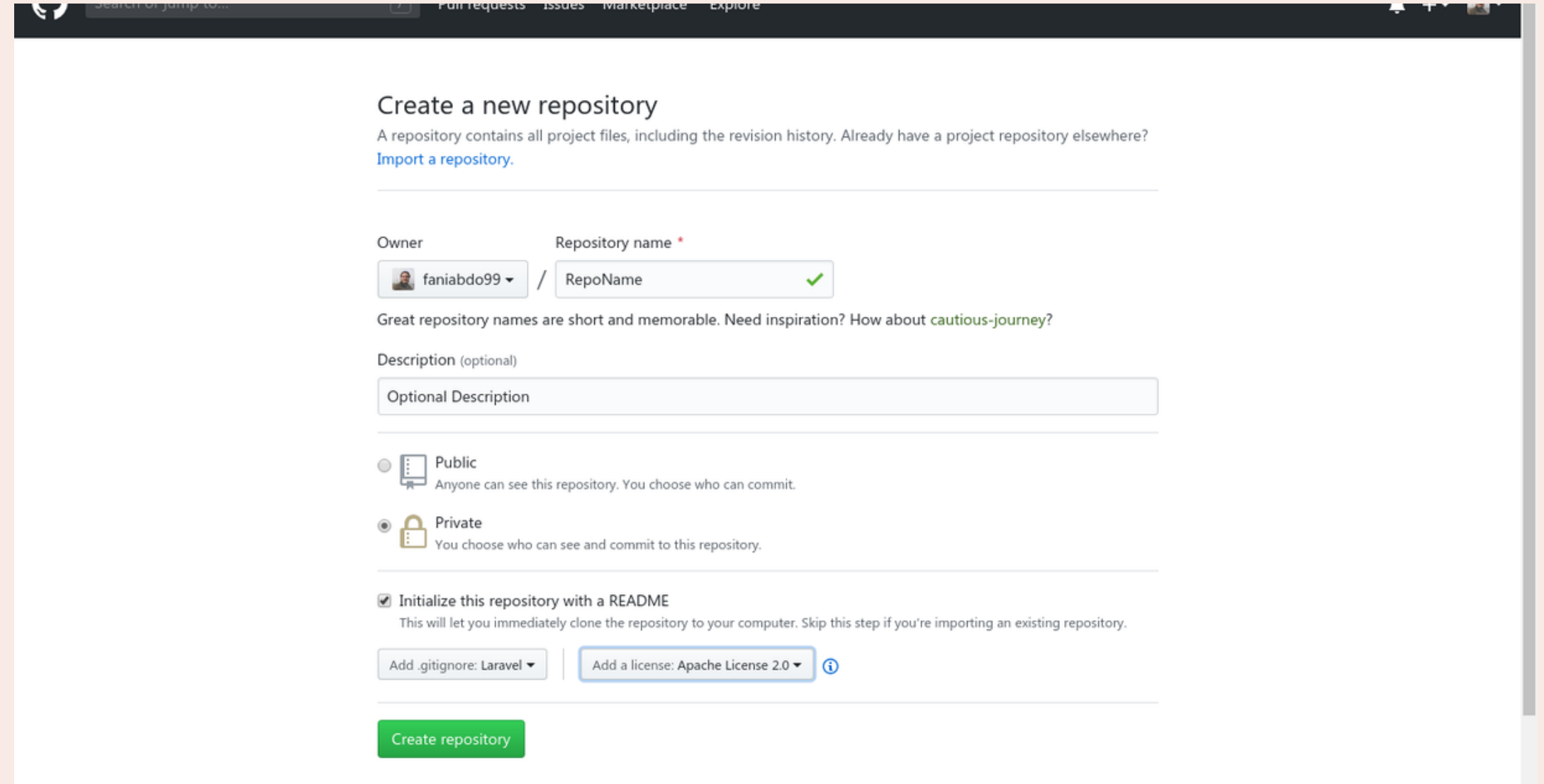
انشاء مشروعك الأول على Github

- في GitHub ، المشاريع تسمى Repository او بالعربية مستودع ، وتختصر عادة بكلمة Repo فعند سماعك لهذه الكلمة اعلم تماماً ان المتحدث يقصد بها مستودع على GitHub ، ويمكنك انشاء مستودع جديد على GitHub ببساطة عبر زر انشاء مستودع الموجود في موقع GitHub.

The screenshot shows the GitHub homepage. At the top, there is a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. On the left, there is a 'Repositories' section with a search bar and a list of repositories including 'faniabdo99/naqrah', 'faniabdo99/takaful', 'faniabdo99/modern-login-page', 'spatie/laravel-analytics', and 'MostafaRabia/naqrah'. In the center, there is a 'Recent activity' section showing a recent issue opened by 'spatie/laravel-analytics' and 'All activity' showing recent pushes to the 'faniabdo99/naqrah' repository. On the right, there is a 'Discover repositories' section with recommendations for 'PHPJunior/nova-logs' and 'Cacti/cacti'. A dropdown menu is open in the top right corner, showing options: 'New repository', 'Import repository', 'New gist', 'New organization', and 'New project'. A notification banner is also visible in the top right corner.

انشاء مشروعك الأول على Github

وبعد الضغط على زر انشاء
مستودع سيتم ارسالك الى
الصفحة التالية وفي هذه
الصفحة يمكنك وضع
المعلومات الأساسية لمشروعك
او بالأصح "مستودعك" الجديد
على Github, ولنقلها
بشكل أكبر



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner: faniabdo99 / Repository name: RepoName ✓

Great repository names are short and memorable. Need inspiration? How about cautious-journey?

Description (optional): Optional Description

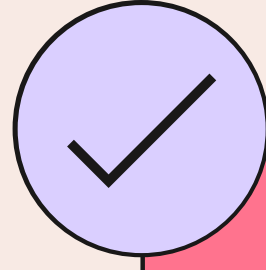
Public: Anyone can see this repository. You choose who can commit.

Private: You choose who can see and commit to this repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

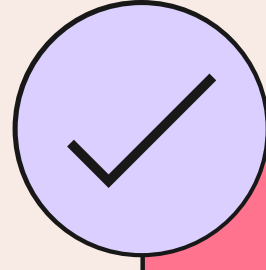
Add .gitignore: Laravel | Add a license: Apache License 2.0 ⓘ

Create repository



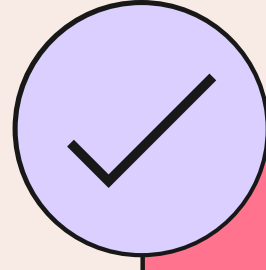
انشاء مشروعك الأول على Github

- **Repository name اسم المستودع** يمكنك هنا وضع اسم المستودع , غالباً ما يكون اسم المشروع او الحزمة التي تعمل عليها , لا يوجد قواعد هنا , فقط تأكد ان اسم المستودع ليس مأخوذ سابقاً .
- **Description الوصف** وصف المستودع أو المشروع , هذا الحقل اختياري يمكنك تركه فارغاً , لكن في حال كنت تعمل على مشروع مفتوح المصدر فينصح باضافة شرح مبسط عن المشروع ولو بكلمات بسيطة
- **Public / Private نوع المستودع** , **خاص او عام** , المستودعات العامة تكون مطروحة للجميع أمام مجتمع مستخدمي GitHub أما المستودع الخاص فهو لك ان فقط وعدة اشخاص يمكنك اضافتهم للعمل معك على المشروع والمشروع الخاص انسب للفرق التي تستخدم GitHub للعمل المشترك.



انشاء مشروعك الأول على Github

- **ملف README** ملف مشهور نوعاً ما , وهو ملف "اقرأني" يمكن هنا وضع معلومات عامة عن المشروع وكيف يعمل وما الفائدة منه , الأمر متعلق بك تماماً هنا ومترك لإبداعك.
- **Add Licenses , gitignore , Add** كما نعلم جميعاً ملف `gitignore` هو ملف يساعد تقنية جيت على اهمال عدة ملفات وعدم رفعهم الى المشروع , كالملفات الاساسية وملفات الخاصة بأطر العمل اما اضافة `Licenses` فهي الشهادة التي يعمل بها المشروع , نوعاً ما عن احقية المستخدمين استخدام جزء او كل مشروعك من الناحية القانونية.



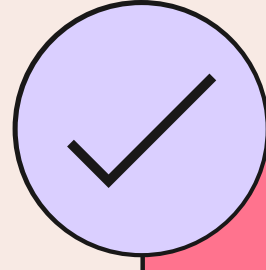
انشاء مشروعك الأول على Github

- وهكذا , بعد أن قمت بتحديد معلومات مستودعك , قم بالضغط على زر Create Repository وسيتم انشاء المشروع وسيتم اعادة توجيهك الى الصفحة الرئيسية للمشروع , وفي بعض الحالات قد يتم توجيهك الى صفحة إعدادات المشروع , لا يمكننا تأكيد المعلومة هنا.

كيف أرفع ملفات المشروع الى مستودع Github؟

وهنا المرحلة الحاسمة , الآن
وبعد أن قمنا بانشاء مستودع
جديد ستجد ان المستودع خال
, ربما ملف README ان كنت
قد اخترت انشائه وملفات ال
LICENSE و .gitignore ,
الخطوة التالية هي رفع ملفات
المشروع الفعلية

لنبدأ العمل...



كيف أرفع ملفات المشروع الى مستودع Github؟

- سنستخدم الطريقة الأفضل وهي سطر الأوامر , سنتجه لسطر الأوامر وننتقل للمشروع المناسب عبر أمر `cd` , ونقوم بتشغيل تقنية `git` في المشروع
 - `cd path/to-project` للوصول إلى الملف
 - `git init` لتفعيل تقنية Git
- وبعد ان قمنا بتفعيل تقنية `git` سنبدأ باضافة ملفات المشروع الى المرحلة الأولى :

لتحديد كافة الملفات . `git add`

بعد ذلك , عليك عمل `Commit` للتغييرات لحفظها مع رسالة معينة , تحتاج

عمل ذلك في حال احتجت العودة الى مشروعك في نقطة معينة مثلاً

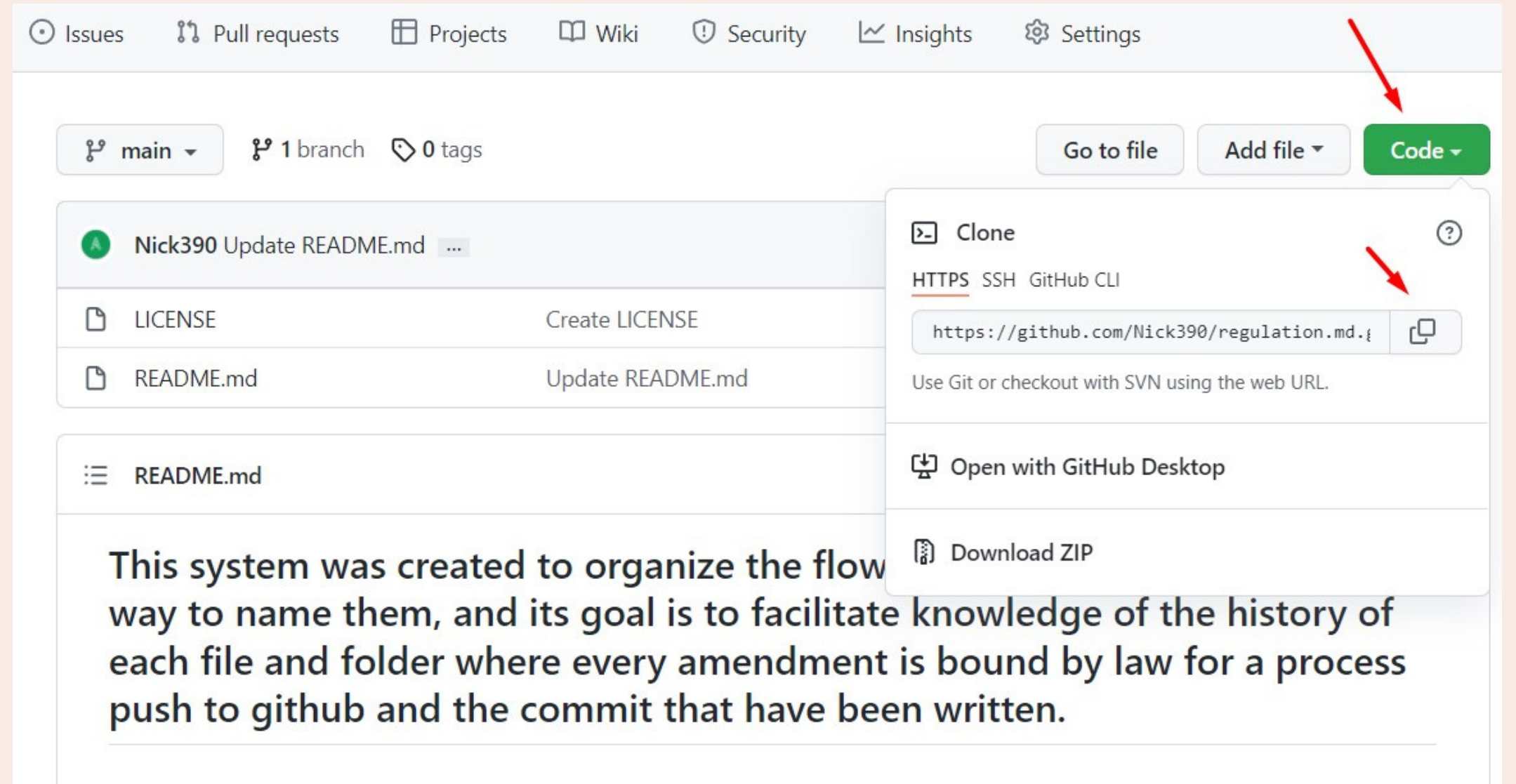
```
git commit -m "First Commit to Github Repository"
```

طريقة حفظ رابط المشروع على موقع GitHub

ندخل على مشروعنا على
GitHub وننقر على زر Code
ومن ثم HTTPS وننسخه ثم
نكتب الأمر التالي:

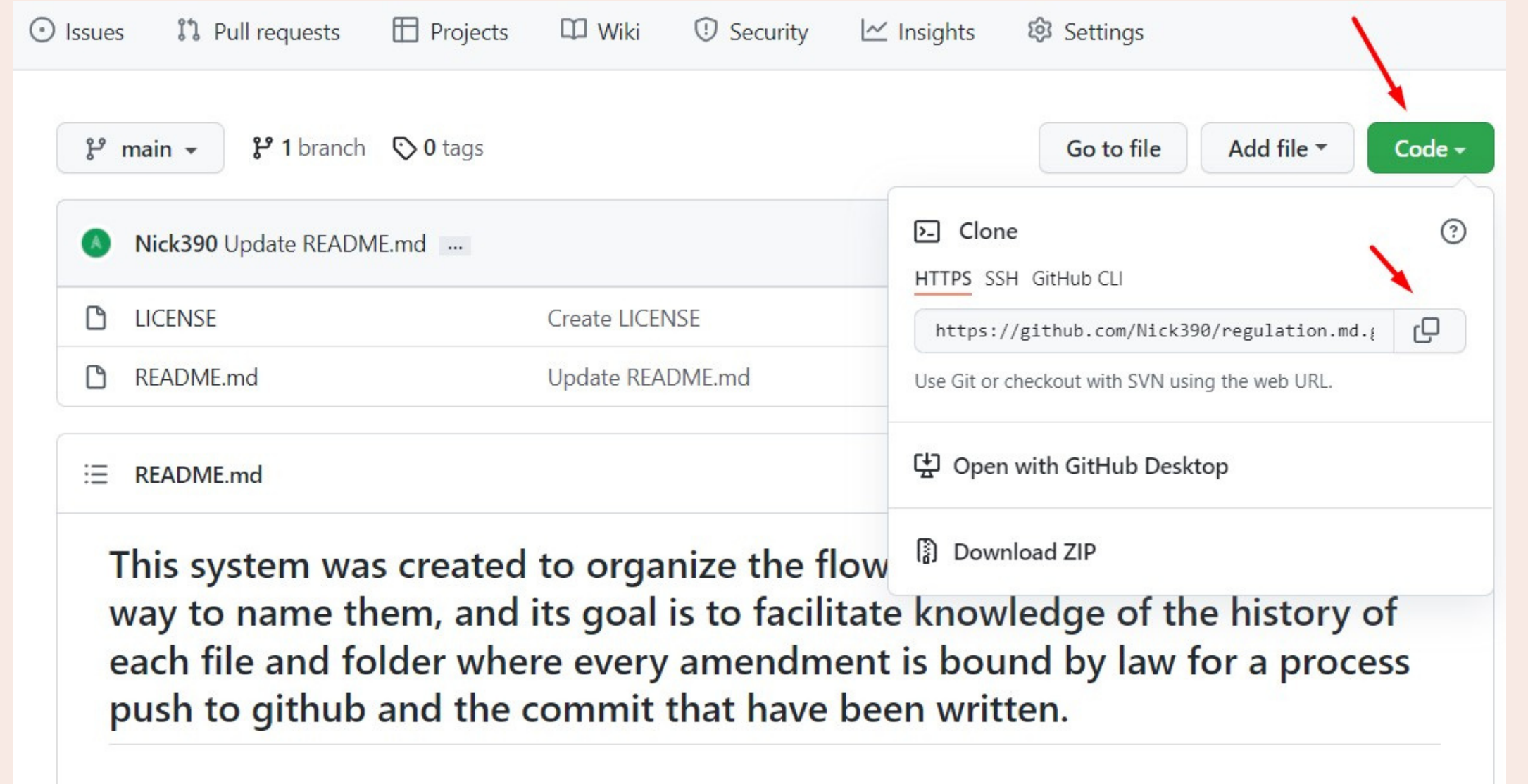
```
git remote add origin  
remote repository URL  
→https://github.com/  
الرابط الذي نسخته
```

ثم إنتر



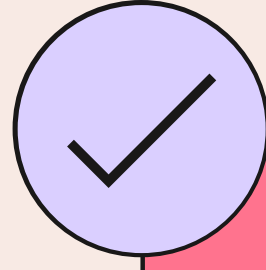
كيف نرفع اول ملف او عدة ملفات

الآن سنقوم برفع الملفات الى خادم Github لتصبح متوفرة على الموقع , وعليك التنبه هنا من نقطة مهمة , الأمر الذي سنكتبه تالياً سيقوم بنشر الملفات الى العامة في حال كان المستودع عام Public وسيستطيع اي شخص الوصول الى الملفات , اما ان كان المستودع خاص Private فلا داع للقلق , لن يستطيع احد الوصول الى الملفات الا بعد ان تسمح انت له بذلك.



The screenshot shows a GitHub repository page for a user named Nick390. The repository is named 'regulation.md'. The page displays the 'main' branch with 1 branch and 0 tags. There are two files listed: 'LICENSE' and 'README.md'. The 'README.md' file is selected, and its content is visible: 'This system was created to organize the flow way to name them, and its goal is to facilitate knowledge of the history of each file and folder where every amendment is bound by law for a process push to github and the commit that have been written.'

The 'Code' button is highlighted with a red arrow, and the 'Clone' dropdown menu is open, showing options for cloning the repository using HTTPS, SSH, or GitHub CLI. The HTTPS URL is displayed as `https://github.com/Nick390/regulation.md.{'`. A red arrow points to the 'Clone' button in the dropdown menu.



أمرين جديده الأول إسمه push والثاني pull

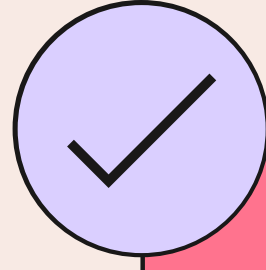
- الامر التالي ستستخدمه كثيراً وهو بكل بساطة سيقوم برفع الملفات إلى النظام

```
git push origin master
```

بعد تنفيذ هذا الأمر سيتم رفع ملفات مشروعك الى المستودع ، و تهانينا تمت العملية بنجاح!

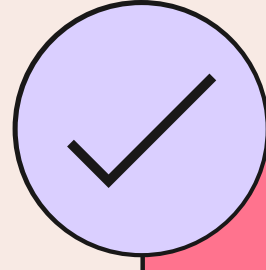
الآن سنضع فرضية بسيطة ان أحد زملائك قام بعمل تعديل ورفعته الى جيتهاب ، فان تم هذا الامر سيمنعك موقع جيتهاب من رفع تعديلاتك قبل تنزيل التعديلات السابقة ، لذا قبل ان تقوم بعمل تعديلاتك تأكد دائماً من تنفيذ الأمر التالي :

```
git pull origin master
```



أمرين جديده الأول إسمه push والثاني pull

- حسناً لنشرح المزيد push يعني إدفع إلى المستودع الذي قمت بعملة على موقع GitHub و pull إسحب ملفات البرمجة التي قام زميلي بتعديلها بتنفيذ هذا الأمر , ستتأكد انك تمتلك أحدث نسخة من ملفات المشروع , ومن ثم يمكنك القيام بتعديلاتك ورفعها وتبليغ اعضاء الفريق ليقوموا بعمل git pull ايضاً.



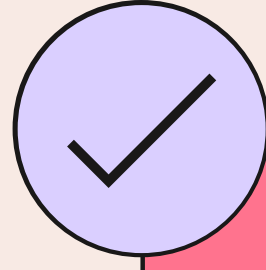
ما المميز حول Github ؟

- على الرغم من وجود منافسين كثر GitHub الا انه الأفضل , لماذا ؟
اليك بعض الأرقام . . .
- عدد المشاريع (المستودعات) الموجودة على GitHub تجاوز عام 2019 ال 57 مليون مشروع , منها 28 مليون مشروع مفتوح المصدر يمكنك الاستفادة منها !
- عدد المستخدمين في موقع GitHub على 37 مليون مستخدم
- يحتوي GitHub على مشاريع ضخمة ومهمة بشكل مفتوح المصدر منها بوتستراب , جيكويري , Node.js , نظام تشغيل جوجل فوشيا , FontAwesome وغيرها الكثير
- من المشاريع المميزة على GitHub مشروع **نواة لينكس** بواسطة لينوس تروفالديس نفسه !

في النهاية

موقع GitHub من المواقع المهمة لكل مبرمج في مسيرته المهنية ، في الواقع بعض الشركات تقيس خبرة المبرمج بنسبة مشاركته في المشاريع المفتوحة المصدر على GitHub ، لذا لا تتردد في البدء في اكتشاف هذه المكتبة الهائلة من الأكواد ، ابدأ الآن !

ستجدني على منصات التواصل
الإجتماعي @AlwaleeAlwabe1



المراجع

- ماهو Git ؟ وماهو github ؟
- جت (برنامج).
- نظام النسخ المتلاقية
- Concurrent Versions System
- أباتشي سيفيرجين
- إدارة المراجعة الموزعة
- ما هو جيتهاب Github ؟ دليل المبتدئين لاستخدام موقع غيتهاب
- Distributed version control
- --everything-is-local
- Introduction to Git
- Learn Git In 15 Minutes
- Introduction to GitHub
- Learn Github in 20 Minutes