

ترجمة م. بكري علوش



www.bakrialloush.com

github.com/bakrialloush

linkedin.com/in/bakrialloush

الفهرس

1.....	الفهرس
2.....	مقدمة المترجم
3.....	مقدمة الكتاب
4.....	الفصل الأول البدء باستخدام GITHUB
14.....	الفصل الثاني نسخ GITHUB احتياطياً
15.....	الفصل الثالث نسخ مشروع من GITHUB
17.....	الفصل الرابع عرض الخط الزمني وآخر أحداث GITHUB في موقعك الإلكتروني
20.....	الفصل الخامس تحميل ملف واحد من مشروع في GITHUB
21.....	الفصل السادس برنامج GITHUB DESKTOP
24.....	الفصل السابع صفحات GITHUB
26.....	الفصل الثامن كيف تنشئ إشارة LABEL وتخصصها؟
27.....	الفصل التاسع الملاحظات ISSUES
28.....	الفصل العاشر طلبات السحب PULL REQUEST
29.....	الفصل الحادي عشر إزالة البيانات الحساسة والملفات الضخمة
32.....	الفصل الثاني عشر تحديث مشروع منسوخ FORKED REPOSITORY
33.....	الفصل الثالث عشر استخدام GIST
35.....	الفصل الرابع عشر استخدام أزرار GITHUB
37.....	الفصل الخامس عشر العمل بـ GITFLOW

بسم الله الرحمن الرحيم

مقدمة المترجم

الحمد لله رب العالمين، والصلاة والسلام على نبينا محمد وعلى آله وصحبه أجمعين، في ظل التطور التكنولوجي المتسارع، أصبحت أدوات إدارة المشاريع البرمجية جزءاً لا يتجزأ من حياة المبرمجين، ومن أبرز هذه الأدوات GitHub، فهي تمثل بيئة متكاملة تسهل التعاون والعمل الجماعي، وتمكن من إدارة الأكواد البرمجية بكفاءة عالية.

وبما أن GitHub أداة Version Control فهي ليست مجرد منصة لتخزين الأكواد، بل أداة قوية تتيح للمطورين تتبع التعديلات على المشاريع، ومشاركة أعمالهم مع الآخرين، مما يساهم في تعزيز جودة العمل وتجنب الأخطاء، الفوائد التي يقدمها GitHub تتجاوز بكثير مجرد حفظ الأكواد، فهو يوفر إمكانيات واسعة لإدارة الإصدارات، مما يتيح العودة إلى النسخ السابقة من الكود البرمجي عند الحاجة بسهولة تامة. في المقابل، فإن عدم استخدام GitHub يعني فقدان هذه الميزات الجوهرية، مما قد يؤدي إلى تضارب في العمل الجماعي وصعوبة في إدارة المشاريع البرمجية المعقدة، كما أن غياب أدوات مثل GitHub قد يتسبب في إهدار الوقت والجهد عند محاولة استرجاع أو تعديل الأكواد بعد تغييرات متعددة.

هذه الميزات دعنتني إلى ترجمة كتيب صغير يعرّف بهذه الأداة السحرية ويشرح ميزاتها ببسر وسهولة، وقد اخترت كتاباً صغيراً لكي يكون سهل القراءة. قد تجد اختلافات في بعض الصور إذا تم تحديث أو تعديل واجهات GitHub، ولكن الأساس غالباً سيبقى واحداً. الميزات كثيرة والرحلة مشوقة، استمتع...

م. بكري علوش 2023

bakrialloush.com

مقدمة الكتاب

بإمكانك مشاركة هذا الكتاب مع كل من تظن أنه سينتفع به، وبإمكانك تحميل تحديثات النسخة

الأصلية باللغة الانكليزية عبر الرابط [GitHub](#).

إنها نسخة مجانية غير رسمية صممت لأهداف تعليمية، تم استخراج كل محتواها من [Stack Overflow](#)

[Documentation](#)، والذي ألفه العديد من صناع المحتوى بشكل فردي في موقع Stack Overflow، لذلك

فهي لا تتبع لموقع Stack Overflow ولا لموقع GitHub الرسمي.

تم نشر هذا المحتوى تحت رخصة المشاع الإبداعي BY-SA، وسيتم تزويدك بقائمة المساهمين بمحتوى كل

فصل في جزء المؤلفين في نهاية الكتاب، قد تكون الصور محمية بحقوق النشر ما لم يتم التصريح بخلاف

ذلك، وإن جميع العلامات التجارية والعلامات التجارية المسجلة المذكورة ملك لأصحابها.

إن استخدامك للمعلومات الواردة في هذا الكتاب على مسؤوليتك الشخصية، فهو غير مضمون الصحة أو

الدقة، أرجو إرسال المقترحات والتصحيحات عبر البريد الإلكتروني

backar-net@hotmail.com

الفصل الأول

البدء باستخدام GitHub

ملاحظات

- ✓ يقدم لك هذا الفصل نظرة عامة عما يعنيه GitHub، ولماذا قد يرغب مطوّر البرمجيات باستخدامه.
- ✓ في الفصل الحالي سيتم ذكر العناوين العريضة والمواضيع الأساسية لـ GitHub.

ما هو GitHub؟

GitHub هو عبارة عن تجميع ضخم من الكثير من المشاريع البرمجية يسمى كل منها Repository.

إنشاء حساب

- ✓ قم بزيارة الصفحة الرئيسية لموقع GitHub عبر الرابط <http://github.com>
- ✓ اختر اسم مستخدم وكلمة مرور وأدخل عنوان البريد الإلكتروني، أنت جاهز الآن للانطلاق!

أدوات مفيدة

في البداية قد يكون فهم طريقة عمل التحكم بالإصدارات Version Control مربكاً قليلاً للمبتدئين، لذلك يوجد أداة بواجهة رسومية (نوافذ وأزرار) تدعى [GitHub Desktop](#) متاحة للمستخدمين للتحميل.

إنشاء مشروعك الأول First Repository

كما أسلفنا في إمكانك اعتبار Repository كمشروع برمجي، وبإمكانك إنشاء المشروع سواء كنت

متصلاً أو غير متصل بالإنترنت، باتباع الخطوات التالية:

متصل بالإنترنت:

1. سجل دخولك وافتح ملفك الشخصي profile.
2. افتح تبويب المشاريع Repositories الموجود أعلى الصفحة.
3. انقر زر جديد New الأخضر، والآن أصبحت جاهزاً.

بدون اتصال:

1. حمل وثبت [git Client](#) (لا تنس اختيار ما يتوافق مع نظام التشغيل لديك).
2. بعد التحميل والتنصيب، بإمكانك استخدام أي من أداة سطر الأوامر أو الواجهة الرسومية.
3. بعد التنصيب، أنشئ حساباً على GitHub.
4. انقر + من أعلى اليمين، واختر إنشاء مشروع جديد أو فتح مشروع موجود.
5. إذا اخترت إنشاء مشروع جديد أدخل اسمه، واختر فيما إذا كنت تريده خاصاً أو متاحاً للعامّة.
6. انقر إنشاء المشروع.

ملف README

إن لم يكن ملف README.md موجوداً ضمن مشروعك، سيحلل GitHub ملف README.rdoc لعرض التفاصيل، وإذا احتوى المشروع الملفين معاً سيتم استخدام README.md بدون أي إشعار بذلك.

محتويات قد تكون في ملف README:

✓ عنوان المشروع:

نبذة مختصرة عن مشروعك، قد يحوي أيضاً روابط كالموقع والبريد الإلكتروني وحسابات مواقع التواصل الاجتماعي.

✓ التحميل:

رابط للملف التنفيذي أو ملف التنصيب، ويمكن أن يحوي روابط للإصدارات السابقة.

✓ التنصيب:

كيف يمكن استخدام الكود، ما هي المتطلبات الأساسية لتشغيله، ما هي المكتبات المطلوبة...

✓ الوصف:

يمكن أن يحوي جزءاً من الكود أو صورة متحركة أو فيديو أو لقطة شاشة...

✓ المؤلفون:

أسماء كاتب الكود وبعض معلومات التواصل...

✓ الشكر:

قائمة بالأشخاص أو المجموعات الذين أعانوا وألهموا المبرمج على مدى فترة المشروع.

✓ المساهمات:

خطوات المساهمة في المشروع (إضافة ميزة، تقرير بطلاً ما، إرسال تصحيح...)، إضافة إلى رابط

التوثيق أو ملف التعليمات.

✓ الرخصة:

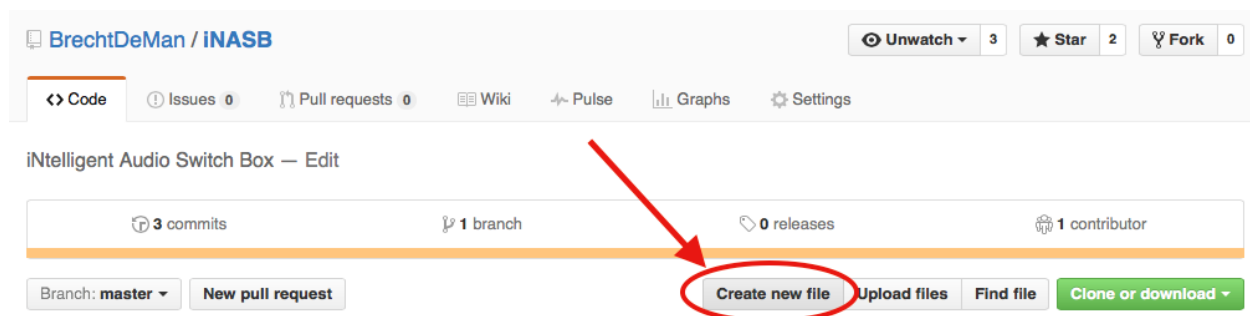
مقدمة بسيطة عن الرخصة المطبقة على المشروع، ويمكن تضمين رابط لموقع الرخصة الرسمي.

ملف الترخيص:

يساعدك GitHub على إضافة الرخصة لمشروعك بأسرع ما يمكن، وكبديل لإضافة ملف الترخيص

الخاص بك يدوياً قم بالتالي:

1. انقر "Create new file" في صفحة مشروعك:



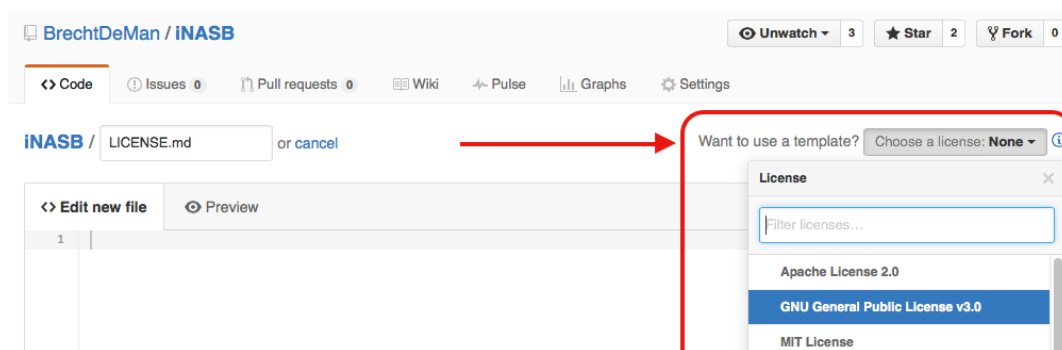
2. في الصفحة التالية:

1. أدخل اسم الملف LICENSE.md أو LICENSE.txt.

2. سيظهر نافذة "Want to use a new template":



3. اختر الرخصة المفضلة:



4. بإمكانك رؤية الرخصة في قسم تفاصيل المشروع:

773 commits	5 branches	94 releases	49 contributors	MIT
-------------	------------	-------------	-----------------	-----

تجويد نصوصك في GitHub:

يزودك GitHub بلغة خاصة [Markdown syntax](#) لإضافة ميزات مفيدة لنصوصك.

الترويسة:

```
# Header1
## Header2
### Header3
#### Header4
##### Header5
##### Header6
H1
===
H2
---
```

Header1

Header2

Header3

Header4

Header5

Header6

Header1

Header2

الأنماط الخاصة:

```
*Italic1* _Italic2_  
**Bold1** __Bold2__  
***Bold_Italic***  
~~Strikethrough~~
```

Italic1 Italic2

Bold1 Bold2

Bold_Italic

~~Strikethrough~~

الخط الأفقي:

```
--  
* * *  
_____
```

unordered list:

```
* item-1
  * sub-item-1
  * sub-item-2
- item-2
  - sub-item-3
  - sub-item-4
+ item-3
  + sub-item-5
  + sub-item-6
```

ordered list:

```
1. item-1
  1. sub-item-1
  2. sub-item-2
2. item-2
  1. sub-item-3
  2. sub-item-4
3. item-3
```

unordered list:

- item-1
 - sub-item-1
 - sub-item-2
- item-2
 - Sub-item-3
 - sub-item-4
- item-3
 - sub-item-5
 - sub-item-6

ordered list:

1. item-1
 1. sub-item-1
 2. sub-item-2
2. item-2

1. sub-item-3
2. sub-item-4
3. item-3

الجدول:

```
Table Header-1 | Table Header-2 | Table Header-3
:--- | :---: | ---:
Table Data-1 | Table Data-2 | Table Data-3
TD-4 | Td-5 | TD-6
Table Data-7 | Table Data-8 | Table Data-9
```

Table Header-1	Table Header-2	Table Header-3
Table Data-1	Table Data-2	Table Data-3
TD-4	TD-5	TD-6
Table Data-7	Table Data-8	Table Data-9

الكود:

```
inline code- `int i = 0`
block code-
``` C
for(int i = 0; i < 10; i++){
printf("Hallow World! \n");
}
```
```

```
inline code- int i = 0
block code-

for(int i = 0; i < 10; i++){
    printf("Hallow World! \n");
}
```

الاقتباسات:

```
> Stay hungry; stay foolish.
>> Quality is better than quantity.
>>> Life is not fair; get used to it.
```

Stay hungry; stay foolish.

Quality is better than quantity.

Life is not fair; get used to it.

الروابط:

```
https://github.com
[GitHub] (https://github.com)
[GitHub] (https://github.com "github website")
[GitHub][1]
[1]: https://github.com
```

<https://github.com>

[GitHub](#)

[GitHub](#)

[GitHub](#)

الصور:

```
![GitHub Logo] ( https://i.stack.imgur.com/rDfU0.png "GitHub")
```



قوائم الاختيار:

- [x] completed item
- [] incomplete item

- ☒ completed item
- ☐ incomplete item

الابتسامات Emoji:

:octocat: :+1: :book: :ghost: :bulb: :imp:



[رابط صفحة اختصارات الابتسامات كاملة](#)

مراجع شيفرات SHA:

e7909ea4fbb162db3f7f543d43c30684a3fb745f

أي مرجع لتشفير مناقلة التثبيت Commit سيتحول تلقائياً لـ رابط لنفس تلك المناقلة.

Write

Preview

e7909ea

طلبات السحب Pull Requests ومراجع الأخطاء Issue References:

أي مرجع لطلبات السحب أو الأخطاء ستتحوّل تلقائياً لروابط لتلك المراجع.

يمكن تنفيذ ذلك بإضافة # أمام رقم ذاك الطلب.

الفصل الثاني

نسخ GitHub احتياطياً

أمثلة:

نسخ كل المشاريع من حساب GitHub:

نفذ أمر لينوكس التالي باستبدال username باسم المستخدم، لنسخ جميع مشاريع GitHub من

حساب المستخدم المحدد بالاسم السابق إلى المجلد الحالي:

```
curl "https://api.github.com/users/username/repos?page=1&per_page=100" | grep  
-e 'get_url*' | cut -d \" -f 4 | xargs -L1 git clone
```

هذا الأمر سينسخ أول 100 مشروع فقط.

الفصل الثالث

نسخ مشروع من GitHub

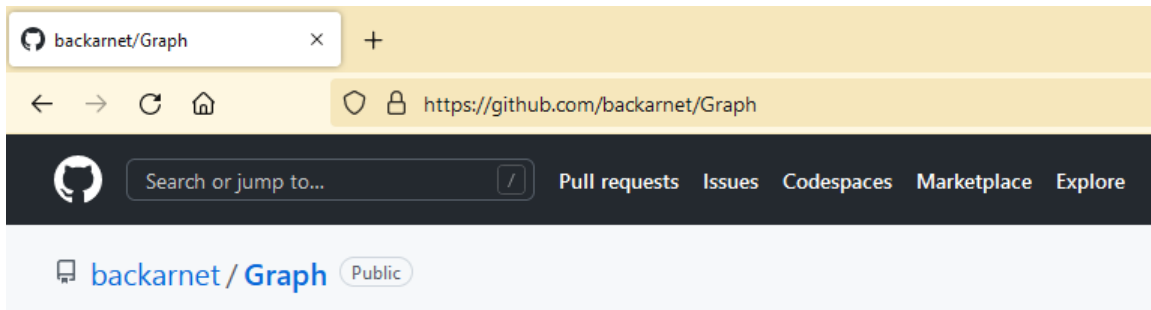
الأمر:

```
git clone github.com/username/repository
```

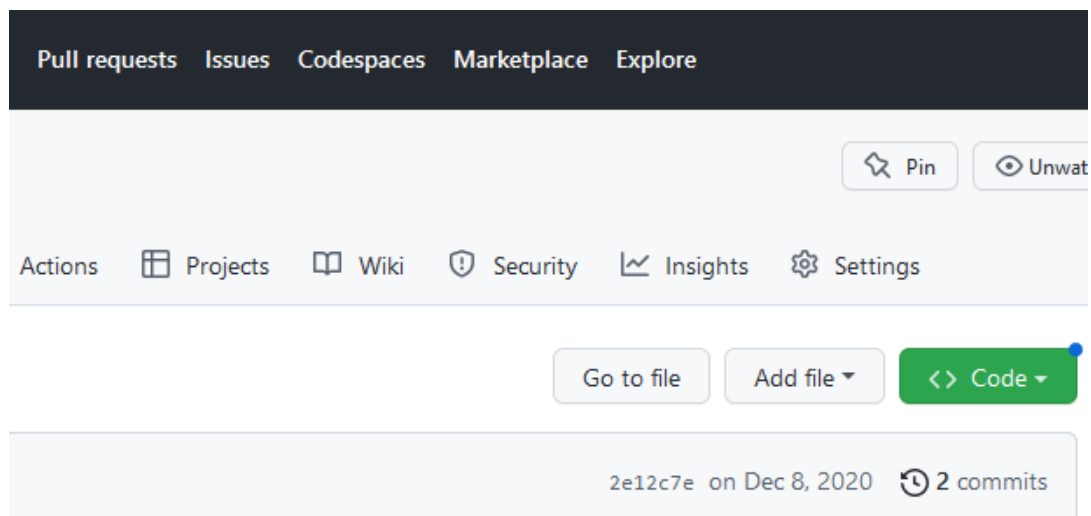
أمثلة:

نسخ مشروع:

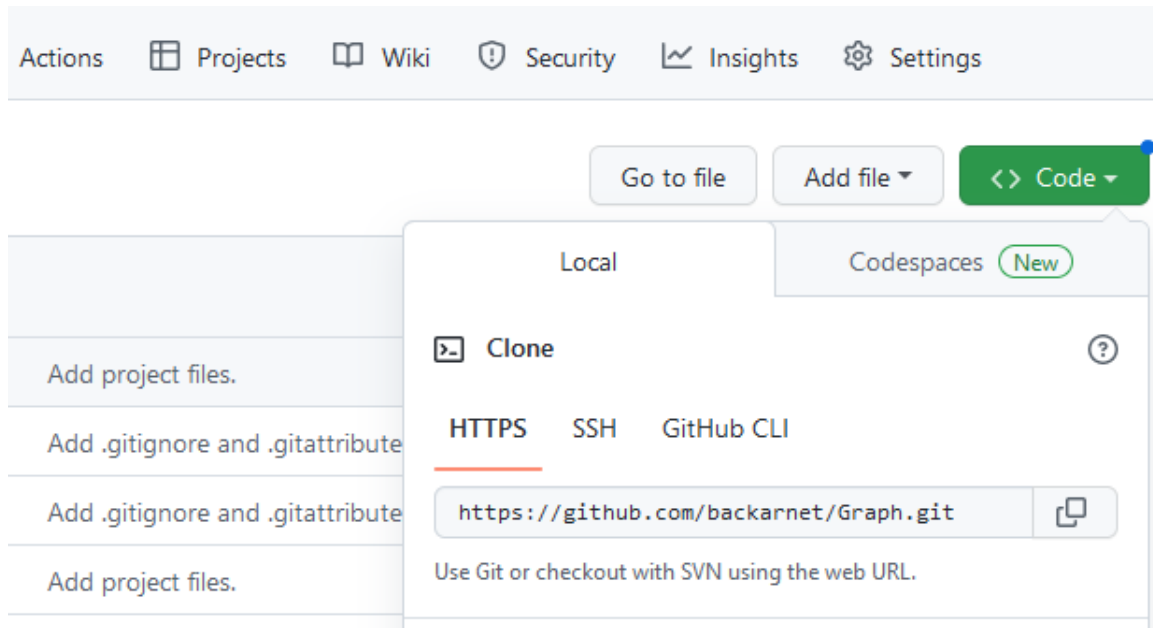
1. انتقل إلى رابط المشروع المطلوب:



2. انقر الزر الأخضر على اليمين:



3. ستظهر نافذة صغيرة، انسخ الرابط المنتهي ب .git.



4. افتح نافذة الأوامر Terminal أو Command prompt على الجهاز الذي تريد نسخ المشروع إليه.

5. انتقل عبر الأوامر إلى المسار الذي تريد نسخ المشروع إليه.

6. أدخل الأمر `git clone <url>` واستبدل url بالرابط المنسوخ من الخطوة 3.

7. انقر Enter.

8. سيظهر الخرج التالي:

```
Cloning into <repo_name>...
remote: Counting objects: 10, done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 10 (delta 1), reused 10 (delta 1)
unpacking objects: 100% (10/10), done.
```

الفصل الرابع

عرض الخط الزمني وآخر أحداث GitHub في موقعك الإلكتروني

أمثلة:

يشرح هذا الفصل كيفية عرض الخط الزمني وآخر الأحداث على موقعك الإلكتروني.

مثال:

في هذا الرابط مثال حي عن هذه الميزة:

<https://newtonjoshua.com>

الخط الزمني في GitHub:

يقدم GitHub خطأ زمنياً عاماً لأي مستخدم بصيغة Atom¹، يمكنك مشاهدة الخط الزمني لحسابك

عبر الرابط:

https://github.com/USER_NAME.atom

المرجع: <https://developer.github.com/v3/activity/feeds>

Feed API الخاصة بـ Google:

عن طريق Feed API تستطيع تحميل أي API أو RSS باستخدام JavaScript فقط، وبإمكانك دمج آخر

الأخبار feeds مع المحتوى الخاص بك مع مخرجات API أخرى ببضعة أسطر من JavaScript، ما يجعل

تكامل موقعك مع محتويات آخر الأخبار سهلاً وسريعاً.

¹ Atom: صيغة بيانات حاسوبية مبنية على XML وتتمثل إحدى تطبيقاتها في نشر آخر الأخبار والتحديثات Feeds. يفيد هذا في إعلام من يتابعونها بتحديث المواقع التي توفر هذه الميزة، مما يسمح بمتابعة عدد ضخم من المواقع (الإخبارية مثلاً) والمدونات دون الحاجة لزيارة المواقع كلها لمعرفة الجديد فيها. (المصدر ويكيبيديا).

المرجع: <https://developers.google.com/feed/v1/devguide>

تحميل JavaScript API:

لتبدأ باستخدام Feed API ضمن الكود التالي في ترويسة صفحة موقعك:

```
<script type="text/javascript" src="https://www.google.com/jsapi"></script>
```

بعد ذلك قم بتحميل Feed API باستخدام (google.load(module, version, package):

```
<script type="text/javascript">
  google.load("feeds", "1");
</script>
```

تحديد رابط آخر الأخبار:

يمكنك استدعاء google.feeds.Feed() بالطريقة التالية:

```
var feed = new google.feeds.Feed("https://github.com/User_Name.atom");
```

تحميل آخر الأخبار:

باستخدام load(callback). يتم تحميل آخر الأخبار المحددة عند تهيئة خدمات Google، وعند الانتهاء

من التحميل يتم استدعاء callback المعطى:

```
<script type="text/javascript">
function initialize() {
  feed.load(function(result) {
    if (!result.error) {
      var container = document.getElementById("feed");
      result.feed.entries.forEach(function (feed) {
        var feedTitle= feed.title;
        var feedLink = feed.link;
        var feedDate = formatDate(feed.publishedDate);
        var feedContent = formatContent(feed.content);
        // display the feed in your website
      });
    }
  });
}
```

```

    });
  }
});
}
google.setOnLoadCallback(initialize);
</script>

```

استدعاء onLoad:

إن `setOnLoadCallback(callback)` هو تابع `static` يسجل تابعاً محدداً `callback` لاستدعائه عند الانتهاء من تحميل الصفحة التي تستدعيه، وعندما يكون API جاهزاً.

```

<script type="text/javascript">
  google.setOnLoadCallback(initialize);
</script>

```

تحديد العدد الأقصى للأخبار:

إن `setNumEntries(num)` تحدد عدد الأخبار الأقصى بالعدد `num`، وكقيمة افتراضية يتم تحميل 4.

```

var feed = new google.feeds.Feed("https://github.com/User_Name.atom");
feed.setNumEntries(500);

```

وهكذا يمكنك إعداد وعرض آخر تحديثات صفحتك في GitHub على موقعك الإلكتروني الخاص.

الفصل الخامس

تحميل ملف واحد من مشروع في GitHub

أمثلة:

من أين مشروع عام باستخدام سطر الأوامر:

هذا المثال من سطر أوامر لينوكس يسحب ملف Graph.sln من مشروع Graph من حسابي الشخصي backarnet، وينزله إلى المجلد الحالي - الذي تطلب الأمر منه - ويعيد تسميته إلى Graph2.sln

```
$ curl https://github.com/backarnet/Graph/raw/master/Graph.sln -o Graph2.sln
```

نسخ رابط الملف الذي تريد تحميله:

1. انتقل إلى الملف المطلوب ضمن المشروع.
 2. انقر زر "row".
 3. انسخ الرابط من شريط عناوين المتصفح.
- المثال السابق يظهر فيه رابط الملف الجاهز للتحميل، يمكنك بسرعة اكتشاف ما يميز رابط التحميل حيث تلاحظ وجود /raw في المسار بعد اسم المشروع Graph.

الفصل السادس

برنامج GitHub Desktop

مقدمة:

كيف تنصب وتستخدم GitHub Desktop؟

كما يتضح من اسم البرنامج فهو أداة تعمل على أنظمة سطح المكتب تتضمن المهام الأساسية لتقنية

Git كنسخ وسحب ورفع ودمج ومزامنة المشاريع...

يهدف هذا البرنامج بشكل رئيسي إلى تقديم طريقة سهلة لإجراء وظائف Git أو GitHub، حيث يقوم ضمناً بتنفيذ الأوامر التي يدخلها المستخدم عادة في واجهة سطر الأوامر.

أمثلة:

التنصيب والإعداد:

التنصيب سهل للغاية، يوجد ملفات تنصيب خاصة بكل أنظمة التشغيل الشهيرة عبر [الرابط](#)، يتوفر

حالياً إصدارين أحدهما تجريبي beta والآخر مستقر stable.

عند بدء الإعداد يجب عليك تسجيل الدخول لحساب GitHub وهذا كل شيء، حيث يمكنك بعد ذلك إنشاء مشروع جديد أو نسخ clone مشروع موجود مسبقاً في ملفات حسابك إلى الجهاز الذي تعمل منه.

ملاحظة:

عند تنصيب GitHub Desktop يتم تنصيب Git معه، لذلك لا داع لتنصيبه يدوياً

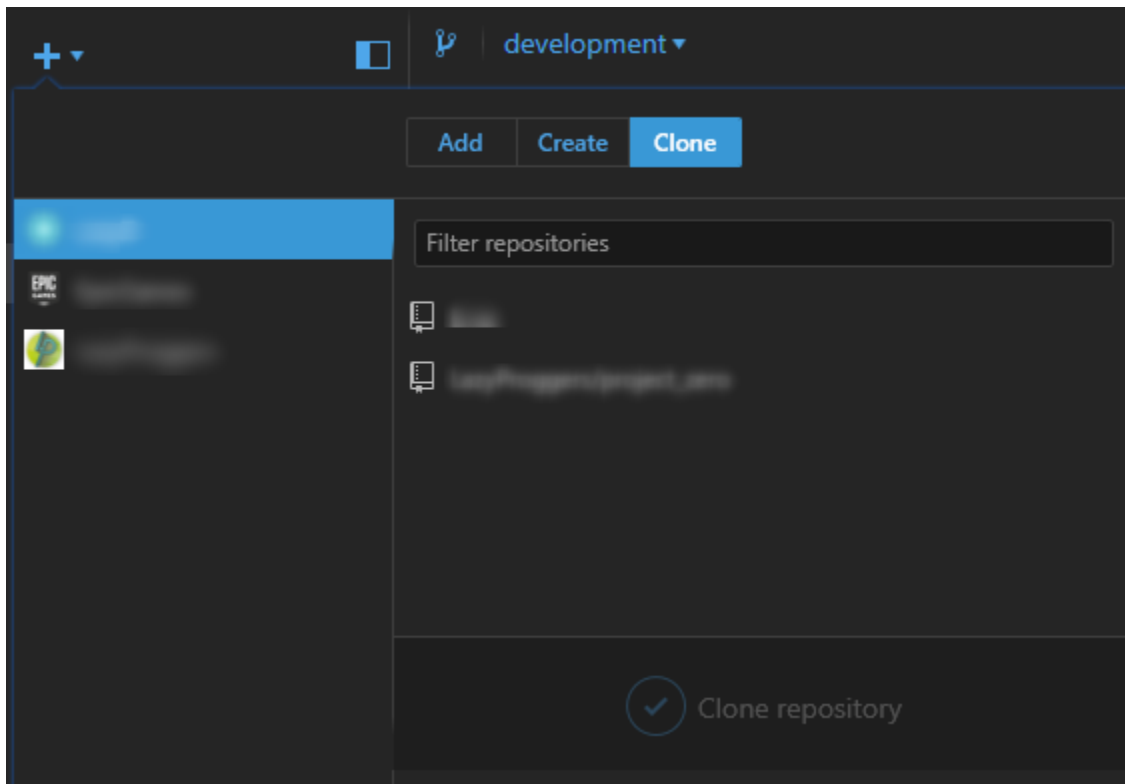
نسخ مشروع موجود مسبقاً Cloning a repository:

يعد النسخ clone باستخدام GitHub Desktop سهلاً جداً كباقي الوظائف: انقر الزر + الأخضر،

حيث تظهر بعض مشاريعك، حدد أحدها أو انسخ رابط أي مشروع آخر تريده.

ملاحظة:

في الإصدارات التجريبية beta لا يتم عرض المشاريع حيث يجب عليك استخدام رابط حصراً



التفرع Branching:

يمكنك تحديد الفرع المطلوب branch من الزاوية العليا اليسرى لواجهة البرنامج، بعد تحديد الفرع الصحيح


يجب نقر زر المزامنة Sync على اليمين، هذه الوظيفة تقوم بتنفيذ الأمر:

`git checkout BRANCHNAME`



في الإصدار الأقدم بإمكانك استعراض فرعين مختلفين معاً ومقارنة نقاط الرفع `pushes` بين الفرعين، بالإضافة إلى إمكانية عرض الخط الزمني للمشروع.

إنشاء فرع جديد:

انقر أيقونة الفرع  أو من القائمة `File > New Branch`، كما بإمكانك تحديد فرع أساسي لإنشاء الفرع الجديد منه بالنقر على اسم الفرع الموجود.

سحب Pull أو رفع Push ملفات المشروع (زر المزامنة Sync):

كما في تعليمات سطر الأوامر، يجب عليك سحب آخر نسخة أو حالة `State` من المشروع من حين لآخر، هذه الوظيفة تسمى مزامنة `Sync` في برنامج `GitHub Desktop`.

وعندما تحدث أي تعديلات على مشروعك محلياً وترغب بتثبيت التعديلات ورفعها إلى نسخة المشروع على موقع `GitHub`، يجب أن تكتب ملخصاً نصياً لوصف طبيعة التعديلات، وتقر `Commit` للفرع المطلوب من مشروعك، والآن تنقر `Sync` لرفع هذا التثبيت إلى الانترنت.

ملاحظة:

بإمكانك استخدام الرموز التعبيرية والإشارات المرجعية في الملخص النصي

إذن فإن زر `Sync` يستخدم من أجل العمليات `Push`, `Pull`, `Checkout`.

الفصل السابع

صفحات GitHub

استخدام مولد الصفحات التلقائي لمشروع معين:

1. انتقل إلى موقع GitHub.
2. افتح مشروعك.
3. انقر Settings.
4. تحت تبويب GitHub Pages انقر Launch Automatic Page Generator.
5. اتبع التعليمات.

إنشاء صفحات من البداية باستخدام Git:

1. أنشئ مشروعاً جديداً أو انسخ مشروعاً موجوداً مسبقاً.
2. أنشئ فرعاً جديداً باسم gh-pages بلا محفوظات سابقة:

```
$ git checkout --orphan gh-pages
# تأكد من أنك في المجلد الصحيح
# احذف جميع الملفات السابقة
$ git rm -rf
```

3. أنشئ ملف index.html على المسار الرئيسي للمشروع:

```
$ echo "Hello World" > index.html
$ git add index.html
$ git commit -a -m "First pages commit"
```

4. ارفع هذه التعديلات إلى GitHub:

```
$ git push origin gh-pages
```

بإمكانك الآن رفع صفحاتك إلى الرابط: `http(s)://<username>.github.io/<projectname>`

إنشاء رابط مخصص لصفحاتك على GitHub:

- ✓ أولاً يجب عليك شراء اسم نطاق Domain name.
- ✓ في فرع gh-pages من مشروعك، أو في الفرع الرئيسي main لمشروع `username.github.io`
- ✓ أنشئ ملفاً باسم CNAME يحتوي اسم النطاق الخاص بك `www.yourdomain.com`.
- ✓ في صفحة إعدادات مزود النطاق قم بربط اسم النطاق برابط موقعك على GitHub.
- ✓ عرف سجلين من نوع CNAME (أحدهما للجذر الرئيسي @ والآخر من أجل `www`).
- ✓ كلاهما يجب أن يشير إلى `username.github.io` أو `username.github.io/repository`.
- ✓ إذا كان مزود نطاق الذي تتعامل معه لا يدعم سجلات ALIAS للجذر @ فالحل بسيط وهو:
- ✓ أنشئ سجل يشير إلى `192.30.252.153` و `192.30.252.154`.

الفصل الثامن

كيف تنشئ إشارة Label وتخصصها؟

تعريف:

إن Labels هي إشارات ملونة يمكن تطبيقها على الملاحظات Issues وطلبات السحب Pull Requests لتأشيرها وتوضيح أية صفات تميزها عن قرائنها مثل صفة الأولوية أو التصنيف أو أي معلومات أخرى قد تجدها مفيدة.

إنشاء إشارة Label مخصصة:

في [هذا الرابط](#) صورة متحركة لشرح إنشاء إشارة وتخصيصها بأبسط صورة ممكنة، وإن لم يعمل لديك الرابط في المستقبل لسبب ما، اتبع الخطوات التالية:

1. تحت اسم المشروع انقر Issues أو Pull request.
2. انقر زر Labels المجاور لحقل البحث.
3. انقر New Label لإضافة إشارة جديدة أو Edit Label لتعديل إشارة موجودة.
4. اكتب اسم الإشارة في حقل الاسم.
5. اختر اللون من حقل الألوان، بإمكانك أيضاً إدخال أي لون بقيمته الست عشرية Hexadecimal.
6. انقر Create Label لحفظ الإعدادات السابقة.

الفصل التاسع الملاحظات Issues

إنشاء ملاحظة:

1. انتقل إلى صفحة المشروع في GitHub الذي تريد إضافة ملاحظة له.
2. انقر Issues.
3. في أعلى اليمين انقر New Issue.
4. أدخل عنوان للملاحظة.
5. أدخل محتوى الملاحظة (متضمناً السجلات logs ومقاطع الأكواد...).
6. (اختياري) لعرض الملاحظة قبل نشرها انقر Preview.
7. لحفظ الملاحظة انقر Submit new issue.

الفصل العاشر

طلبات السحب Pull Request

فتح طلب سحب:

طلب سحب جديد:

عندما تريد إنشاء طلب سحب (لنقل إنه إما لآخر تعديل على المشروع أو حتى لتعديل أقدم) بإمكانك ترك GitHub يقوم بالكثير عنك، قم فقط بنقر زر Compare & Pull request الأخضر في الأعلى (حاول ألا تخطئ بسبب التشابه بينه وبين زر Clone or download) ثم أدخل المعلومات في نافذة طلب السحب وانقر Create pull request.

عدا ذلك انقر زر New pull request تجده قرب اسم الفرع في الأعلى.

Repository used for Documentation in Stack Overflow — Edit

2 commits
2 branches
0 releases
1 contributor

Your recently pushed branches:

example-branch (less than a minute ago)
Compare & pull request

Branch: example-branch
New pull request
Create new file
Upload files
Find file
Clone or download

This branch is 1 commit ahead of master.
Pull request
Compare

maxcell committed on GitHub Update README.md
Latest commit 4029e0a just now

README.md
Update README.md
just now

Update README.md

Write
Preview
AA B i

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported
Create pull request

الفصل الحادي عشر

إزالة البيانات الحساسة والملفات الضخمة

مقدمة

إذا قمت بتثبيت التغييرات وكانت تحوي بيانات حساسة مثل كلمات المرور أو مفاتيح التشفير الخاصة بإمكانك إزالتها من سجلات GitHub، لحذف أية ملفات غير مرغوبة بشكل نهائي قم إما باستخدام الأمر `git filter-branch` أو باستخدام تقنية BFG Repo-Cleaner.

ملاحظات

1. اطلب من كل من يعملون معك بالمشروع أن يعيدوا تأسيس (ولا يدمجوا merge) أي فرع أنشؤوه من سجل تنبيلات مشروعك القديم (الذي يحوي الملفات غير المرغوبة)، إن تثبيت commit دمج واحد فقط، بإمكانه إعادة بعض أو كل الملفات غير المرغوبة التي تقوم بحذفها إلى سجل المشروع.
2. بعد مرور بعض الوقت وعندما تتأكد من أن الأمر `git filter-branch` ليس له أية آثار جانبية غير مقصودة، عندها يمكنك إجبار إلغاء مؤشرات جميع العناصر في مشروعك المحلي من أجل حذفها باستخدام الأوامر التالية:

```
git for-each-ref --format='delete %(refname)' refs/original | git
update-ref -stdin

git reflog expire --expire=now -all

git gc --prune=now
```

أمثلة:

استخدام filter-branch

```
git filter-branch --force --index-filter \
'git rm --cached --ignore-unmatch PATH-TO-FILE-WITH-SENSITIVE-DATA' \
--prune-empty --tag-name-filter cat -- --all
```

أضف الملفات غير المرغوبة إلى ملف .gitignore. لتتأكد من عدم تثبيتها ثانية مع المشروع.

```
echo "YOUR-FILE-WITH-SENSITIVE-DATA" >> .gitignore
git add .gitignore
git commit -m "Add YOUR-FILE-WITH-SENSITIVE-DATA to .gitignore"
```

بعد تثبيت مشروعك ارفع نسخة اجبارية forced-push إلى GitHub

```
git push origin --force --all
```

أيضاً حتى تحذف بياناتك الحساسة من الإصدارات tagged يجب عليك أن ترفع نسخة اجبارية:

```
git push origin --force --tags
```

استخدام منظم المشاريع BFG

BFG هو برنامج بديل لـ filter-branch. يمكن استخدامه لإزالة البيانات الحساسة أو الملفات الضخمة التي

أضيف خطأً إلى المشروع كالملفات التنفيذية وغيرها. البرنامج مصمم باستخدام Scala.

موقع البرنامج: [BFG Repo Cleaner](#)

يتطلب تشغيل البرنامج تنصيب Java JRE إضافة إلى تنصيب مكتبة Scala وكل اعتمادياتها.

استخدام البرنامج:

```
bfg --delete-files YOUR-FILE-WITH-SENSITIVE-DATA
```


الفصل الثاني عشر

تحديث مشروع منسوخ Forked Repository

ملاحظات

✓ من تعليمات GitHub: [تجهيز مشروع للنسخ](#).

✓ من تعليمات GitHub: [مزامنة النسخة](#).

✓ [سؤال مشهور على StackOverFlow](#).

أمثلة:

1. تجهيز remote للنسخ:

```
$ cd my local repo
$ git remote add upstream
https://github.com/ORIGINAL OWNER/ORIGINAL REPOSITORY.git
# Specify a new remote upstream repository that will be synced with
the fork
$ git remote -v
# Verify the new upstream repository specified for my fork
```

2. مزامنة النسخة محلياً:

```
$ cd my_local_repo
$ git fetch upstream
# Fetch the branches and their respective commits from the upstream
repository
# Commits to master will be stored in a local branch, upstream/master
$ git checkout master
$ git merge upstream/master
# Merge the changes from upstream/master into your local master
branch
# brings your fork's master branch into sync with the upstream repo
```

3. مزامنة النسخة مع GitHub:

```
$ git push origin master
```

الفصل الثالث عشر

استخدام Gist

مقدمة

تعد Gist طريقة ممتازة لمشاركة عملك، بإمكانك مشاركة ملفات مفردة أو أجزاء منها أو حتى

تطبيقات كاملة، تصل إليه من خلال الرابط <https://gist.github.com>

كل gist هو مشروع Git أي من الممكن نسخه وربطه محلياً، تم نشر محرر gist من قبل CodeMirror.

هناك نوعين لـ gist، عام وسري، إضافة إلى ذلك بإمكانك إنشاء gist كمجهول إن لم تكن مسجلاً الدخول

إلى GitHub.

العام Public Gists

تظهر في قائمة Discover حيث يمكن للناس تصفح gists الجديدة بالترتيب وفق إنشائها، وهي قابلة

للبحث، لذلك يمكنك استخدام هذه الميزة للناس التي تريدها أن تبحث وترى عملك.

السرية Secret Gists

لا تظهر في قائمة Discover وغير قابلة للبحث، لذلك يمكنك استخدام هذه الميزة لتدوين أفكارك الخاصة،

أو قوائم المهام أو تجهيز أكواد أو أشياء لا تنوي مشاركتها مع الناس.

المجهولة Anonymous Gists

إذا أنشأت gist بدون تسجيل الدخول إلى GitHub ستكون gist مجهولة، يمكن أن تكون عامة أو سرية،

لحذف أية gist مجهولة تستطيع التواصل مع دعم GitHub وتزويدهم بالرابط.

أمثلة

العامّة Public Gists

يمكن أن تكون أي شيء تقريباً، كمثال بسيط يمكن تخزين دالة javascript:

```
function randomInt(min, max) {  
  return Math.floor((max - min + 1) * Math.random()) + min;  
}
```

السريّة Secret Gists

تستخدم لأي شيء لا تريد مشاركته مع العامة، مثل تخزين المفاتيح الخاصة أو المعلومات الخاصة بشكل

عام، وكمثال بسيط يمكن تخزين كود JSON يحوي سرية:

```
{  
  "id": AKIAIOSFODNN7EXAMPLE,  
  "secret": wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY  
}
```

الفصل الرابع عشر

استخدام أزرار GitHub

مقدمة

ما هي أزرار GitHub؟ هي أزرار يمكنك إضافتها إلى موقعك الخاص لتحويل الزوار مباشرة إلى مشروعك على GitHub.

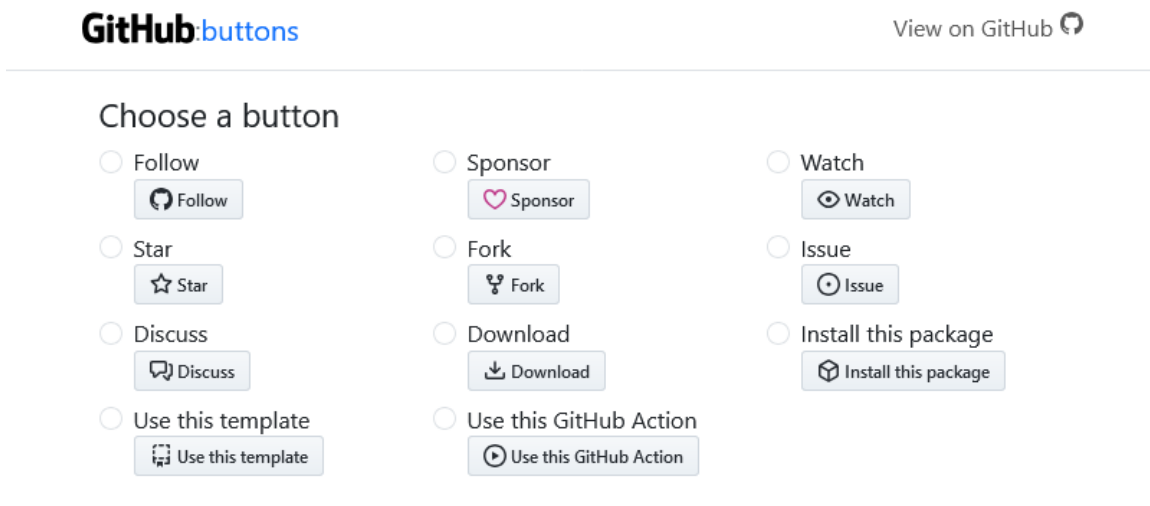
أمثلة

زر المتابعة Follow Button

هو زر مرتبط بصفحتك على GitHub لترويجها وحث المستخدمين على متابعتها، هنا مثال على إنشائه:

1. انتقل إلى <https://buttons.github.io>

2. انقر Follow



3. أدخل اسم المستخدم الخاص بك على GitHub في خانة user.

4. خصص شكل الزر كما ترغب (الحجم، الأيقونة، إظهار العدد)

Button options

bakrialloush

/

:repo

☒ Color scheme

light

light

dark

@media (prefers-color-scheme: light)

@media (prefers-color-scheme: dark)

☒ Large button

☐ Show count
 ☐ Standard icon

Syntax

html

Preview and code

Try out your button, then copy and paste the code below into the HTML for your site.



5. ضع كود javascript في صفحة موقعك في <HEAD> أو قبل نهاية <BODY>.

```
<!-- Place this tag in your head or just before your close body tag. -->
<script async defer src="https://buttons.github.io/buttons.js"></script>
```

6. ضع كود الزر المخصص في صفحة موقعك في المكان الذي تريده أن يظهر.

```
<!-- Place this tag where you want the button to render. -->
<a class="github-button" href="https://github.com/bakrialloush" data-color-scheme="no-preference: light;
light: light; dark: dark;" data-size="large" data-text="Follow @bakrialloush" data-type="follow">
```

الفصل الخامس عشر

العمل بـ Gitflow

الأوامر

- `git flow <subcommand>`
- `git flow init`
- `git flow [feature|release|hotfix] [start|finish]`

البرامترات

| الأمر الفرعي | التفاصيل |
|--------------|--|
| init | تهيئة مشروع جديد يدعم التفرع .branching. |
| feature | إدارة أفرع نسخ الميزات. |
| release | إدارة أفرع نسخ الإصدارات. |
| hotfix | إدارة أفرع نسخ الصيانة. |

ملاحظات

✓ [أساسيات Gitflow من مؤسسها.](#)

✓ [صورة نموذج التفرع.](#)

أمثلة

العمليات على 5 أفرع نشأعة محلياً

أكثر حالات استخدام Gitflow شيوعاً

1. تهيئة المشروع وتعريف الأفرع:

```
$ git flow init
# if you use default setup, you'll define six types of branches:
#
# main branches (lives forever)
#
# 1. master: for production releases
# 2. develop: for "next release" development
#
# supporting branches
#
# 3. feature: for a product feature
# 4. release: for preparation of a new production release
# 5. hotfix: for resolving critical bug of production version
# 6. Support
#
# also, two main branches are created: master, develop
```

2. بدء وإنهاء Feature.

```
$ git flow feature start my feature
# create branch 'feature/my_feature' based on the 'develop'
# made development and commits...
$ git flow feature finish my feature
# merge 'feature/my_feature' back to the 'develop'
# delete 'feature/my feature'
```

3. بدء وإنهاء Release.

```
$ git flow release start my_release
# create branch 'release/my_release' based on the 'develop'
# made bug fixes...
$ git flow release finish my_release
# merge branch 'release/my_release' to the 'master' and add tag
# merge branch 'release/my release' back to the 'develop'
# delete 'release/my_release'
```

4. بدء وإنهاء Hotfix.

```
$ git flow hotfix start my_hotfix
# create branch 'hotfix/my_hotfix' based on the 'master'
# made some hotfixes...
$ git flow hotfix finish my_hotfix
# merge branch 'hotfix/my_hotfix' back to the 'master' and add tag
# merge branch 'hotfix/my_hotfix' to the 'develop'
# delete 'hotfix/my_hotfix'
```

بفضل الله

مَشَّ