

# GitHub

- GitHub هي منصة لإدارة الكود باستخدام Git.
- يستخدمها المطوروون لحفظ المشاريع، العمل الجماعي، ومشاركة الأكواد.
- للطلاب: وسيلة لعرض مشاريعهم ومهاراتهم بشكل احترافي.
- للشركات: معيار عالمي لمتابعة العمل البرمجي.

## 1. إنشاء حساب GitHub

1. ادخل على الرابط <https://github.com>
  2. اضغط **Sign Up** وأنشئ حساب باسمك.
  3. فعال حسابك عبر البريد الإلكتروني.
  4. جهز **Profile** بوضع صورة وكتابة نبذة بسيطة عنك.
- 

## 2. إنشاء مستودع Repository جديد :

المستودع أو **Repository** هو المكان الذي نخزن فيه كل ملفات المشروع والكود الذي نكتبه على GitHub

المستودع صندوق كبير يحتوي على كل شيء يخص المشروع: ملفات الكود، الصور، المستندات، وأي تغييرات تجريها. نعمل المستودع لكي نسهل متابعة المشروع، نحافظ على النسخ المختلفة من العمل، ونشارك المشروع مع الآخرين.

### 1. اضغط **New Repository**

2. اختر اسم المشروع باسمك أو فكرة المشروع.
3. اختر نوع المستودع:

- **Public** مرئي للجميع
- **Private** خاص بك أو بفريقك فقط

4. ضع **README.md** لشرح المشروع.

## 5. اضغط Create repository

### 3. رفع الملفات والمجلدات

1. جهز ملفات مشروعك على جهازك (html,css,php,....)

2. في صفحة المستودع، اضغط Add file --> Upload files.

3. اسحب الملفات أو اخترها من جهازك.

4. اضغط Commit changes.

### 4. التعامل مع الكود على جهازك :

عندما نريد العمل على مشروع موجود على GitHub من جهازنا الشخصي، نبدأ أولاً بـ تحميل المشروع باستخدام الأمر git clone. وهذا الأمر ينقل نسخة من كل ملفات المشروع إلى جهازك، لتمكن من تعديلها محلياً. بعد إجراء أي تغييرات على الملفات، نستخدم الأمر git add لتحديد الملفات التي نريد حفظ التعديلات عليها، ثم git commit لإضافة وصف مختصر لهذه التعديلات، وأخيراً git push لرفع التغييرات إلى المستودع على GitHub. أما إذا قام شخص آخر بإجراء تعديلات على المشروع منذ آخر مرة عملت فيها، نستخدم الأمر git pull لتحديث نسخة المشروع على جهازنا بأحدث التغييرات. يُنصح باستخدام هذه الأوامر دائمًا عند العمل على المشاريع المشتركة، لأنها تضمن أن كل شخص يعمل على أحدث نسخة من المشروع وتحافظ على تنظيم التعديلات وتتجنب فقدان أي جزء من العمل.

تحميل المشروع إلى جهازك

```
git clone https://github.com/username/repository-name.git
```

حفظ التعديلات ورفعها

```
git add filename.py      # إضافة كل الملفات .
```

```
git commit -m "شرح أو تعديل معين"
```

```
git push origin main
```

تحديث المشروع من GitHub

```
git pull origin main
```

## 5. العمل على فرع Branch جديد :

عندما نريد تجربة ميزة جديدة في المشروع دون التأثير على الكود الأساسي، ننشئ فرعاً جديداً باستخدام الأمر `git checkout -b`، بحيث يمكننا تطوير الميزة والتعديل على الملفات بأمان. بعد الانتهاء من التعديلات والتأكد من أنها تعمل بشكل صحيح، ندمج هذا الفرع مع الفرع الرئيسي باستخدام الأمر `git merge`. قد يحدث أن يقوم شخص آخر أو أنت بتعديل نفس السطر في فرعين مختلفين، عندها يظهر تعارض `Merge Conflict`. في هذه الحالة، يوضح Git أي الملفات تحتوي على تعارض، ويجب فتح الملف المتعارض، تعديل ما يلزم لحل المشكلة، ثم حفظ التعديلات باستخدام `git add` و`git commit`. يُنصح به دائماً لأنه يحافظ على استقرار المشروع ويسمح بتجربة أفكار جديدة بأمان قبل دمجها مع الكود الأساسي.

### 1. لإنشاء فرع لتجربة ميزة جديدة:

```
git checkout -b feature-new-login
```

### 2. بعد الانتهاء، دمج التغييرات:

```
git checkout main
```

```
git merge feature-new-login
```

### ► التعامل مع التعارضات Merge Conflicts

- يحدث عند تعديل نفس السطر في فرعين مختلفين.
- يظهر Git رسالة مثل:

CONFLICT (content): Merge conflict in index.html

- افتح الملف المتعارض، عدل، ثم:

```
git add index.html
```

```
" حل تعارض الدمج "
```

## 6. إنشاء Pull Request

بعد الانتهاء من التعديلات على الفرع الخاص بك، نبدأ بإنشاء Pull Request لاقتراح دمج التغييرات مع الفرع الرئيسي في المستودع. أول خطوة هي دفع التعديلات إلى فرعك باستخدام `git push origin`، ثم في

نضغط على GitHub Compare & pull request، ونضيف عنواناً ووصفًا يوضح ما أضفناه أو عدّلناه. بعد ذلك، يقوم الفريق بمراجعة الكود أو ما يعرف بـ Code Review، حيث يقرأ أعضاء الفريق التعديلات سطراً بسطراً، ويضيفون تعليقات أو يتطلبون تغييرات إذا لزم الأمر. بعد الموافقة على التعديلات، يتم دمج الفرع مع الفرع الرئيسي باستخدام أحد أساليب الدمج: الدمج المباشر Merge، أو دمج كل التعديلات في كوميت واحد Squash، أو إعادة ترتيب التعديلات Rebase. هذه العملية تضمن أن يكون الكود النهائي منظماً وأمناً قبل إضافته إلى المشروع الأساسي.

#### 1. ادفع التعديلات لفرعك:

```
git push origin feature-login
```

#### 2. في GitHub ، اضغط Compare & pull request

#### 3. أضف عنواناً + وصفًا Create pull request

#### 4. الفريق يراجع الكود Code Review، يضيف تعليقات أو يتطلب تغييرات

#### 5. بعد الموافقة، يتم الدمج Merge

- Merge مباشر
  - (دمج كل التعديلات في كوميت واحد) Squash
  - (إعادة ترتيب التعديلات) Rebase
- 

### 7. المشاريع المشتركة بين الطلاب

في المشاريع المشتركة بين الطلاب، نبدأ بإنشاء مستودع رئيسي Main Repo يحتوي على جميع ملفات المشروع الأساسية. يمكننا منح الصلاحيات للأعضاء المشاركين أو جعل المستودع عاماً ليتمكن الجميع من الاطلاع عليه. للمساهمة في المشروع، يقوم كل طالب بعمل Fork للمستودع إلى حسابه الشخصي، ثم يجرّب التعديلات على فرعه الخاص ويرفعها باستخدام git push. بعد ذلك، ينشئ Pull Request لاقتراح دمج التعديلات مع المستودع الرئيسي. من المهم متابعة المشروع بشكل مستمر، لذلك يُنصح بتنشيط Notifications لتلصّل أي تغييرات، ومراقبة Issues وPull Requests لضمان التعاون السلس بين جميع الأعضاء والحفاظ على تنظيم المشروع.

#### 1. إنشاء مستودع رئيسي Main Repo

#### 2. أعطِ الصلاحيات للأعضاء أو اجعله عام

#### 3. المساهمة عبر Fork + Pull Request:

- اعمل **Fork** للمستودع إلى حسابك
- عدّل الكود وادفع التغييرات إلى فرعك
- أنشئ **Pull Request** لاقتراح الدمج

4. متابعة التحديثات:

- فُعل **Notifications** لاتصالك التغييرات
- راقب **Pull Requests** أو **Issues**

## 8. كتابة المستندات Documentation

### الوثيق Documentation

لكل مشروع على GitHub أهمية أن يكون موثقاً بشكل جيد ليسهل على الآخرين فهمه والعمل معه. README.md هو الملف الأساسي الذي يشرح فكرة المشروع، خطوات تشغيله، ويحتوي على صور أو أمثلة توضيحية. كما يمكن إضافة LICENSE لتحديد حقوق استخدام المشروع، مثل MIT أو Apache أو GPL، حتى يعرف الآخرون ما يمكنهم فعله بالكود. بالنسبة للمشاريع الكبيرة، يمكن استخدام Wiki لتوفير توثيق مفصل وأكثر تنظيماً. بشكل عام، README الجيد يعتبر بطاقة التعريف الأولى لمشروعك أمام أصحاب العمل أو المساهمين، ويترك انطباعاً احترافياً عن جودة العمل.

### تفعيل GitHub Pages

لمن يريد عرض مشروعه على الإنترنت بشكل مباشر، يمكن تفعيل GitHub Pages. يتم ذلك بالدخول إلى إعدادات المستودع **Settings** → **Pages** → **Settings** الفرع الرئيسي main والمجلد /root، والضغط على **Save**. بعد ذلك، يعطيك GitHub رابطاً لموقعك مثل:

/ <https://<username>.github.io/<repository>

أي تعديل أو تحديث تقوم برجاته على الفرع الرئيسي main يتم عرضه مباشرة على الموقع، مما يسمح بمشاركة مشروعك على الإنترنت بسهولة وسرعة، سواء مع الأصدقاء أو مع أي شخص مهتم بالمشروع.

- **README.md**: يشرح فكرة المشروع، خطوات التشغيل، صور وأمثلة
  - **LICENSE**: (مثل MIT, Apache, GPL) لتحديد حقوق الاستخدام
  - **Wiki**: لتوثيق أكبر للمشاريع الكبيرة
- README: الجيد هو بطاقة التعريف لمشروعك أمام أصحاب العمل أو المساهمين.

## .....

### 9. تفعيل GitHub Pages لعرض المشروع

1. في المستودع، اذهب إلى Settings --> Pages

2. اختر:

- Branch: main
- Folder: /root

3. اضغط Save

4. سيعطيك GitHub رابط موقعك:

<https://<username>.github.io/<repository>/>

5. أي تغيير ترفعه على main branch يتم تحديثه مباشرة على الموقع.