**مراحل إستخدام github:**

* إنشاء حساب في github.
* Create new repo
* نسخ رابط repo
* إنشاء مجلد جديد من خلال الأمر .mkdir My-Github
* فتح المجلد من خلال الأمر cd My-Github.
* ثم إنشاء clone من خلال الأمر التالي: git clone <https://github.com/karimlearns/zero_formation.git>
* أنشأنا مجلدات css و js و images وملف index.html وأنشأنا داخل css ملف main.css.
* فتحنا مجلد المشروع من خلال الأمر cd My\_Github
* فتحنا ملف المشروع من خلال الأمر cd zero\_formation
* أضفنا الملفات من خلال الأمر git add css\main.css index.html
* حدفنا ملف من خلال الأمر git reset head osama.txt
* قمنا بإضافة commit من خلال الأمر التالي: "comment"git commit –m
* قمنا بإدفع الملفات من خلال الأمر التالي: git push origin main
* لإضافة شخص ل repo ندخل ل setting ثم Collaborators ثم Add People.
* يستقبل الشخص المضاف email ليقبل الطلب.
* يمكننا جلب ملف من Github من خلال الأمر التالي: git pull origin

**Git Configuration**

* يمكننا الحصول على قائمة config من خلال الأمر التالي: git config –l
* يمكننا الحصول على كافة configuration من خلال الأمر التالي: git help config
* يمكننا الحصول على على email من خلال الأمر التالي: git config --global user.email وللتعديل عليها: "karim@gmail.com"git config --global user.email
* لحدف config نكتب الأمر التالي: git config –global –unset user.name

**للتعديل على config من خلال editor:**

* نكتب الأمر التالي في cmd: git config –-global –-edit
* لإنشاءملف جديد نكتب الأمر touch new\_file.txt

**Generate Public Key**

* لعمل public key نكتب الأمر التالي: ssh-keygen –t rsa –b 4096 –C ‘[tahari.abdelkarim@gmail.com](mailto:tahari.abdelkarim@gmail.com)’
* ندخل password لن يظهر في الشاشة، ونعيد تأكيده.
* نكتب الأمر cat ~/.ssh/id\_rsa.pub )يمكننا الأمر cat من معرفة محتوى الملف(.
* ننسخ المحتوى ثم ندخل على setting في Github ثم SSH and GPG keys ثم new SSH key ونلصق المحتوى في key ونكتب عنوان في title ونضغط على add SSH key.
* نكتب في cmd التالي: ssh –T [git@github@gmail.com](mailto:git@github@gmail.com) ثم نكتب password وبدلك ندخل ل Github.
* لإنشاء ملف جديد touch index.html.
* نقوم بكل إجراءات add و commit.
* نقوم بنسخ git remote add origin [git@github.com:karimlearns/course.git](mailto:git@github.com:karimlearns/course.git) من SSH ونلصقه في cmd.
* نقوم بنسخ git push –u origin main ونلصقه في cmd.

**Video-10 : Pull Request in main and in branch**

* **الإختصارات :Aliases**
* نكتب في cmd git config –global alias.st status
* إدا كتبنا git st كأننا كتبنا git status
* **Branching And Merging:**
* ننشئ branch جديد من خلال الأمر التالي: git branch scroll
* نتحول ل branch من خلال الأمر التالي: git checkout scroll
* نمسح branch من خلال الأمر التالي: git branch –d scroll
* من خلال الأمر التالي يمكننا إنشاء branch والدخول له: git checkout –b Dev-Feature
* لإعادة تسمية branch نكتب: git branch –m Development
* لعمل merge ندخل أولا ل main من خلال الأمر التالي: git checkout master ثم أمر merge التالي: git merge Development

# Stash –

# من خلال الأمر git stush يمكننا أن نحتفظ بالملفات أو نضعها على جنب بعد القيام بأمر git add.

# يمكننا إرجاعه من خلال الأمر git stash pop بدون أن يبقى الملف في status

# يمكننا إرجاعه من خلال الأمر git stash apply ويبقى الملف في status

* لإرجاع ملف بشكل محدد نكتب: git stash pop stash@{2}
* نحدف ملف من **stush من خلال الأمر التالي: git stush drop stash{1}**
* لمعرفة مادا يوجد في الملف نكتب الملف التالي: stash@{0} git stash show
* من خلال الأمر git stash clear لحدف كل الملفات

# Restore And Clean

* من خلال الأمر git restore –staged new.txt يمكننا الرجوع إلى الوضع قبل مرحلة add
* من خلال الأمر git clean –n يمكننا معرفة الملفات التي ستحدف.
* من خلال الأمر git clean –f يمكننا حدف الملفات.

# Resetting The Head

* Head هي اخر commit من خلال الأمر git log يعطينا commit head
* لنحدف commit نكتب الأمر التالي: git reset –hard e5656646f66565g546h446

# Ignoring Files And Directories

* إدا أردنا أن يتجاوز git ملف ما أو مجلد ما، ننشئ الملف .gitignore ثم نكتب الأمر code .gitignore وفي vsc نكتب \*.log وبدلك سيتجاوز github أي ملف بإمتداد .log، ولإستثناء ملف من هدا التجاوز نكتب في vsc !vip.log.
* يمكن أن تضيف ملف عملت له ignore بالطريقة التالية: git add –f t.txt

# Tagging And Releasing

* لعمل tag نكتب التالي في cmd: git tag v1.0 ونقوم بالأمر التالي: git push origin v1.0
* git tag –a v2.0 –m "Our Project" => Second Version ثم git push origin v2.0
* للحصول على tags نكتب الأمر التالي: git tag –l " v1.\*"
* لعمل release نتجه ل github وننشئ release في tags ونملئ الخانات المطلوبة.
* لحدف tag نكتب الأمر التالي: git tag –d v1.0