

git

les Bases pour introduire Git

Gherghouch Abdessalam



TABLE DE MATIÈRE

INTRODUCTION	-----	Page 3
INSTALLATION	-----	Page 4
INITIALISATION DU COMMIT	-----	Page 7
COMMENCER À VERSIONNER	-----	Page 8
CYCLE DE TRAVAIL	-----	Page 9
HISTORIQUE DES CHANGEMENTS	-----	Page 10
GITHUB	-----	Page 12
GIT ET GITHUB	-----	Page 14
JAVA DOC	-----	Page 16
COMMENTAIRE	-----	Page 18
DOCUMENTATION	-----	Page 22

TABLE DE FIGURE

<u>figure 1-1:git.exe</u>	Page 4
<u>figure 2-1:commande_line1</u>	Page 7
<u>figure 2-2:commande_line2</u>	Page 8
<u>figure 2-3:commande_line3_historique</u>	Page 10
<u>figure 2-4:partie_graphique</u>	Page 11
<u>figure 3-1:github_upload_files</u>	Page 14
<u>figure 3-2:github_repositories_and_commande_line</u>	Page 15
<u>figure 4-1: exemple_de_commentaire</u>	Page 18
<u>figure 4-2: exemple_de_commentaire_linéaire</u>	Page 19
<u>figure 4-3: exemple_de_commentaire_multilignes</u>	Page 19
<u>figure 4-4: exemple_de_commentaire_avec_javaDoc</u>	Page 21
<u>figure 4-5: exemple_de_commentaire_avec_javaDoc2</u>	Page 21
<u>figure 5-1: etap1_documentation</u>	Page 22
<u>figure 5-2: etap2_documentation</u>	Page 22
<u>figure 5-3: javadoc_file</u>	Page 22
<u>figure 6-1: inde_html_documentation1</u>	Page 23
<u>figure 6-2: index_html_documentation2</u>	Page 23
<u>figure 6-3: index_html_documentation3</u>	Page 24

INTRODUCTION



Git est de loin le système de contrôle de version le plus largement utilisé aujourd'hui. Git est un projet open source avancé, qui est maintenu. À l'origine, il a été développé en **2005** par **Linus Torvalds**, le créateur bien connu du noyau du système d'exploitation Linux. De plus en plus de projets logiciels déposés sur Git pour le **contrôle de version**,

y compris des projets commerciaux et en open source. Les développeurs qui travaillent avec Git sont bien représentés dans le vivier de talents disponibles, et la solution fonctionne bien sur une vaste gamme de systèmes d'exploitation et d'environnements de développement intégrés (IDE).

Git considère ses données plus comme une série d'instantanés d'un système de fichiers miniatures. Avec Git, chaque fois que vous validez ou enregistrez l'état de votre projet, Git prend essentiellement une photo de ce à quoi ressemblent tous vos fichiers à ce moment et stocke une référence à cet instant.

INSTALLATION

[Installer Git sur Windows](#)

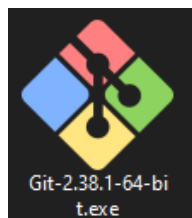
[Installer Git sur Mac OS](#)

[Installer Git sur Linux](#)

il existe plusieurs methodes pour inataller Git sur windows :

installer la dernière version du git : [cliquer ici pour l'installer](#)

figure 1-1:git.exe



installer avec l'outil Winget : [Installez l'outil Winget](#)

tapez cette commande dans l'invite de commande ou Powershell.

```
winget install --id Git.Git -e --source winget
```

INSTALLATION

[Installer Git sur Windows](#)

[Installer Git sur Mac OS](#)

[Installer Git sur Linux](#)

il existe plusieurs methodes pour inataller Git sur Mac OS:

Homebrew :

[cliquer ici pour l'installer](#)

tapez cette commande dans l'invite de commande ou Powershell.

```
$ brew install git
```

Homebrew :

[cliquer ici pour l'installer](#)

tapez cette commande dans l'invite de commande ou Powershell.

```
$ brew install git
```

INSTALLATION

[Installer Git sur Windows](#)

[Installer Git sur Mac OS](#)

[Installer Git sur Linux](#)

L'exécution de Git sur Linux est plus facile en utilisant le gestionnaire de paquets préféré dans la distribution Linux.

Debian/Ubuntu :

tapez cette commande dans l'invite de commande la dernière version du Git

```
apt-get install git
```

INITIALISATION DU COMMIT


Concepts de base:

- **Commit**

Un commit est l'équivalent Git d'un enregistrement. Ce dernier devrait être considéré comme une opération sur un système de fichiers,

Il y'a Deux paramètres utiles pour identifier l'auteur des commit :

- `git config --global user.name "<your name>"`
- `git config --global user.email "<your@mail.com>"`

A terminal window with a black background and white text. The prompt is 'Dell@DESKTOP-3RC2HAV MINGW64 ~'. The first command is '\$ git config --global user.name "DDOS"'. The second command is '\$ git config --global user.email "abde.gher9@gmail.com"'.

```
Dell@DESKTOP-3RC2HAV MINGW64 ~  
$ git config --global user.name "DDOS"  
  
Dell@DESKTOP-3RC2HAV MINGW64 ~  
$ git config --global user.email "abde.gher9@gmail.com"
```

figure 2-1:commande_line1

COMMENCER À VERSIONNER

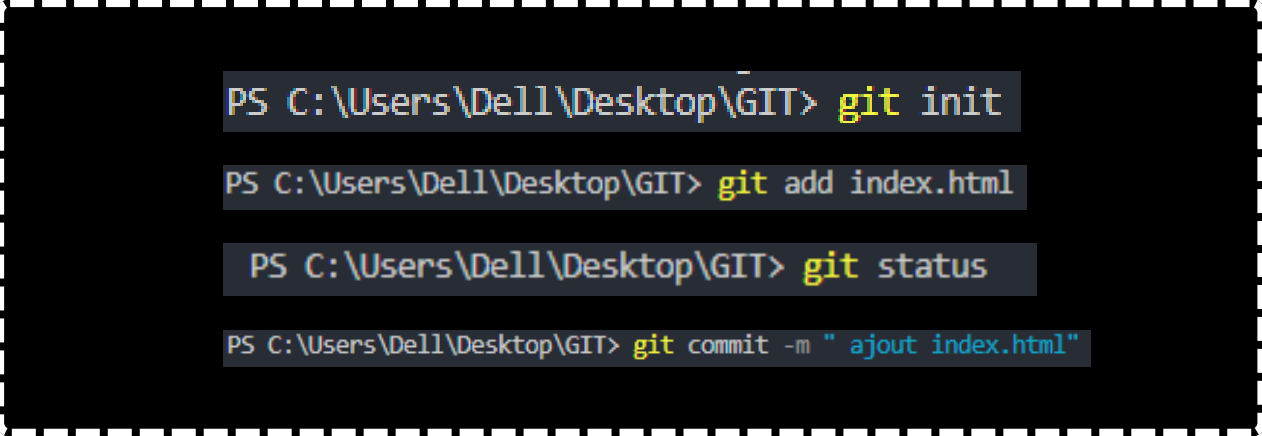
nous avons déjà des fichiers pour ce projet:

- **CAS :** nous avons déjà des fichiers pour ce projet

Un commit est l'équivalent Git d'un enregistrement. Ce dernier devrait être considéré comme une opération sur un système de fichiers,

Il y'a Deux paramètres utiles pour identifier l'auteur des commit :

- `cd <dossier_existant>`
- `git init`
- `git add .`
- `git status`
- `git commit -m " nom_de _Commit"`



```
PS C:\Users\Dell\Desktop\GIT> git init
PS C:\Users\Dell\Desktop\GIT> git add index.html
PS C:\Users\Dell\Desktop\GIT> git status
PS C:\Users\Dell\Desktop\GIT> git commit -m " ajout index.html"
```

figure 2-2:commande_line2

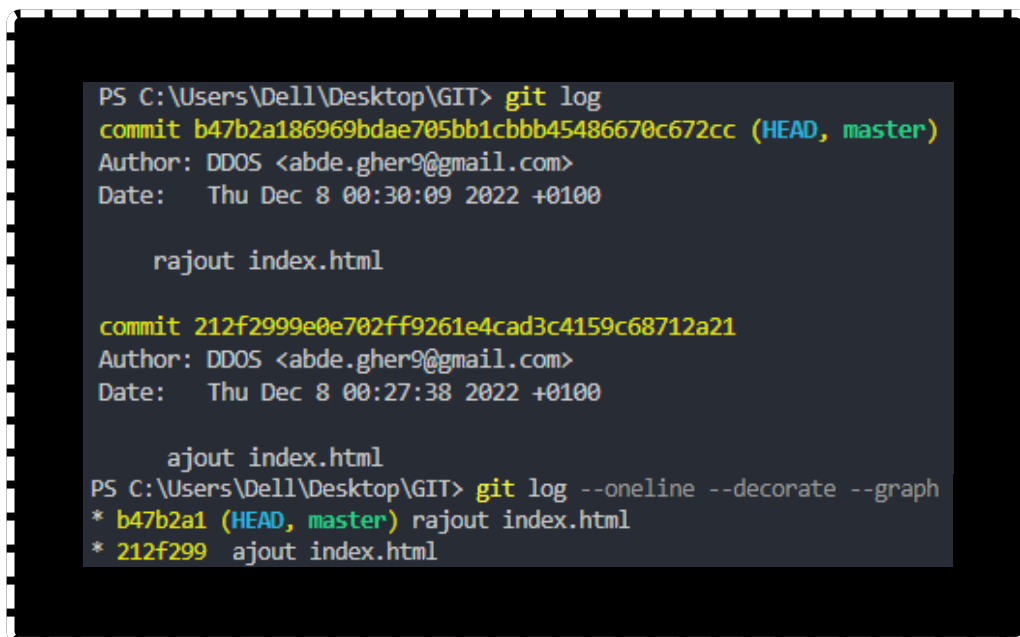
CYCLE DE TRAVAIL

- Cycle de Travail au niveau Locale
 - **Ajout D'un Fichier**
 - **git status** : affiche l'état du répertoire de travail
 - **git add** : pour enregistrer un instantané de l'état actuel du projet dans l'historique des commits
 - **git commit**
 - **Modifier un Fichier**
 - **git diff** : voir les modifications
 - **git add** : pour enregistrer un instantané de l'état actuel du projet dans l'historique des commits
 - **git commit**
 - **Supprimer un fichier**
 - **git rm** : utilisée pour supprimer des fichiers individuels ou une série de fichiers.
 - **git checkout** : -- <nom_du_fichier_supprimé>
 - **Déplacer un fichier**
 - **git-mv** : utilisée pour Déplacer ou renommer un fichier, un répertoire

HISTORIQUE DES CHANGEMENTS

Historique des Commits

- **Avec ligne de commande**
 - **git log** : utilisé pour afficher l'historique des modifications validées
 - **git log --oneline --decorate --graph**



```
PS C:\Users\Dell\Desktop\GIT> git log
commit b47b2a186969bdae705bb1cbbb45486670c672cc (HEAD, master)
Author: DDOS <abde.gher9@gmail.com>
Date: Thu Dec 8 00:30:09 2022 +0100

    rajout index.html

commit 212f2999e0e702ff9261e4cad3c4159c68712a21
Author: DDOS <abde.gher9@gmail.com>
Date: Thu Dec 8 00:27:38 2022 +0100

    ajout index.html
PS C:\Users\Dell\Desktop\GIT> git log --oneline --decorate --graph
* b47b2a1 (HEAD, master) rajout index.html
* 212f299 ajout index.html
```

figure 2-3:commande line3_historique

HISTORIQUE DES CHANGEMENTS

Historique des Commits

- **Avec outil Graphique**
 - `gitk --all`

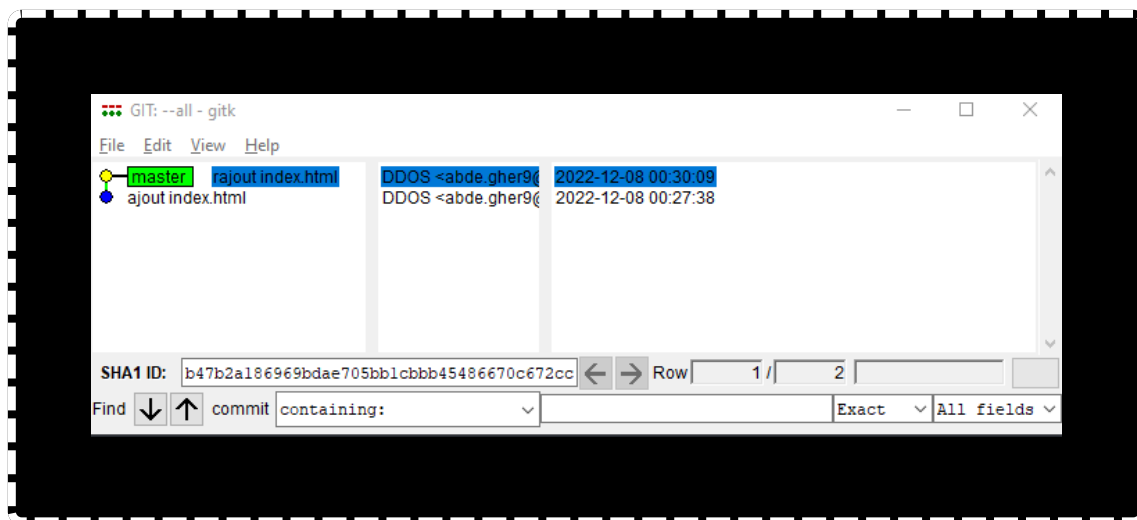
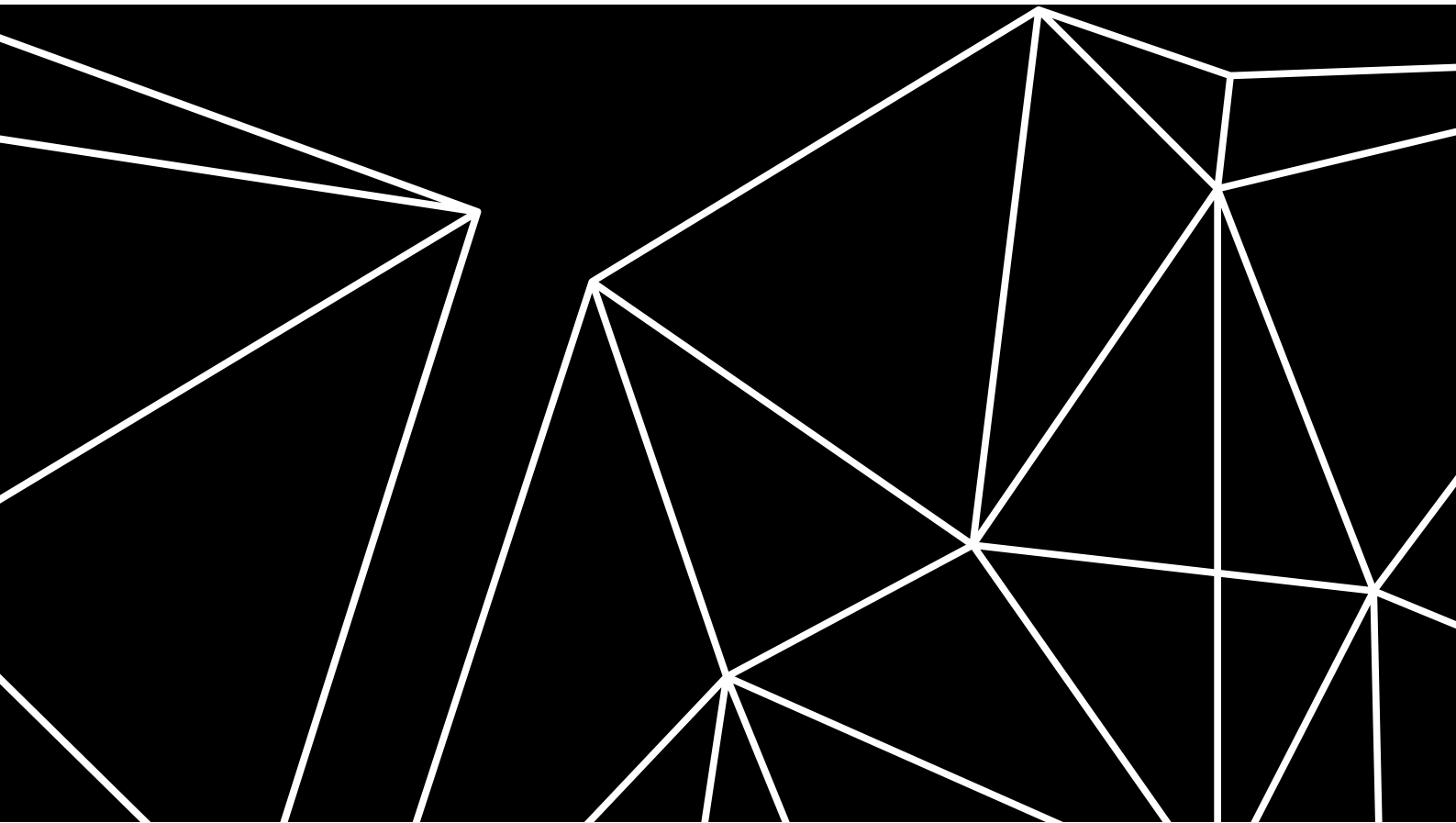


figure 2-4:partie_graphique



GitHub



GIT ET GITHUB

- **Github**



GitHub est une **interface Web** qui utilise **Git**, le logiciel de contrôle de version open source qui permet à plusieurs personnes d'apporter des modifications distinctes aux pages Web en même temps. Comme le note Carpenter, parce qu'il permet une collaboration en temps réel, GitHub encourage les équipes à travailler ensemble pour créer et modifier le contenu de leur site.

GitHub permet à plusieurs développeurs de travailler sur un seul projet en même temps, réduit le risque de travail en double ou conflictuel et peut aider à réduire le temps de production

GIT ET GITHUB

- Charger les dossiers du projet sur github
- **Première méthode** : faites glisser des fichiers

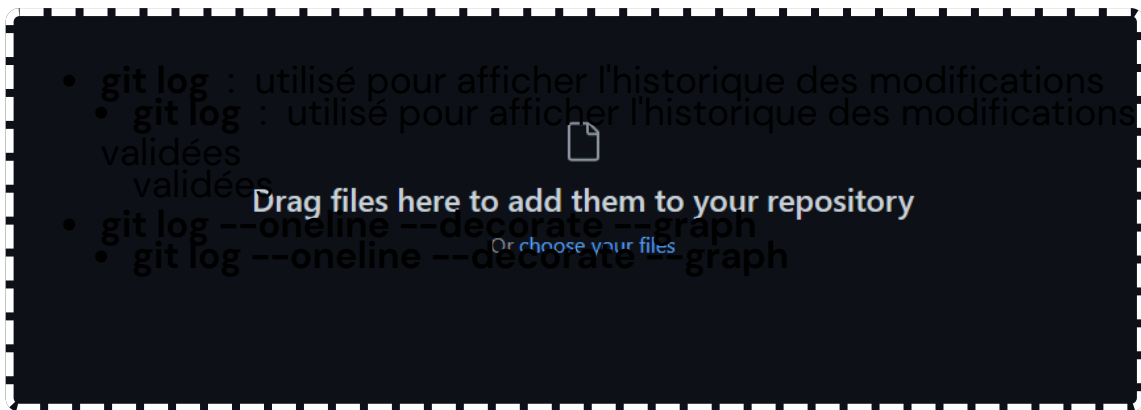


figure 3-1:github upload files

- **Deuxième méthode:** en ligne de commande
 - 1-ouvrir votre fichier avec Gitbah
 - 2-exécute ces commandes
 - **git init**
 - **git status**
 - **git commit -m "name_of_commit"**
 - **git remote add origin < HTTPS_Clone_de_votre_repositories>**
 - **git push -u origin master**

GIT ET GITHUB

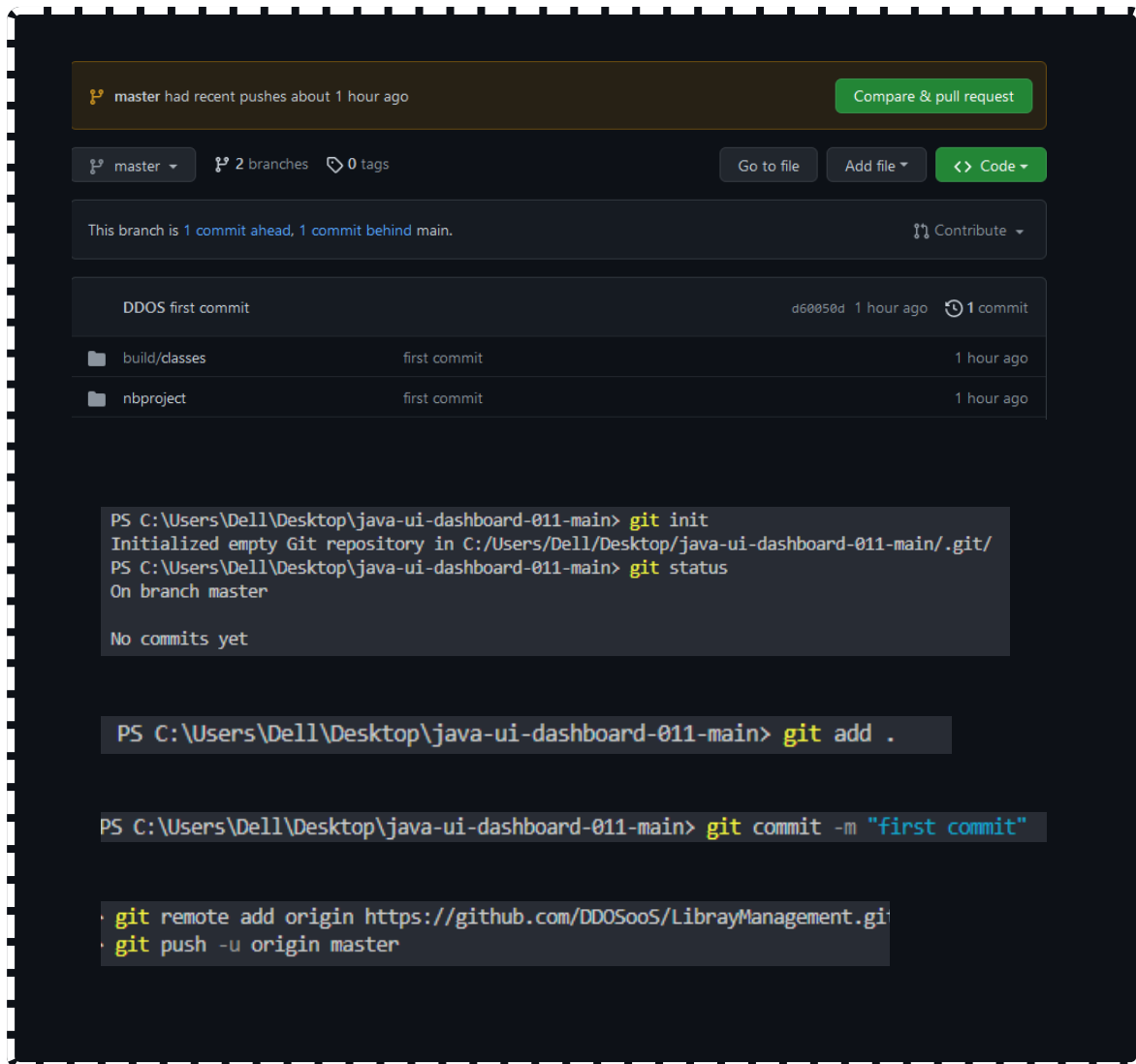
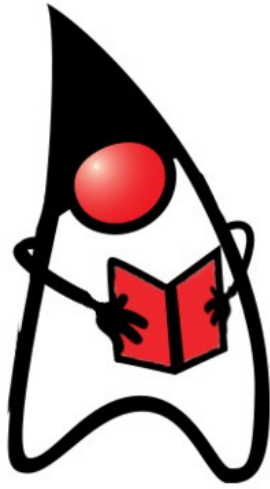
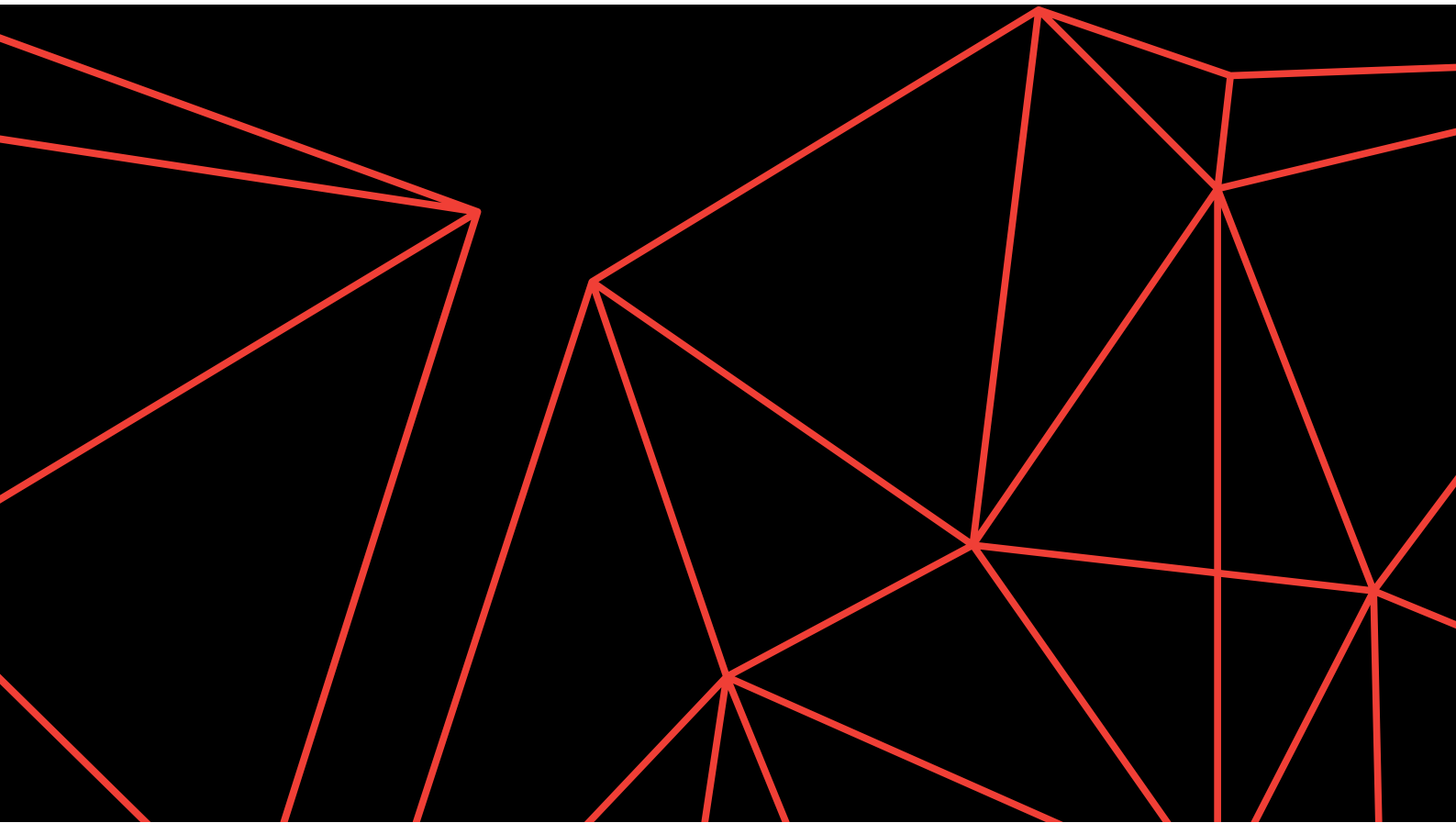


figure 3-2:github repositories and commande line

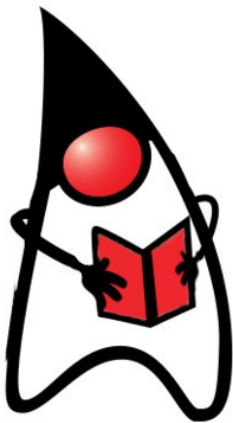


Java DOC



JAVA DOC

- **java DOC**



JavaDoc est un outil utilisé pour générer de la documentation pour les programmes Java. Il est généralement utilisé par les développeurs pour générer de la documentation pour leur code, ainsi que par les utilisateurs qui veulent comprendre comment fonctionne un programme particulier.

La documentation fournit des informations détaillées sur chaque élément, y compris sa fonction, ses paramètres d'entrée, ses valeurs de retour et toute autre information pertinente.

COMMENTAIRE

- Importance des commentaires

Lors de la programmation, il est fortement recommandé d'ajouter un bref commentaire à toutes les procédures décrivant les caractéristiques fonctionnelles de la procédure (ce qu'elle fait).

```
/**
 * Utilitaires divers.
 */
public class Utils {

    /**
     * Afficher un message à la console
     * @param message le message à afficher
     */
    public static void afficher(String message) {
        // @start region="exemple" :
        System.out.println(message);
        // @end
    }
}
```

figure 4-1: exemple de commentaire

- Ceci est utile à la fois pour l'auteur du code et pour toute autre personne susceptible de rencontrer le code. Vous devez séparer les informations d'implémentation (comment le processus fait ce qu'il doit faire) des commentaires qui décrivent les fonctions fonctionnelles. Si vous fournissez des informations sur les performances dans la description, assurez-vous de les mettre à jour lorsque vous mettez à jour la fonctionnalité.
- **Commenter est une bonne pratique que tout les programmeur devrait suivre, que vous soyez un expert ou un débutant Aujourd'hui,**

COMMENTAIRE

- Les différents types de commentaires
 - **Les commentaires sur une seule ligne**

```
//la variable counter est pour stocker le nombre d'utilisateur  
static int counter =0 ;
```

figure 4-2: exemple de commentaire linéaire

- **Le commentaire multilignes**

vous pouvez commenter un texte écrit sur plusieurs lignes. Les développeurs l'utilisent très souvent. Ces commentaires sont définis par les symboles /* et */. L'exemple ci-dessous illustre leur utilisation

```
/*  
 * la variable counter est pour stocker le nombre d'utilisateur  
 * counter est un variable de la class  
 */  
  
static int counter =0 ;
```

figure 4-3: exemple de commentaire multilignes

COMMENTAIRE

- Generer les commentaires avec Java Doc

L'outil Javadoc utilise plusieurs types de fichiers sources pour générer la documentation :

- Les fichiers sources .java
- Les fichiers de commentaires d'ensemble
- Les fichiers de commentaires des packages
- D'autres fichiers tels que des images, des fichiers HTML,

- Les tags définis par javadoc

Tag	Rôle
@author	vous pouvez spécifier le ou les auteurs de l'élément.
@param	permet la documentation des paramètres des éléments
@return	vous pouvez donner une description de la valeur de retour de la méthode qui l'a : il n'est donc pas nécessaire de l'utiliser pour une méthode qui retourne vide.
@exception	vous pouvez spécifier une exception qu'un élément peut lever
@throws	identique à @exception
@see	Le tag génère un lien vers une entité ayant un lien avec celle documentée.
@version	vous pouvez spécifier un numéro de version. Cet identifiant ne doit être utilisé que pour un élément d'une classe ou d'une interface.
@since	peut spécifier le numéro de version de la classe ou de l'interface à partir de laquelle l'élément décrit est disponible

COMMENTAIRE

- Exemples

```
/**
 * constructeur pour le constructeur la classe person
 * @param age :c'est un entier (age>=0)
 * @param name:c'est le nom de la personne
 * @param Cin :identifiant de la personne
 * @throws AgeException si (age<0)
 */
int age;
String name;
int cin;
public Person(int age, String name,int cin)throws ageException {
    this.age = age;
    this.name = name;
    this.cin=cin;
}
```

figure 4-4: exemple de commentaire avec javaDoc

```
@Override
/**
 * la creation du reporesentation pour l'objet
 * @return une chaine qui contient les informartion du l'objet persoone
 */
public String toString() {
    return "Person [age=" + age + ", name=" + name + ", cin=" + cin + "]";
}
```

figure 4-5: exemple de commentaire avec javaDoc2

DOCUMENTATION

- Generer la documentation sous forme html

- 1.cliquer droit sur votre projet
- 2.choisir export
- 3.sélectionner l'option du javadoc
- 4.sélectionner votre projet
- 5.cliquer sur le betton finish
- 6.cliquer sur index.html

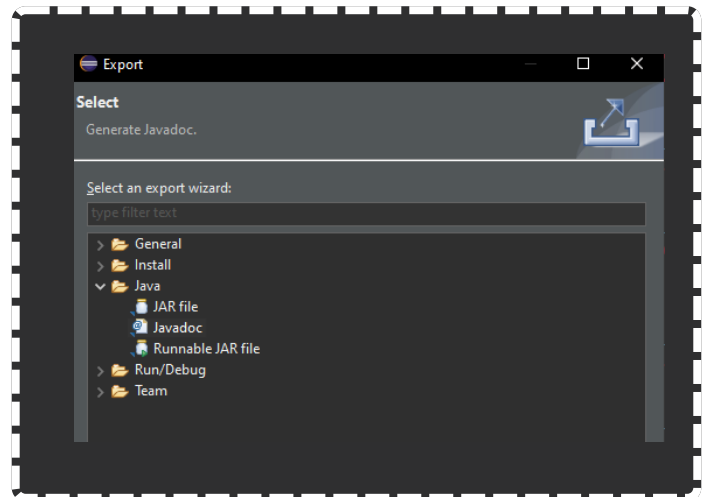


figure 5-1: etap1 documentation

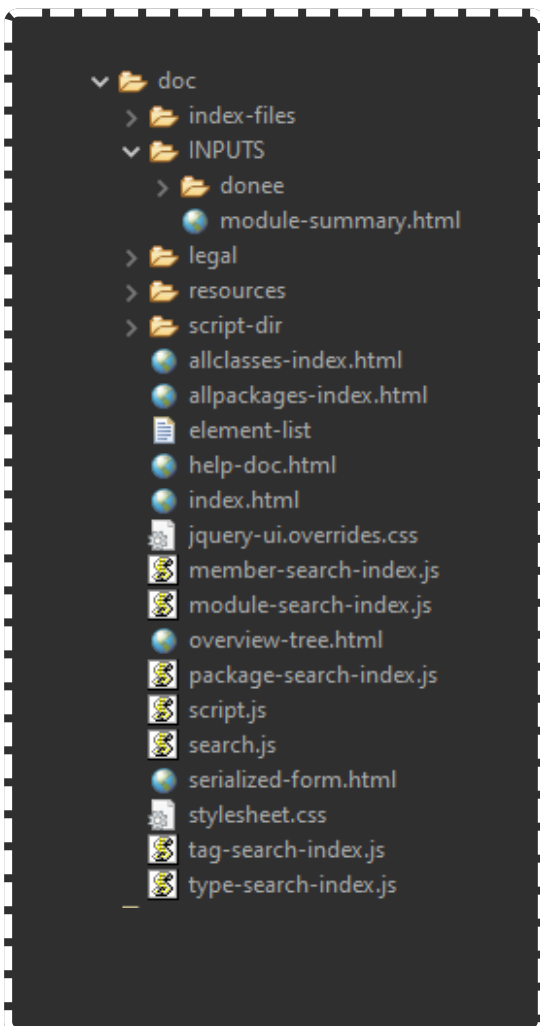


figure 5-3: javadoc file

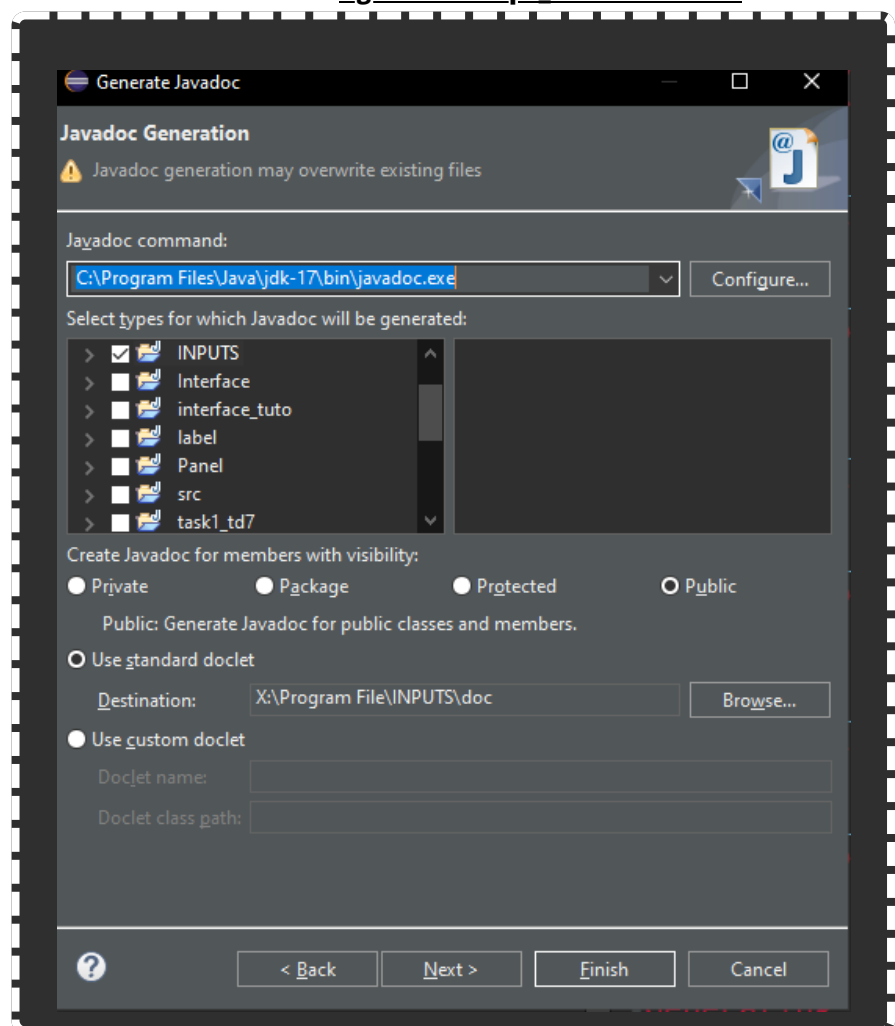


figure 5-2: etap2 documentation

DOCUMENTATION

- Generer la documentation sous forme html

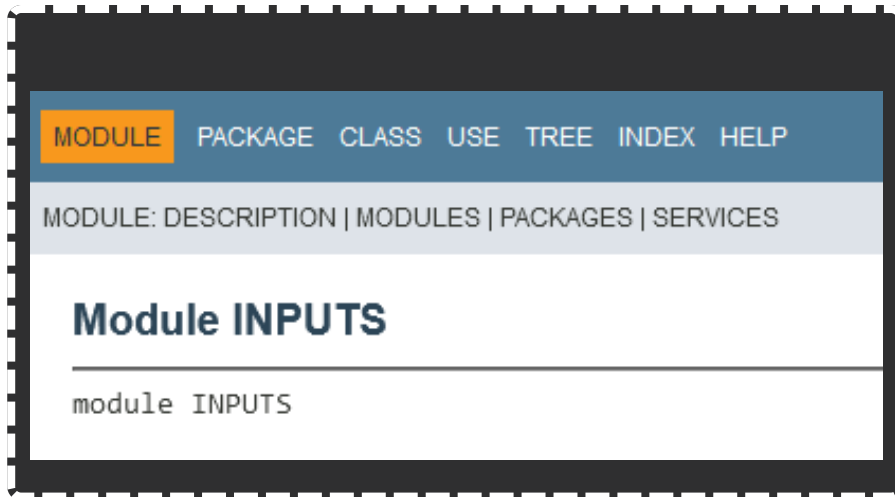
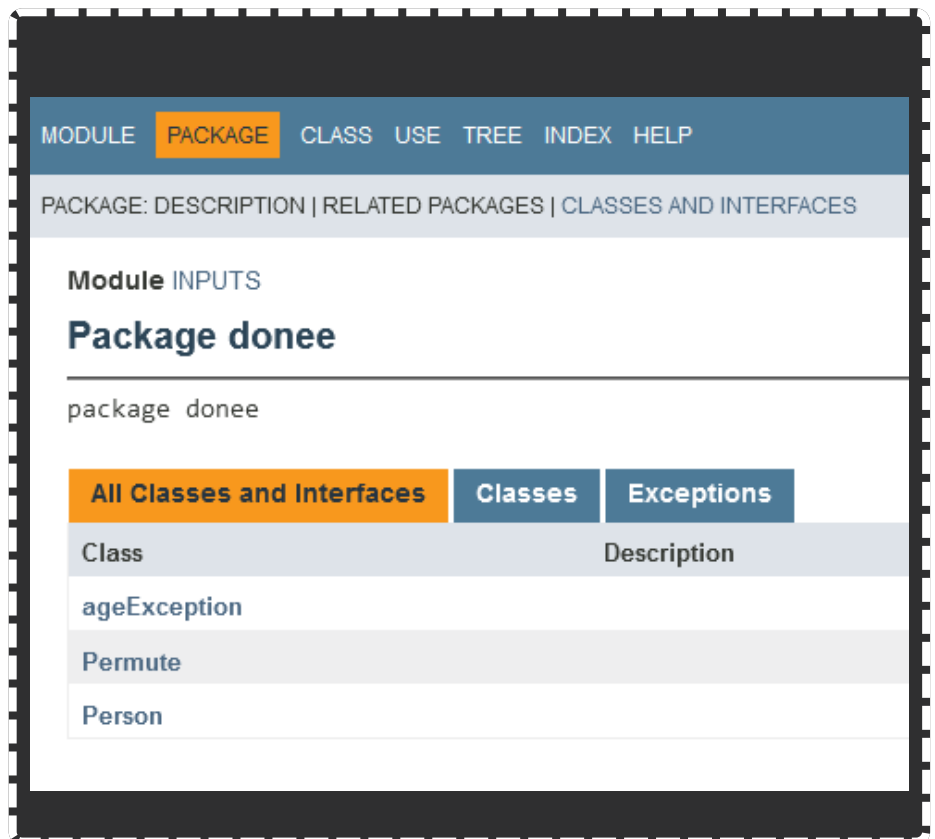


figure 6-1: inde html docummentation1

figure 6-2: index_html_docummentation2



DOCUMENTATION

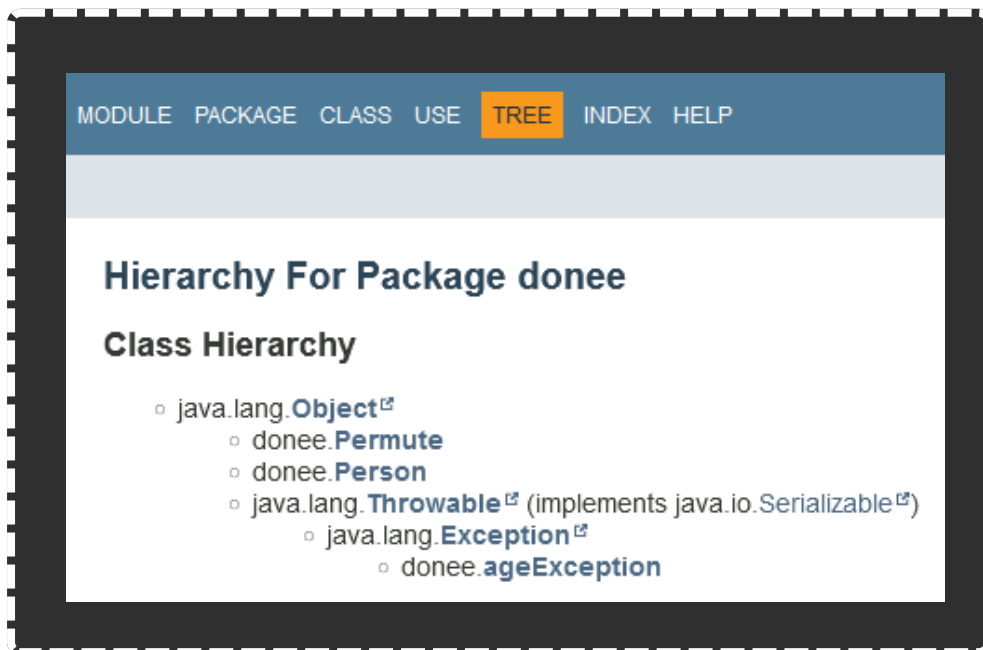


figure 6-3: index_html_documentation3