

# TP1

Formateur : Boris Rose

## Git

Actuellement, normalement, vous avez sur votre bureau un dépôt local git (un dossier initialisé comme dépôt local git) - Vous l'avez obtenu avec la première commande qui suit :

```
# dans le dossier en question
git init
# la commande ci-dessus initialise un référentiel local git
# vous associez à ce dépôt local un dépôt distant git@remote_repo_address
git remote add origin git@remote_repo_address
# si vous avez mal écrit le nom du dépôt distant - normalement - "origin"
# vous pouvez supprimer l'association
git remote remove origin
# et la recréer
git remote add origin git@remote_repo_address
# votre repremier push vous le faites sur la branche main ou master
# faites un git status pour voir le nom correct de la branche
# pour faire un commit vous devez ajouter qqch au dossier
# par exemple, créons un fichier README.md
echo "# IHM" >> README.md
# Cela crée un fichier README.md et ajoute à son contenu # IHM
git add README.md # stage le fichier ajouté
git commit -m 'initial commit' # première sauvegarde
git push origin master # ou main en fonction de votre branche par défaut
# Une fois le premier push effectué vers votre dépôt distant
# créez votre première branche ( chaque tp doit être sur une branche différente )
git branch feature/tp-1
# Cela crée une branche locale qui s'appelle feature/tp-1
# Pour vous rendre sur cette branche tapez
git checkout feature/tp-1
# si vous faites git status vous verrez que vous êtes
# mtn sur la branche feature/tp-1
# pour aller sur une autre branche faites:
git branch
# cela vous montrera vos choix possibles
# et votre emplacement actuel *
git checkout main # vous ferez revenir par exemple sur la branche main
```

Le workflow qui suit est à reproduire à chaque fois que vous faites des changements significatifs dans votre référentiel git local autrement dit dans votre dossier sur votre bureau

```
git status # afin de voir l'état actuel de votre dossier
git add . # afin de TOUT mettre en "stage" c-t-d prêts à être "commités"(= sauvegardés)
# ensuite vous réalisez le commit ( la sauvegarde locale de votre travail )
```

---

```
# quand je dis local c'est-à-dire qu'elle n'existe QUE sur votre ordinateur pas sur Internet
git commit -m "feat/"
# feat/ si vous ajoutez une feature
# fix/ si vous corrigez un bogue
# ensuite vous envoyez votre travail sur votre dépôt distant ( sur gitlab ou github )
git push origin tp_branch # pas main / master ! c'est une branche sur laquelle vous ne devez pas pousser
#normalement vous travaillez sur une branche différente pour chaque tp
```

Pour avoir plus d'informations je vous ai mis un cours sur git dans le dossier scripts ainsi qu'un cours sur bash pour mieux utiliser le terminal

## TP1

Sur une branche tp-1, répondez dans un fichier avec l'extension **.md** aux questions suivantes :

- Utiliser le cours User Centered Design
- Utiliser les cours du dossier Figma
- Eviter les fautes d'orthographe ( 3 points en moins si plus de 5 fautes )
- Le non-respect des consignes des questions entraîne **3 points en moins** par consigne non respectée.
- En donnant la définition d'un besoin utilisateur, justifiez ceux ou celui auquel vous répondez dans l'interface ou les interfaces que vous êtes en train de concevoir sur Figma.
- En donnant le processus du Design Centré Utilisateur, à quelle étape êtes-vous ? . Si vous n'avez pas réalisé une étape du processus mais que vous êtes en train de réaliser une étape ultérieure, réalisez l'étape en question et ajoutez les documents l'attestant dans votre dépôt git
- Ajoutez à votre projet Figma des observateurs. Quel principe fondamental du User-Centered Design (UCD) appliquez-vous ?
- En vous aidant du cours sur les calques, donnez leur définition et les bonnes pratiques d'utilisation de ces calques. Si en vérifiant votre travail de maquettage je me rends compte que les bonnes pratiques ne sont pas respectées ce sera **moins 3 points**
- En vous aidant du cours sur les composants, donnez la définition d'un composant (réutilisable).

Pour aller plus loin :

- En vous aidant du cours sur les guides, styles et grilles, rendez-vous jusqu'à la section bouton primaire. Créer un bouton primaire dans votre design system, un bouton secondaire et un bouton tertiaire. En utilisant la notion de variantes vue dans le cours composants variantes et bibliothèques, définir des étapes afin de garantir un feedback pour chaque état.
- En vous aidant du dossier docs, définissez le wayfinding et implémentez-le dans vos interfaces
- Pour vous assurer de l'affordance d'un élément tel que définie par Don Norman ( voir cours auteurs ), que pouvez-vous faire quand vous concevez un bouton pour une interface ?