

GIT

Objectifs du cours

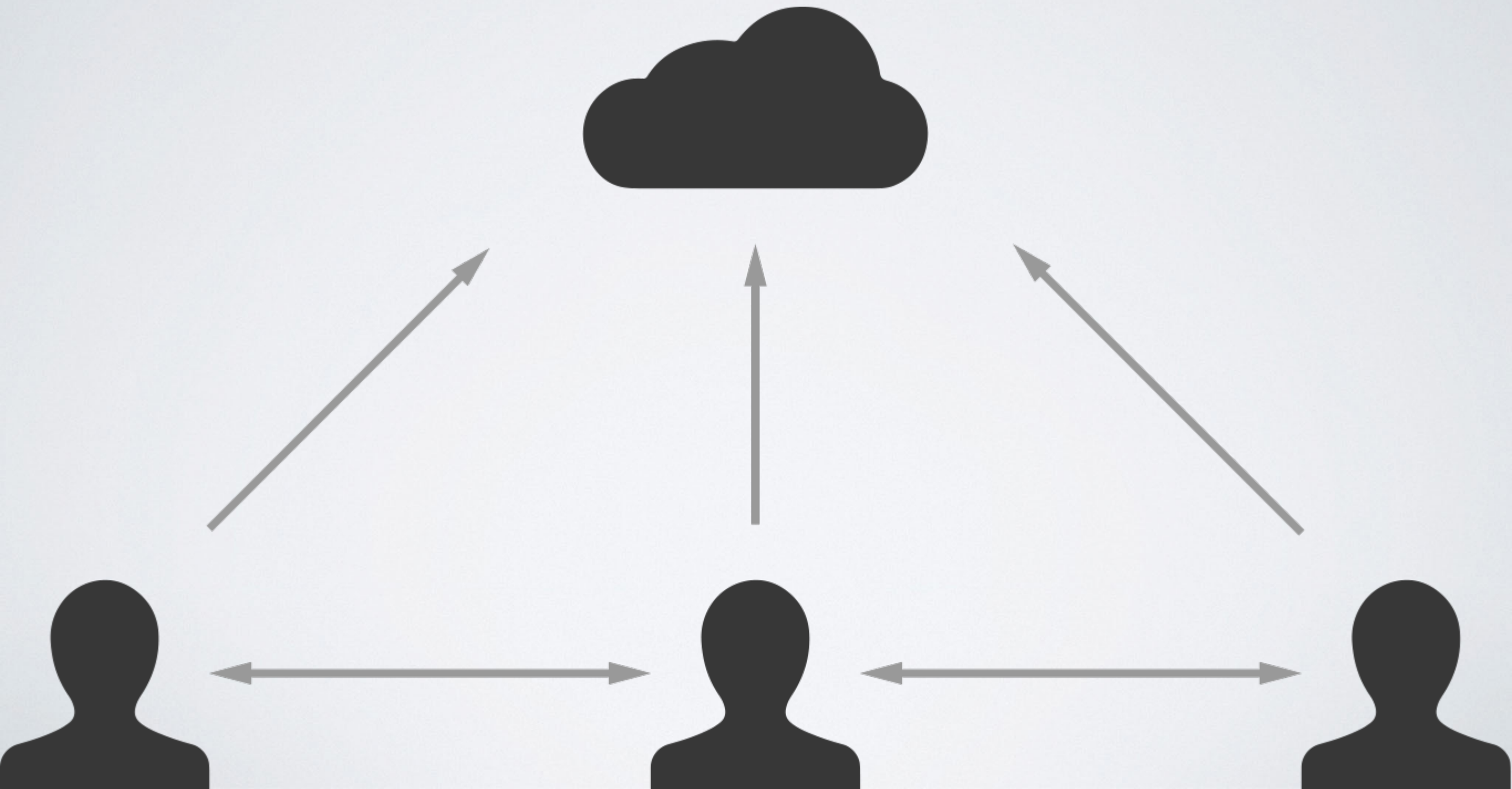
18h (7 février - 11 juin)

- Comprendre le fonctionnement de git
- Connaître les principales commandes
- Apprendre à travailler de manière collaborative

Qu'est ce que GIT ?

- Outil de versionning
- Rapide
- Gestion des branches

A quoi ça sert ?



Principe

Installation de GIT

&

Inscription sur GitHub

Créer un dépôt

Configuration de Git

```
git config --global user.name "John Doe"
```

```
git config --global user.email johndoe@example.com
```


Mise en commun du travail

Mise en commun du travail

Afficher les fichiers modifiés en local :

Git status

```
Changes to be committed:
```

```
modified:   .htaccess
```

```
-
```

```
Changes not staged for commit:
```

```
modified:   README.md
```

```
-
```

```
Untracked files:
```

```
.DS_Store
```

Gitignore

Fichier qui permet d'ignorer des éléments

```
/dossier_a_cacher  
.DS_Store
```


Mise en commun du travail

Afficher les changements :

```
Git diff [nom_fichier]
```

```
--- a/web/robots.txt
+++ b/web/robots.txt
-# www.robotstxt.org/
-
+User-agent: *
```

Mise en commun du travail

Ajouter des fichiers :

```
Git add [chemin/du/fichier]
```

Envoyer son paquet :

```
Git commit -m "[Description]"
```



Mise en commun du travail

Lister les commits :

Git log

```
commit d45062dc97eec6631624bbca384dcce2fc96660e
Author: Yoann Coualan <yoann.coualan@gmail.com>
Date:   Wed Sep 13 15:17:29 2017 +0200
Nettoyage code
```

```
commit 5e4dd4947170bb001900f57c2d5aa33c80586a82
Author: Yoann Coualan <yoann.coualan@gmail.com>
Date:   Wed Sep 13 15:14:18 2017 +0200
Script browsersync
```

```
commit 1648477d1c97c49aba1eb9ed150fd3c626f915a5
Author: Yoann Coualan <yoann.coualan@gmail.com>
Date:   Wed Sep 13 15:12:53 2017 +0200
gitignore package-lock.json
```


Mise en commun du travail

Annuler les commit non push

```
Git reset HEAD
```

Annuler un commit précis :

```
Git reset [SHA]
```

Annuler un commit et réinitialiser ses fichiers :

```
Git reset --hard HEAD
```

Annuler les modifications d'un fichier avant le commit :

```
Git checkout [chemin/du/fichier]
```

Mise en commun du travail

Récupérer les nouveautés :

```
Git pull [remote] [branch]
```

Envoyer ses fichiers :

```
Git push [remote] [branch]
```

Annuler un commit publié :

```
Git revert [SHA]
```



Les branches

Les branches

Créer une branche :

```
Git checkout -b [nom_branche]
```

Afficher les branches existantes :

```
Git branch
```

Se déplacer sur une branche :

```
Git checkout [nom_branche]
```

Supprimer une branche :

```
Git branch -d [nom_branche]
```

Travail collaboratif

(sans risque)

Travail collaboratif

| branche par personne ou par tâche

Garder main constamment propre

| commit par fonctionnalité

Pull / Push

```
Git add [fichier] [fichier]
Git commit -m "[Description]"
-
Git pull origin main
// Résolution conflits
Git push origin [votre_branche]
-
Git checkout main
Git pull origin [votre_branche]
Git push origin main
-
Git checkout [votre_branche]
```

Merge

```
Git add [fichier] [fichier]
Git commit -m "[Description]"
-
Git checkout main
Git merge [votre_branche]
// Résolution de conflits + commits
-
Git checkout [votre_branche]
```

⚠ avec le merge il peut y avoir de la perte

Evaluation

10 Points sur le projet :

Organisation et la propreté des commits

10 points sur le QCM

Projet Devises