

TD 2d

Créer et résoudre un conflit

Objectif : fusionner deux branches qui modifient la même ligne, et résoudre le conflit qui en résulte.

`cours2/2d_conflicts/` $\approx 25'$

TD 2d — la première version

On travaille dans le dépôt créé au TD 2a, `cours2/2a_premier_depot/travail/projet_2`.

- 21** Sur `main_code`, ajoutez le code de `depart/main_regex.py` à la suite de celui de la question 11, puis faites un commit. Ne fusionnez pas ce commit dans `develop`.

```
#Au début du fichier :
from operations import *
import re

#À la fin du fichier, dans la boucle if :
s = input()
search_result = re.search(r"\d+[\+\-\*\\/]\d+", s)
if search_result:
    operation = search_result[0]
    ...
```

TD 2d — la seconde version

22 Repassez sur **develop**, créez une branche **main_code_bis** et placez-vous dedans. Les modifications de la question précédente ne doivent plus être visibles.

23 Ajoutez dans `main.py` le code de `depart/main_operations.py`, puis faites un commit.

```
#Au début du fichier :  
import operations as op  
  
#À la fin du fichier, dans la boucle if :  
    print("Tapez une opération :")  
    operation = input()  
    if "+" in operation:  
        [x,y] = operation.split("+")  
        ...
```

TD 2d — le conflit

24 Retournez sur **develop**. Fusionnez **main_code_bis** dans **develop**, puis **main_code** dans **develop**. Le second merge s'arrête sur un conflit.

25 Réglez les conflits en choisissant le code à garder, puis poursuivez le merge. Git ouvre l'éditeur avec le message de fusion : tapez `:wq` puis `Entrée`.

Étape

le merge s'arrête

dans le fichier

une fois résolu