

# Projet Fonctionnel, rendu 0

Ce répertoire contient un programme de départ à partir duquel vous pouvez programmer votre fouine.

## Compilation et execution

Les commandes ci-dessous sont à exécuter dans le répertoire principal, qui est au-dessus de bin/ et lib/.

pour (re)compiler, lancer

```
dune build
```

pour executer le programme, lancer

```
dune exec bin/fouine.exe
```

entrez ensuite une expression arithmetique, avec juste + - et \*, comme par exemple  $4+3*5$

On vous conseille d'exécuter une fois la commande

```
ln -s _build/default/bin/fouine.exe fouine
```

pour ne pas avoir à saisir le \_build/default/main.exe

Après quoi vous pouvez - taper ./fouine pour lancer l'exécutable et saisir une expression au clavier - taper ./fouine test/basic.ml pour lancer fouine sur le fichier basic.ml

## Ce qui est attendu

Lisez les fichiers dans cette archive, compilez le code (cf. ci-dessous)

Puis étendez fouine. Voici une succession d'étapes, traitez celles que vous pouvez, dans la mesure de vos possibilités.

1. expressions arithmétiques. Contemplez l'évaluation des expressions arithmétiques, améliorez les choses du point de vue de la factorisation du code si vous voyez un moyen de le faire ; sinon, laissez en l'état pour le moment.
2. opérateurs booléens et if.. then.. else
3. fonction prInt (cf. les notes de cours)
4. ajout des déclarations let.. in
5. ajout des fonctions
6. fonctions récursives etc., ad lib

Pour chaque étape, référez-vous aux notes de cours (partie 1.2). Écrivez à chaque fois de petits fichiers de tests dans le répertoire test/

## Travailler sur fouine, modifier le code

bin/fouine.ml : fichier principal

lib/expressions.ml : définition des expressions et de l'évaluation

`lib/affiche.ml` : affichage des expressions et des valeurs

`lib/lexer.mll` : lexèmes, analyse lexicale

`lib/parser.mly` : règles de grammaire, analyse syntaxique

`tests/` : sous-répertoire de tests

fichiers `dune` et `dune-project` : pour la compilation, à ne pas modifier a priori

En principe vous n'avez à modifier que les fichiers `expr.ml` `lexer.mll` `parser.mly` dans le répertoire `lib/`

## **Erreurs à la compilation en lien avec le lexer et le parser :**

référez-vous au fichier *\*README-parser.md*, dans cette archive

## **Date limite pour téléverser ce rendu sur la page du portail des études :**

**lundi 2 février 2025 à 23h59**

- . indiquez brièvement au début du fichier `Readme` ce que vous avez pu ajouter à
- . envoyez une archive **\*\*qui compile\*\***