

# Pico-Project : Calculatrice Simple

## Interface utilisateur

- Le programme doit demander à l'utilisateur de saisir deux nombres
- L'utilisateur doit ensuite choisir une opération parmi les opérations proposées :
  - Addition (+)
  - Soustraction (-)
  - Multiplication (\*)
  - Division (/)
  - Division entière (//)
  - Modulo (%)
  - Puissance (\*\*)
  - Taper 'q' pour quitter le programme

## Fonctionnalités

- Le programme tourne en boucle jusqu'à ce que l'utilisateur tape 'q' comme opération
- Le programme doit afficher le résultat de l'opération choisie
- Si l'utilisateur choisit une opération non valide, un message d'erreur doit être affiché
- Gérer les cas où l'utilisateur tente de diviser par zéro
- Pour la division entière et le modulo, gérer également la division par zéro

## Tests manuels

### Tests des opérations

#### 1. Addition

- Entrer 5 et 3, vérifier que le résultat est 8

```
Donner X: 5
Donner Y: 3
Donner Operation (+, -, *, /, //, %, **): +
5.0 + 3.0 = 8.0
```

- Entrer 3.5 et 2.7, vérifier que le résultat est 6.2

```
Donner X: 3.5
Donner Y: 2.7
Donner Operation (+, -, *, /, //, %, **): +
3.5 + 2.7 = 6.2
```

- Entrer -5 et 3, vérifier que le résultat est -2

```
Donner X: -5
Donner Y: 3
Donner Operation (+, -, *, /, //, %, **): +
-5.0 + 3.0 = -2.0
```

## 2. Soustraction

- Entrer 10 et 4, vérifier que le résultat est 6

```
Donner X: 10
Donner Y: 4
Donner Operation (+, -, *, /, //, %, **): -
10.0 - 4.0 = 6.0
```

- Entrer 5.5 et 2.2, vérifier que le résultat est 3.3

```
Donner X: 5.5
Donner Y: 2.2
Donner Operation (+, -, *, /, //, %, **): -
5.5 - 2.2 = 3.3
```

- Entrer -10 et -4, vérifier que le résultat est -6

```
Donner X: -10
Donner Y: -4
Donner Operation (+, -, *, /, //, %, **): -
-10.0 - -4.0 = -6.0
```

## 3. Multiplication

- Entrer 6 et 7, vérifier que le résultat est 42

```
Donner X: 6
Donner Y: 7
Donner Operation (+, -, *, /, //, %, **): *
6.0 * 7.0 = 42.0
```

- Entrer 2.5 et 3.0, vérifier que le résultat est 7.5

```
Donner X: 2.5
Donner Y: 3.0
Donner Operation (+, -, *, /, //, %, **): *
2.5 * 3.0 = 7.5
```

- Entrer 10.5 et 2.5, vérifier que le résultat est 4.2

```
Donner X: 10.5
Donner Y: 2.5
Donner Operation (+, -, *, /, //, %, **): /
10.5 / 2.5 = 4.2
```

## 4. Division entière

- Entrer 17 et 5, vérifier que le résultat est 3

```
Donner X: 17
Donner Y: 5
Donner Operation (+, -, *, /, //, %, **): //
17 // 5 = 3
```

## 5. Modulo

- Entrer 17 et 5, vérifier que le résultat est 2

```
Donner X: 17
Donner Y: 5
Donner Operation (+, -, *, /, //, %, **): %
17 % 5 = 2
```

## 6. Puissance

- Entrer 2 et 3, vérifier que le résultat est 8

```
Donner X: 2
Donner Y: 3
Donner Operation (+, -, *, /, //, %, **): **
2.0 ** 3.0 = 8.0
```

## 7. Quitter le programme

- Taper '/q' comme opération, vérifier que le programme se termine

```
Donner X: /q
Program terminated
```

## Tests des cas d'erreur

1. Division par zéro : Entrer 10 et 0, vérifier qu'un message d'erreur approprié s'affiche

```
Donner X: 10
Donner Y: 0
Donner Operation (+, -, *, /, //, %, **): /
Erreur : Division par zéro
```

2. Division entière par zéro : Entrer 10 et 0, vérifier qu'un message d'erreur approprié s'affiche

```
Donner X: 10
Donner Y: 0
Donner Operation (+, -, *, /, //, %, **): //
Erreur : Division par zéro
```

3. Modulo par zéro : Entrer 10 et 0, vérifier qu'un message d'erreur approprié s'affiche

```
Donner X: 10
Donner Y: 0
Donner Operation (+, -, *, /, //, %, **): %
Erreur : Division par zéro
```

4. Opération invalide : Entrer un opérateur non listé, vérifier qu'un message d'erreur s'affiche

```
Donner X: 10
Donner Y: 0
Donner Operation (+, -, *, /, //, %, **): a
Erreur : Opérateur invalide
```