

Labo d'introduction à l'informatique

pour les mathématiques

Yann Thorimbert



**UNIVERSITÉ
DE GENÈVE**

CENTRE UNIVERSITAIRE
D'INFORMATIQUE

Semaine 4

Les erreurs les plus courantes



Avant toute chose

- Identifiants secrets :
 - À demander aujourd'hui auprès des assistants.
 - Seront utilisés pour communiquer les bonus de façon anonyme sur Moodle.
- Exercices papier disponibles.
- Le test est ouvert dès à présent.



La session d'aujourd'hui

- Consolidation des acquis précédents.
- Test Moodle court.
- Retour sur les difficultés et erreurs les plus fréquentes.
- Temps pour effectuer les exercices papier sur le codage des nombres.



Difficulté n° 1 | Déclaration vs utilisation d'une fonction

```
function val = cube(x)
    val = x * x * x;
end
```

cube.m

Appel de fonction.
Fait référence à du
code écrit dans un
autre fichier.

```
a = 3;
resultat = cube(a);
```

monscript.m



Difficulté n° 1 | Déclaration vs utilisation d'une fonction

```
function val = cube(x)
    val = x * x * x;
end
```

cube.m

Comment Matlab peut
trouver le code qui définit la
fonction ?

```
a = 3;
resultat = cube(a);
```

monscript.m



Difficulté n° 1 | Déclaration vs utilisation d'une fonction

```
function val = cube(x)
    val = x * x * x;
end
```

cube.m

Comment Matlab peut trouver
le code qui définit la fonction ?
Réponse : grâce au **nom de
fichier !**

```
a = 3;
resultat = cube(a);
```

monscript.m



Difficulté n° 1 | **Déclaration vs utilisation d'une fonction**

Nom de la fonction



```
function val = cube(x)
    val = x * x * x;
end
```



Difficulté n° 1 | Déclaration vs utilisation d'une fonction

```
function val = cube(x)
    val = x * x * x;
end
```

Signale la fin du code de la fonction



Difficulté n° 1 | Déclaration vs utilisation d'une fonction

Variable contenant la **valeur de retour**

Assignment
d'une valeur à
la variable de
retour

```
function val = cube(x)
    val = x * x * x;
end
```



Difficulté n° 1 | **Déclaration vs utilisation d'une fonction**

Pour le moment, si une fonction ne retourne pas, c'est que son rôle est d'afficher quelque chose !

```
function cube(x)
    val = x * x * x;
    disp(val);
end
```



Difficulté n° 1 | Déclaration vs utilisation d'une fonction

```
function val = cube(x)
    val = x * x * x;
end
```

Argument de la fonction

Au moment où l'on code, on ne connaît pas la valeur de l'argument !

Cette généricité est désirable. On ne veut pas spécifier ici la valeur de x.

“Utiliser” une fonction : **effectuer un appel** à la fonction.



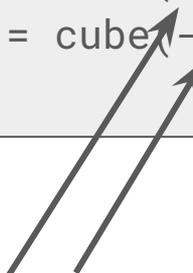
Difficulté n° 1 | Déclaration vs utilisation d'une fonction

Définition de la fonction

```
function val = cube(x)
    val = x * x * x;
end
```

Appel de la fonction

```
a = cube(3) %a = 27
b = cube(-2) %c = -8
```



La valeur de l'argument
est connue au moment de
l'appel



Difficulté n° 2 | Problèmes de syntaxe

- Exemple : oublier d'écrire le mot-clé `end` à la fin d'un bloc conditionnel.
- Les messages d'erreurs ne sont pas toujours explicites.
- Si vous ne comprenez pas le message, c'est à vous d'enquêter sur l'erreur en commençant par chercher les problèmes de syntaxe, avant de modifier le code plus en profondeur.



Difficulté n° 2 | Compréhension des erreurs

- Exemple :

```
function res=cube(x)
    valeur = x^3;
end
```

(Ici, le problème est qu'aucune valeur n'est assignée à la variable de retour)

- Quand on appelle cette fonction (par exemple : `a = cube(3)`), on obtient avec Octave l'erreur : `error: value on right hand side of assignment is undefined.`
- avec Matlab : Output argument "res" (and possibly others) not assigned a value in the execution with "cube" function.



Difficulté n° 2 | Compréhension des erreurs

- Utilisez la documentation, les forums ou les outils d'IA si vous ne comprenez pas l'erreur.
- Attention : Matlab et Octave peuvent différer.
- Si votre code semble marcher dans Matlab mais pas avec Coderunner :
 - 1) Problème de compréhension du format des arguments ?
 - 2) Problème de compréhension du format d'output ?
 - 3) Différences Matlab-Octave ? Exemple : Matlab est permissif sur l'oubli du "end".



Difficulté n° 3 | Problèmes de formatage de l'output

- Utilisation de `disp()` : à préférer lorsqu'il n'y a **qu'une valeur à afficher**. Permet de ne pas avoir à spécifier le format.
- Utilisation de `fprintf()` : à préférer lorsqu'il y a **plusieurs valeurs à afficher** où lorsque vous devez contrôler précisément le format. Se référer aux diapositives ou à la documentation, car la syntaxe est stricte !
- Mettre un **point-virgule** si vous ne voulez pas l'output automatique d'une assignation.