

Labo d'introduction à l'informatique

pour les mathématiques

Yann Thorimbert



**UNIVERSITÉ
DE GENÈVE**

CENTRE UNIVERSITAIRE
D'INFORMATIQUE

Semaine 7

Chaînes de caractères

Tableaux de cellules

Lecture de fichiers



Boucles | **Approfondissement : mot-clé continue**

Comment forcer le passage à l'itération suivante, sans exécuter le reste du code ?

```
for i=1:10
    if mod(i,2) == 0
        continue; % itération suivante et ignore le reste
    end
    disp(i);
end
```

Ce mot-clé existe dans la plupart des langages, tout comme break.



Boucles | Approfondissement : mot-clé continue

Deux codes équivalents :

```
% code 1
for i=1:10
    if mod(i,2) == 0
        continue; % itération suivante et ignore le reste
    end
    disp(i);
end
```

```
% code 2
for i=1:10
    if mod(i,2) ~= 0
        disp(i);
    end
end
```

Boucles | Approfondissement : mot-clé continue

Ne pas confondre avec break !

```
% continue
for i=1:10 .....
    if mod(i,2) == 0
        continue;
    end
    disp(i);
end
```



```
% break
for i=1:10
    if mod(i,2) == 0
        break;
    end
    disp(i);
end
.....
```





Les chaînes de caractères (sujet du jour)

“ABCDE”



Chaînes de caractères

- "Chaîne" → *string*.
- Réutilise le concept de **tableau** pour représenter des suites ordonnées de caractères, au lieu de suite ordonnées de nombres.
- Rappel : ASCII établit une **correspondance** nombre ↔ caractères.
Un tableau de nombres peut donc être interprété comme un tableau de caractères.



Correspondance avec code ASCII

```
n = double('A');  
disp(n); % affiche 65
```

```
c = char(65);  
disp(c); % affiche 'A'
```




Déclaration d'une chaîne en Octave/Matlab

Les guillemets indiquent à Matlab que c'est un **vecteur de caractères** (tableau 1D contenant des caractères) :

```
texte = 'Salut !';
```

```
% plus pratique que texte = [83 97 108 117 116 32 33]
```

Déclaration d'une chaîne en Octave/Matlab

- `length( )` s'utilise comme pour les tableaux de nombres.
- TOUS les caractères sont comptés (espaces, ponctuation, ...).

```
texte = 'Salut !';  
n = length(texte);  
disp(n); % affiche 7
```

Différences Matlab/Octave

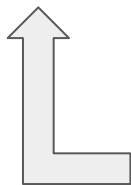
- En Matlab, on différencie tableaux de strings (*double quote* comme dans "salut") et vecteurs de caractères (*single quote* comme dans 'Salut !').
- Avec un vecteur de caractères, on peut accéder facilement à chaque élément.
- En Octave, il n'y a pas cette différence.
⇒ **Dans ce cours, on utilise des vecteurs de caractères** pour représenter du texte.

Exemple de parcours d'un texte

Affichage des lettres une par une :

```
texte = 'Salut !';
```

```
for i=1:length(texte)  
    disp(texte(i))  
end
```



Accès aux éléments comme pour un tableau !



Exemple de détection de caractère

Affichage des lettres une par une :

```
texte = 'Salut !';  
  
for i=1:length(texte)  
    if texte(i) == 'u'  
        fprintf("Il y a un 'u' dans le texte en position %d\n", i)  
    end  
end
```

Exemple de modification

Affichage des lettres une par une :

```
texte = 'Bonjour tout le monde.';

for i=1:length(texte)
    if texte(i) == 'o'
        texte(i) = 'a';
    end
end

disp(texte) % Affiche 'Banjaour taut le mande.'
```

Aparté sur l'affichage des guillemets

- Quand des guillemets (simples ou doubles) font partie d'une chaîne délimitée par le même symbole, on les répète deux fois.
- Exemples :

```
disp("Ceci affiche le mot ""pomme"" simplement");  
disp('Ceci affiche le mot ''pomme'' simplement');  
disp('Ceci affiche le mot "pomme" simplement');
```

Et si l'on veut traiter d'un tableau de textes ?

- Exemple : on veut traiter une liste de mots du dictionnaire.
- Les tableaux en Matlab ne peuvent contenir que des nombres.
- Les string arrays peuvent contenir des éléments qui sont des textes atomiques (non traités ici, car pas disponible en Octave).
- Solution pour un “tableau” contenant autre chose que des nombres :
Les **tableaux de cellules**.

Tableau de cellules

- Les tableaux de cellules sont des structures pouvant comporter des éléments de types variés.

```
a = {1, 'blabla', 3, 'truc', 5.5}
```

- L'accès aux éléments se fait avec les **accolades** :

```
disp(a{2}) % affiche 'blabla'
```

- Fonction `length( )` définie comme pour les tableaux.



Tableau de cellules | **Exemple**

```
mots = {'Pomme', 'Banane', 'Truc'};
for i=1:length(mots)
    mot = mots{i};
    fprintf("Le mot \"%s\" contient %d caractères.\n", mot, length(mot));
end
```

Résultat :

Le mot "Pomme" contient 5 caractères.
Le mot "Banane" contient 6 caractères.
Le mot "Truc" contient 4 caractères.



Lecture d'un fichier texte

```
f = fopen('liste_mots.txt', 'r'); % Ouverture en mode lecture
data = textscan(f, '%s'); % lit les données en tant que strings
fclose(f);
mots = data{1}; % data est un tableau cellulaire de tableaux cellulaires !
% À partir d'ici, on peut utiliser le tableau cellulaire nommé mots.
fprintf("Il y a %d mots dans le fichier.\n", length(mots))
```



Lecture d'un fichier texte : version plus élaborée

```
filename = 'liste_mots.txt';  
fid = fopen(filename, 'r'); % Ouverture en mode lecture  
if fid == -1  
    error('Le fichier n''a pas pu être ouvert.');
```

end

```
data = textscan(fid, '%s'); % '%s' lit les données en tant que strings  
fclose(fid);  
mots = data{1}; % data est un tableau cellulaire de tableaux cellulaires !  
% À partir d'ici, on peut utiliser le tableau cellulaire nommé mots.  
fprintf("Il y a %d mots dans le fichier.\n", length(mots))
```