POLYTECHNIQUE MONTRÉAL

LE GÉNIE EN PREMIÈRE CLASSE

INF1005A: Programmation procedurale

Chapitre 6: Les fichiers fonctions

Agenda

- Définition, introduction
- Structure d'une fonction
- L'appel d'une fonction
- `nargin() / nargout()`
- Fonctions avec un nombre indéfini de paramètres / valeurs de retour
- Autres fonctions, particularités appel d'une fonction en MATLAB

Les fonctions – définition

- Réduction de la complexité des programmes.
- Ensemble d'opérations servant à exécuter une tâche particulière utilisée fréquemment dans la résolution d'un problème.
- Chaque fonction peut être testée et mise au point indépendamment des autres.
- Un programme peut alors se résumer à une séquence d'appels à des fonctions.

Les fichiers fonction – introduction

- ~~Fichier script~~ (fichier M) contenant une fonction.
- Doit répondre à toutes les exigences d'un fichier script (indentations, commentaires, en-tête de fichier, etc...)
- Par convention, le nom de la fonction doit être le même que celui du fichier.
- L'appel de la fonction se fait par le nom du fichier.
- Si le nom est différent, la recherche d'une fonction sera complexifiée, car `lookfor` et `help` sont basés sur le nom de la fonction, et non pas le nom du fichier (voir `lookfor` et `help` au chapitre 2).

Les fichiers fonction – introduction

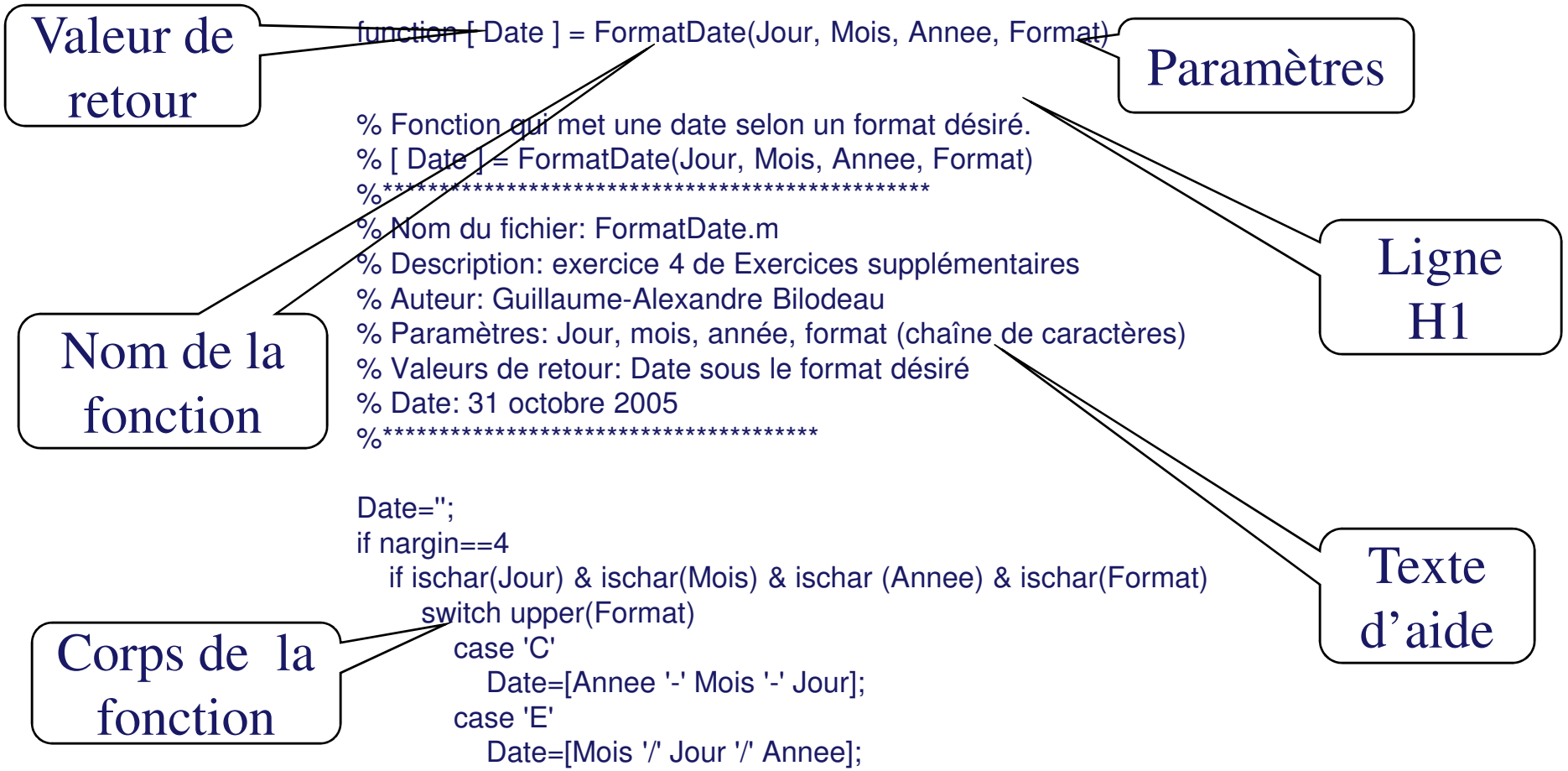
- Les variables qui sont utilisées dans le corps des fonctions, pour les paramètres et pour les valeurs de retour sont locales.

- C'est donc dire que même si elles ont le même nom que des variables d'une autre fonction ou d'un fichier script, elles correspondent à des entités complètement différentes, occupant des espaces mémoires différentes **(Très, très important)**.

Agenda

- Définition, introduction
- Structure d'une fonction
- L'appel d'une fonction
- `nargin() / nargout()`
- Fonctions avec un nombre indéfini de paramètres / valeurs de retour
- Autres fonctions, particularités appel d'une fonction en MATLAB

Structure d'une fonction – présentation




```
function [ Date ] = FormatDate(Jour, Mois, Annee, Format)

% Fonction qui met une date selon un format désiré.
% [ Date ] = FormatDate(Jour, Mois, Annee, Format)
%*********************************************
% Nom du fichier: FormatDate.m
% Description: exercice 4 de Exercices supplémentaires
% Auteur: Guillaume-Alexandre Bilodeau
% Paramètres: Jour, mois, année, format (chaîne de caractères)
% Valeurs de retour: Date sous le format désiré
% Date: 31 octobre 2005
%*************************************

Date='';
if nargin==4
    if ischar(Jour) & ischar(Mois) & ischar (Annee) & ischar(Format)
        switch upper(Format)
            case 'C'
                Date=[Annee '-' Mois '-' Jour];
            case 'E'
                Date=[Mois '/' Jour '/' Annee];
```

Structure d'une fonction – ligne de définition / prototype / en-tête

- La ligne de définition indique à MATLAB que le fichier contient une fonction.
- Elle est construite de la façon suivante :

  `function [`*`valeurs_retour`*`] = nom_fonction(`*`paramètres`*`)`

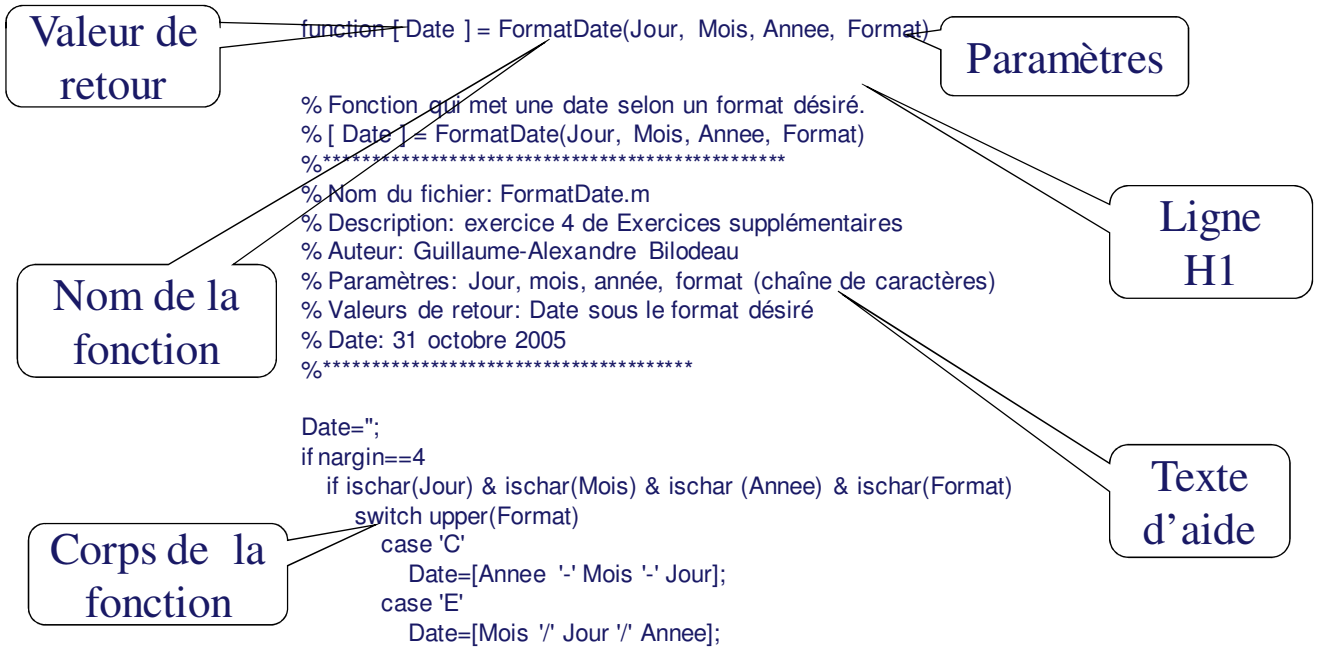
- Les paramètres et les valeurs de retour ne sont pas obligatoires.
- Le mot-clé `function` et le nom de la fonction sont cependant obligatoire.

Structure d'une fonction – ligne de définition / prototype / en-tête

- Nom de la fonction:
  - Élément obligatoire de la ligne de définition de la fonction.
  - Par convention, toujours identique au nom de fichier.
  - C'est le nom de fichier qui est utilisé pour l'appel de la fonction.
- Ordre de recherche des noms de MATLAB :
  1. Vérifie si le nom est le nom d'une variable.
  2. Vérifie si le nom est le nom d'une fonction intégrée à MATLAB.
  3. Vérifie si le nom est le nom d'une fonction utilisateur.

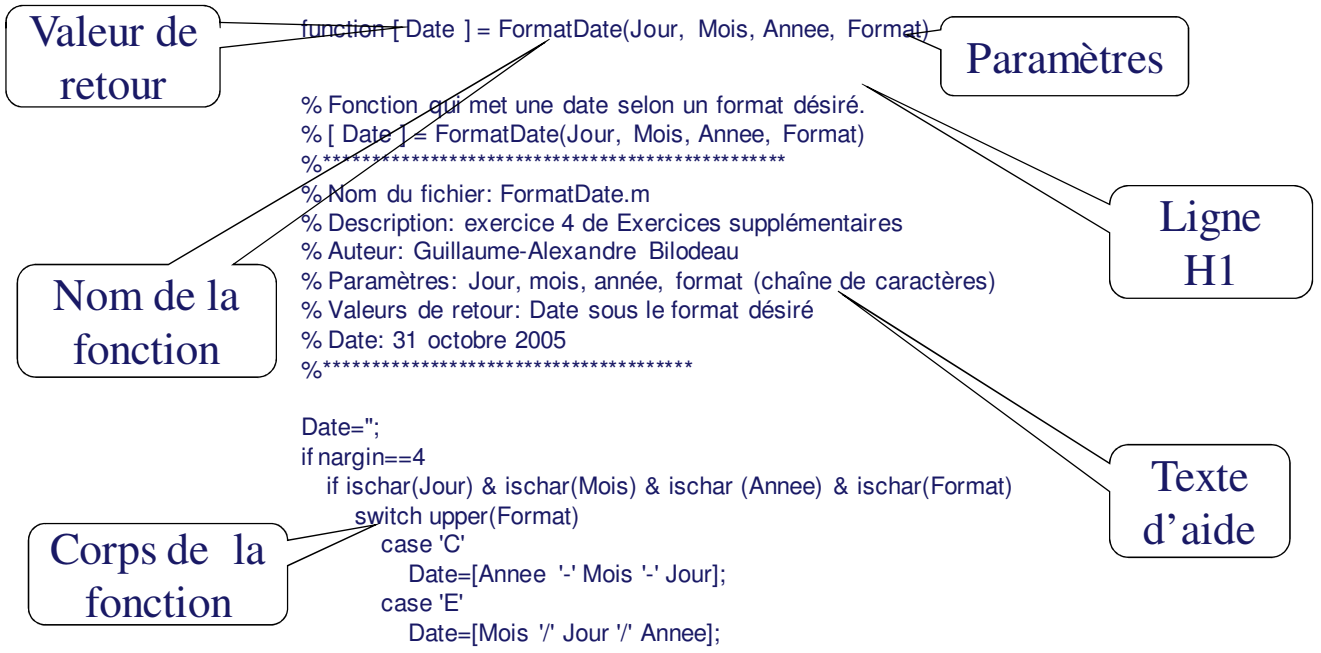
Structure d'une fonction – paramètres

- Les paramètres servent à recevoir les valeurs d'entrée de la fonction. (Lien de communication unique pour obtenir les valeurs d'entrée). **(Très, très important.)**
- Les paramètres sont donnés entre parenthèses et séparés par des virgules.



Structure d'une fonction – valeurs de retour

- Les valeurs de retour sont les valeurs de sortie de la fonction. Elles servent à transmettre le résultat de la fonction au point d'appel.
- Une fonction n'a pas nécessairement de valeur de retour.



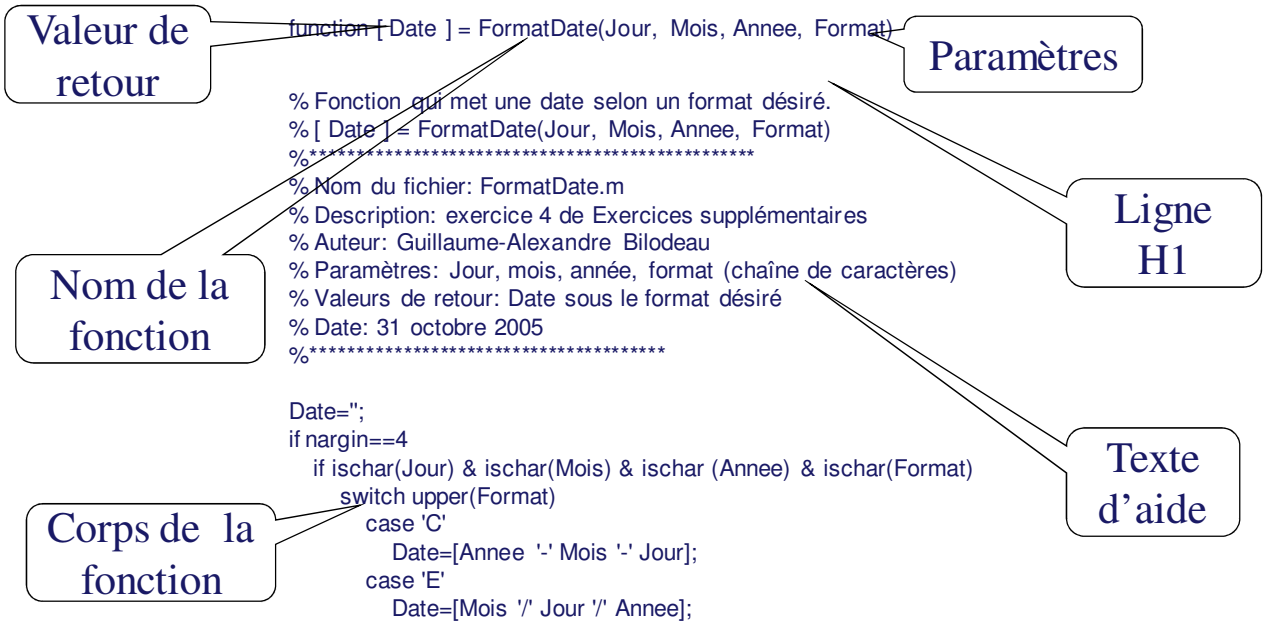
Structure d'une fonction – valeurs de retour

- Il est possible d'en définir plusieurs en les mettant séparées par des virgules.

```
function [a,b,c] = toto(d,e,f)
```

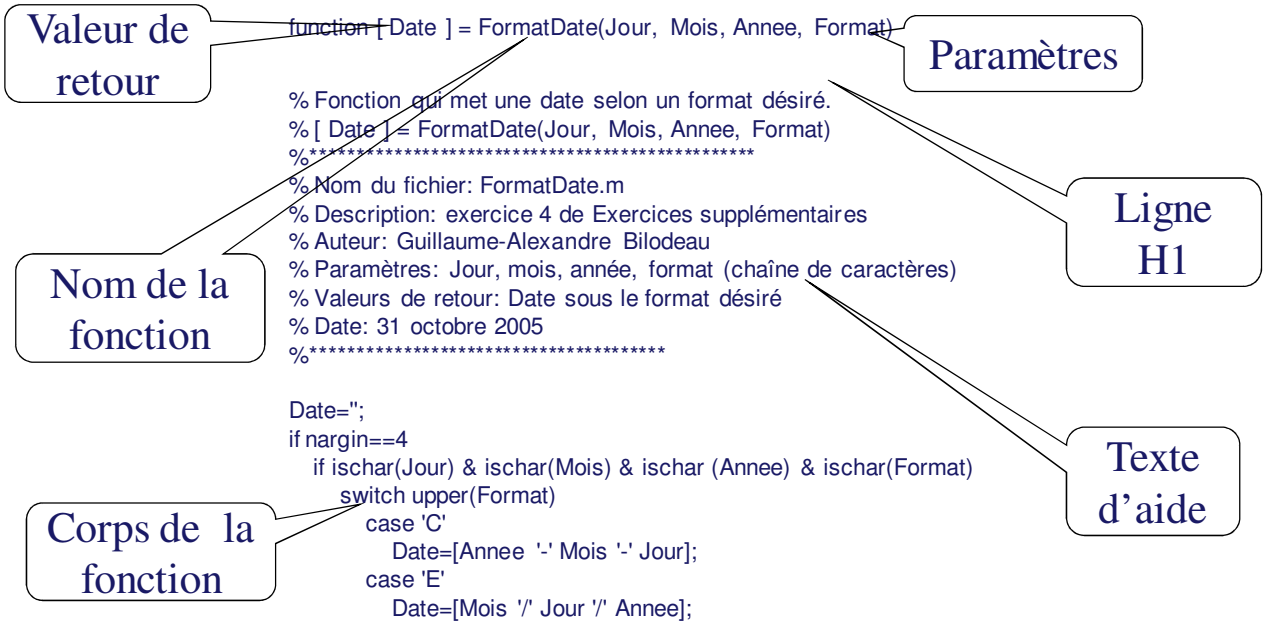
- Pour ne pas définir de valeur de retour, il suffit de mettre des crochets vides.

```
function [ ] = toto2(a)
```



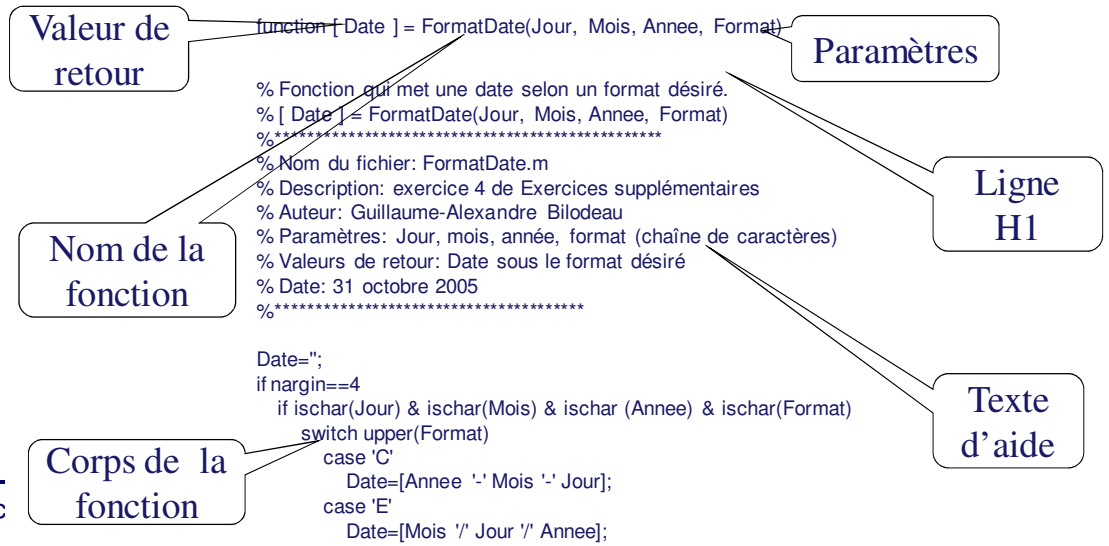
Structure d'une fonction – ligne H1

- La ligne H1 est la première ligne de commentaires située juste après la ligne de définition de la fonction.
- Elle résume l'opération fait par fonction en UNE ligne.
- Utilisée notamment par la fonction `lookfor()` pour rechercher des fonctions.
- Cette ligne est aussi affichée lors de l'utilisation de la fonction `help()`.



Structure d'une fonction – texte d'aide

- Le texte d'aide est composé de toutes les lignes de commentaires situées entre la ligne de définition de fonction et la première ligne du code exécutable.
- Affiché lors de l'appel de la fonction `help()`.
- Pour cette raison, il doit décrire en détail les fonctionnalités et les particularités de l'utilisation de la fonction.
- Ne doit pas être trop long. Doit être clair et concis.



Agenda

Définition, introduction

Structure d'une fonction

- L'appel d'une fonction
- `nargin() / nargout()`
- Fonctions avec un nombre indéfini de paramètres / valeurs de retour
- Autres fonctions, particularités appel d'une fonction en MATLAB

L'appel d'une fonction – présentation

- Une fonction est appelée soit de la fenêtre de commande de MATLAB, soit d'un fichier script.
- Quand une fonction est exécutée, elle est précompilée et stockée en mémoire par MATLAB pour s'exécuter plus rapidement au prochain appel.
- Exemple d'appel:

```
arg1 = [1 4 5 7];
reponse = moyenne(arg1)
```

donne

```
reponse = 4.2500
```

L'appel d'une fonction – arguments

- Le nombre d'arguments lors de l'appel doit être inférieur ou égal au nombre de paramètres définis pour la fonction.
- L'ordre des arguments dans l'appel de la fonction est important puisque c'est par leur ordre qu'ils sont associés aux paramètres.
- Les noms des arguments lors de l'appel de la fonction sont généralement différents de ceux des paramètres, puisque ce sont des variables différentes même si elles portent le même nom.

L'appel d'une fonction – exemple

Fenêtre de commandes de MATLAB

```
>> Ladate=FormatDate('23','06','2006','C')
```

Appel de la fonction FormatDate()

Arguments de la fonction

```
function [ Date ] = FormatDate(Jour, Mois, Annee, Format)

% Fonction qui met une date selon un format désiré.
% [ Date ] = FormatDate(Jour, Mois, Annee, Format)
%*****************************************
% Nom du fichier: FormatDate.m
% Description: exercice 4 de Exercices sur les commentaires
% Auteur: Guillaume-Alexandre Bilodeau
% Paramètres: Jour, mois, année (tous en
      caractères)
% Valeurs de retour: Date
% Date: 31 octobre 2005
%*************************************

Date='';
if nargin==4
   if ischar(Jour) & ischar(Mois) & ischar (Annee) &
      ischar(Format)
       switch upper(Format)
          case 'C'
             Date=[Annee '-' Mois '-' Jour];
          case 'E'
             Date=[Mois '/' Jour '/' Annee];
```

Paramètres de la fonction

Agenda


Définition, introduction


Structure d'une fonction


L'appel d'une fonction

- `nargin() / nargout()`
- Fonctions avec un nombre indéfini de paramètres / valeurs de retour
- Autres fonctions, particularités appel d'une fonction en MATLAB

nargin() / nargout()

Fonctions MATLAB qui permettent de connaître le nombre d'arguments reçus et le nombre de valeurs à retourner.

Étant donné que les arguments et les valeurs de retour sont optionnels dans MATLAB, il est conseillé de toujours vérifier leur nombre avec ces fonctions.

Exemple:

```
function [] = Affiche(Message)
...
if nargin() < 1   % ou nargin < 1
  fprintf('Vous devez donner une chaîne de
  caractères comme arguments');
end
if nargin() ==1 % ou nargin == 1
  fprintf('Voici le message: %s',Message)
end
```

nargin() / nargout() – exemple

```
function [Nombre1,Nombre2] = Tirage()
Nombre1=0; Nombre2=0;
if nargout()==1   % ou nargout == 1
  Nombre1=floor(rand(1)*10);
end
if nargout() ==2 % ou nargout == 2
  Nombre1=floor(rand(1)*10);
  Nombre2=floor(rand(1)*10);
end
```

Agenda


Définition, introduction


Structure d'une fonction


L'appel d'une fonction


`nargin() / nargout()`

- Fonctions avec un nombre indéfini de paramètres / valeurs de retour

- Autres fonctions, particularités appel d'une fonction en MATLAB

Fonctions avec un nombre indéfini de paramètres / valeurs de retour

Il est possible de créer des fonctions qui utilisent un nombre indéfini de paramètres et de valeurs de retour (i.e. `fprintf()`).

Il faut alors utiliser les ensembles de cellules réservés `varargin` et `varargout`.

Exemple:

```
function valeur_retour = nom_fonction(varargin)
```

Pour les manipuler, il faut utiliser la syntaxe réservée aux ensembles de cellules.

Pour connaître le nombre de paramètres contenus dans varargin, il faut utiliser la fonction `length()`.

Fonctions avec un nombre indéfini de paramètres - varargin

```
testvar([-1 0],[3 -5],[4 2],[1 1])
```

Exemple d'appel de la fonction

```
function []=testvar(varargin)
for k = 1:length(varargin)
   x(k) = varargin{k}(1);
   y(k) = varargin{k}(2);
end
xmin = min(0,min(x));
ymin = min(0,min(y));
axis([xmin fix(max(x))+3 ymin fix(max(y))+3]);
plot(x,y)
```

Récupération des paramètres

Traçage du graphique à partir des vecteurs reçus en paramètres

Fonctions avec un nombre indéfini de valeurs de retour - varargout

`function varargout = nom_fonction(n)`

Pour retourner les résultats au point d'appel, il faut connaître avec combien de valeurs de retour la fonction a été appelée en utilisant `nargout()`.

Il faut ensuite utiliser les opérations sur les cellules.

```
a = [1 2; 3 4; 5 6; 7 8; 9 0];
    [p1, p2, p3, p4, p5] = testvar2(a)
    p1 =  1  2    p4 = 7  8
    p2 =  3  4    p5 = 9  0
    p3 =  5  6
function [varargout] = testvar2(Matrice)
    for k = 1:nargout
          varargout{k} = Matrice(k,:)
    end
```

Appel de la fonction

Résultat de la fonction

Matrice reçue en paramètre

Nombre de valeurs de retour

Cellule k

Agenda

- Définition, introduction
- Structure d'une fonction
- L'appel d'une fonction
- `nargin()` / `nargout()`
- Fonctions avec un nombre indéfini de paramètres / valeurs de retour
- Autres fonctions, particularités appel d'une fonction en MATLAB

Libérer la mémoire allouée pour une fonction - clear

Pour effacer une fonction stockée en mémoire, la fonction `clear`

Paramètres possibles:

`Nom de la fonction` : Efface seulement une fonction spécifique.

```
clear Tirage;
```

`functions` : Efface toutes les fonctions pré-compilées en mémoire.

```
clear functions;
```

`all` : Efface toutes les fonctions pré-compilées et les variables.

```
clear all;
```

Appel d'une fonction – particularité de MATLAB

Pour les fonctions qui n'ont pas de paramètres, il n'est pas nécessaire de faire l'appel en mettant les parenthèses.

Soit la définition suivante:

```
function [nombre]=Tirage()
```

L'appel peut être fait par:

```
>> chiffre=Tirage  % Peu porter à confusion.
        % Variable ou fonction ? Pas recommandé.
>> chiffre=Tirage()
```

Cela explique:

`nargin` ou `nargin()`,

`help` ou `help()`, etc.

Appel d'une fonction – particularité de MATLAB

Pour les fonctions qui ne reçoivent comme paramètres que des chaînes de caractères, on peut faire l'appel sans parenthèses.

Soit la définition suivante:

```
function [ Date ] = FormatDate(Jour, Mois, Annee, Format)
```

L'appel peut être fait par:

```
>> Ladate=FormatDate('2007','01','01','C')  % ou

>> Ladate=FormatDate 2007 01 01 C  %Les arguments
% sont convertis automatiquement en chaînes de
%  caractères avant d'être transmis à la fonction.
```

Cela explique l'utilisation de

`format short g` ou `format('short','g')`, etc.

Agenda

- Définition, introduction
- Structure d'une fonction
- L'appel d'une fonction
- `nargin()` / `nargout()`
- Fonctions avec un nombre indéfini de paramètres / valeurs de retour
- Autres fonctions, particularités appel d'une fonction en MATLAB

Sommaire