

Mémo Matlab (1/2)

- Bonnes pratiques :

- Se placer dans un dossier dédié
- Toujours écrire dans un script
- Surveiller le workspace pour voir l'état et surtout la taille des variables
- Consulter l'aide MATLAB pour utiliser une fonction (doc `fonction`)
- Ne pas relancer tout le script à chaque fois

- Script :

- Un script est toujours un fichier d'extension `.m`
- Toujours commencer son script par :

```
clear; close all; clc
```

- **ctrl+S** pour sauvegarder le fichier

- Chemins :

- Pour ajouter des dossiers aux chemins connus :

```
addpath('images/');  
%Image stockée ici images/img.png  
img = imread('img.png');
```

- Fonctions :

- A écrire dans un fichier qui porte le nom de la fonction.

Exemple : `min_max.m`

```
function [min, max] = min_max(T)  
    min = min(tab(:));  
    max = max(tab(:));  
end
```

- Exécution :

- Tout le script : **F5** ou **<Run>**
- Par section : **ctrl+enter** ou **<Run section>**

```
[h,w,c] = size(img);
```

```
%% Affichage  
figure, imagesc(img)
```

```
%% Vectorisation  
img_vect = img(:);
```

- Par sélection : **F9**

```
[h,w,c] = size(img);
```

```
%Affichage  
figure, imagesc(img)
```

Mémo Matlab (2/2)

- Commandes de bases :

%Manipulation d'image (format uint8)

```
img = imread('img.png');  
[h,w,c] = size(img) ← affichés dans  
la console
```

%Vectorisation

```
img_vect = img(:);  
max_img = max(img_vect);
```

%Accès dimensions

```
G = img(:,:,2); %canal vert
```

%Mise à zéro

```
img = zeros(h,w,c); %ones() existe aussi  
img = img*0;
```

%Sous-échantillonnage

```
img = imread('img.png');  
figure, imagesc(img(1:2:h,1:4:end,:))
```

%Création d'un vecteur/d'une matrice

```
mat = [1 1; 2 2; 3 3] %de taille 3x2
```

%Produits vecteurs/matriciels

```
vect = (1:2:11); % (début:(pas):fin)  
vect_2 = vect.*vect; %terme à terme  
vect_x = vect*vect'; %produit matriciel  
↑  
transposée
```

%Somme des termes au carrés

```
sum_G = sum(G(:).^2); %équivalents  
sum_G = sum(G(:).*G(:)); %équivalents
```

%Seuillage sur une matrice

```
mask = G > 100; %mask = carte binaire (hwx)
```

%Équivalent à faire :

```
mask = zeros(h,w);  
for i=1:h  
    for j=1:w  
        if (G(i,j)>100)  
            mask(i,j) = 1;  
        end  
    end  
end
```

```
G(mask==0) = 0; %Mise à zéro des pixels  
%de G où mask est nul (0)
```

```
G = G.*mask; %Equivalent à mul. terme à terme
```

%Affichage

```
img_L = (img(:,:,1)+img(:,:,2)+img(:,:,3))/3;  
figure, subplot(121) %Affichage multiples 1x2  
plot(img_L(1,:))  
title('Profil de la première ligne de L')  
subplot(122)  
plot(img(1,:,1),'r'); hold on; %superposition  
plot(img(1,:,2),'g'); hold on;  
plot(img(1,:,3),'Color',[0 0 1]);  
title('Profil RGB de la première ligne');  
xlabel('x'); ylabel('Intensité');
```

Autres fonctions utiles :

squeeze, repmat, ginput, find, ...