

Image binaire :

```
clc; clear all;  
im1=imread('text.png');%en workspace la matrice est de type logical donc im1 est  
binaire sa matrice contient seulement les valeurs 0 et 1.%  
figure(1);  
imshow(im1);
```

```
val=im1(:);%transforme une matrice en un vecteur%  
M=unique(val);%supprime les valeur doubles%
```

Workspace	
Name ▲	Value
<input checked="" type="checkbox"/> im1	256x256 logical
<input checked="" type="checkbox"/> M	2x1 logical
<input checked="" type="checkbox"/> val	65536x1 logical

The term watershed
refers to a ridge that ...

... divides areas
drained by different
river systems.

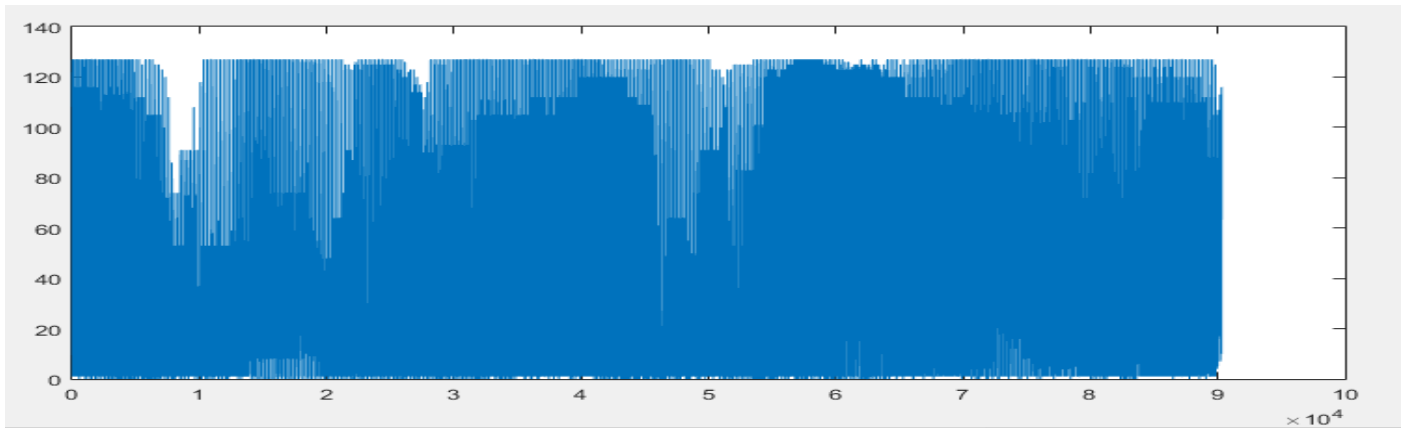
```
im2=imread('trees.tif')  
figure(1);imshow(im2);
```

Workspace	
Name ▲	Value
<input checked="" type="checkbox"/> im2	258x350 uint8



Image en niveau de gris :

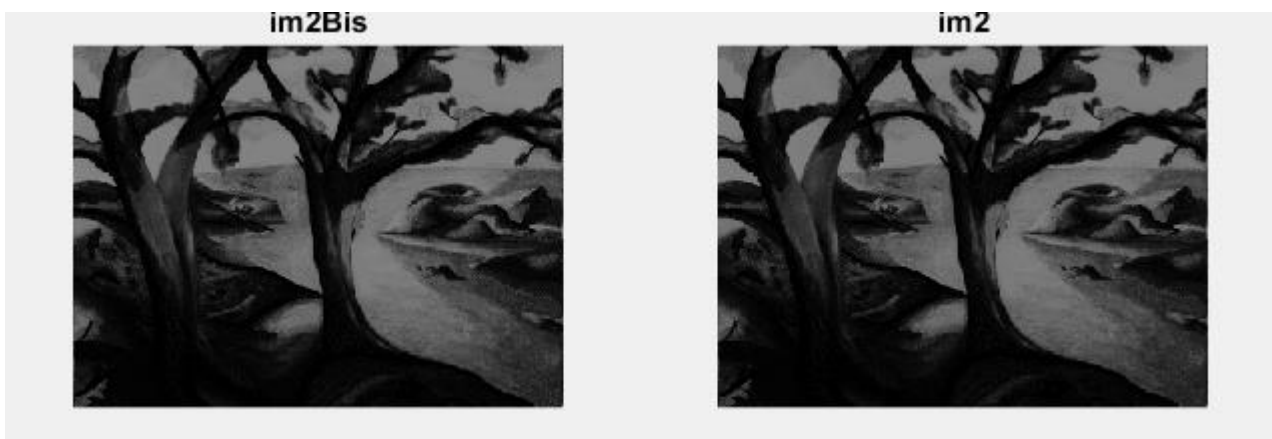
```
val2=im2(:);  
M2=unique(val2);figure(1); plot(val2) ; %donne en axe des abscisses nbre de pixels  
telque length(val2)=Nlig*Ncol de im2 et en axe des ordonnées les niveaux de gris%
```



Convertir les niveaux de gris de [0 255] à [0 1] :

```
im2=imread('trees.tif');
im2Bis=double(im2)/255;
figure(1);
subplot(1,2,1); imshow(im2Bis); title('im2Bis');
subplot(1,2,2); imshow(im2); title('im2');
val3=im2Bis(:);
figure(2); plot(val3);
```

Workspace	
Name ▲	Value
im2	258x350 uint8
im2Bis	258x350 double
val3	90300x1 double



On remarque que les niveaux de gris sont entre 0 et 1

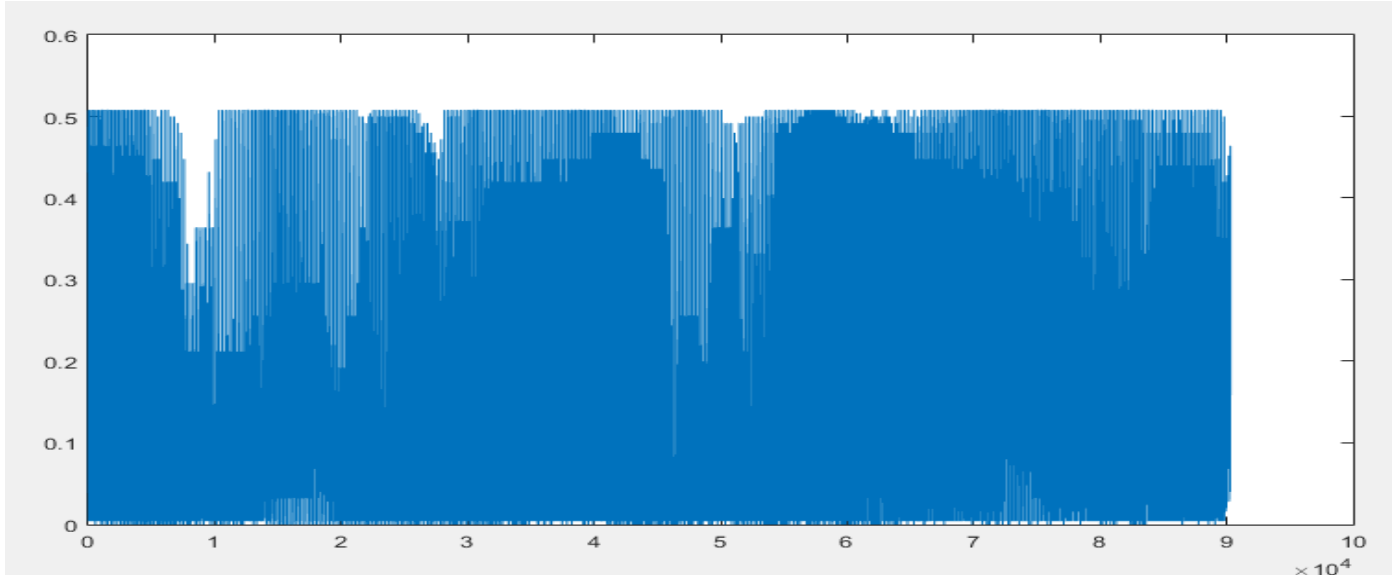
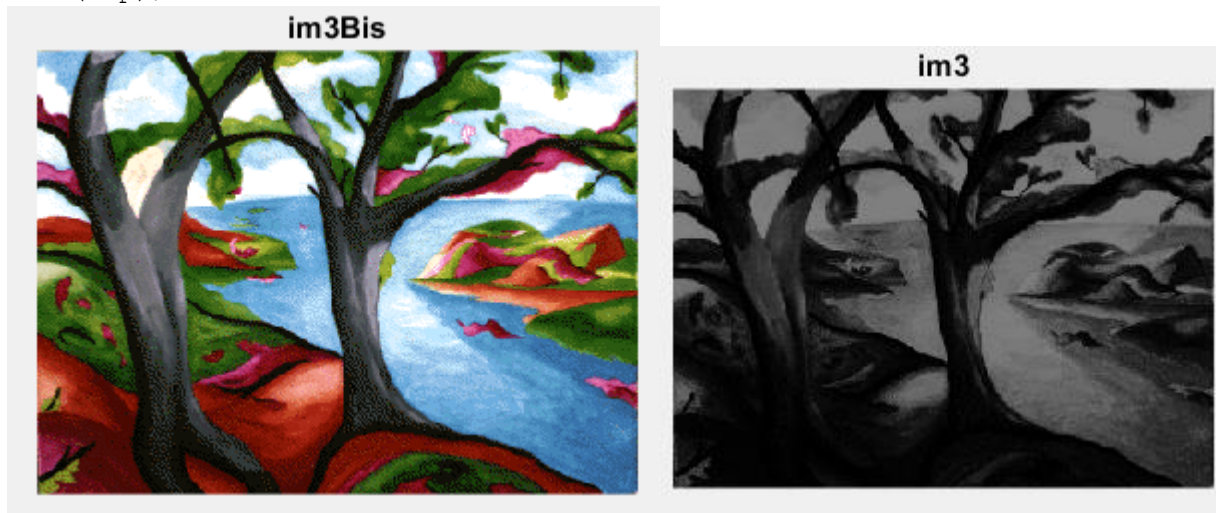


Image indexée vers l'image RGB :

```
clear all;
[im3,map]=imread('trees.tif');%im3 en niveau de gris et
map matrice de composition de couleurs %
im3Bis=ind2rgb(im3,map);%convertir l'image indexé im3
sa colormap correspondante map en une image RGB avec
ses vrais couleurs
figure(1); imshow(im3Bis);title('im3Bis');
figure(2); imshow(im3);title('im3');
size(map);
```

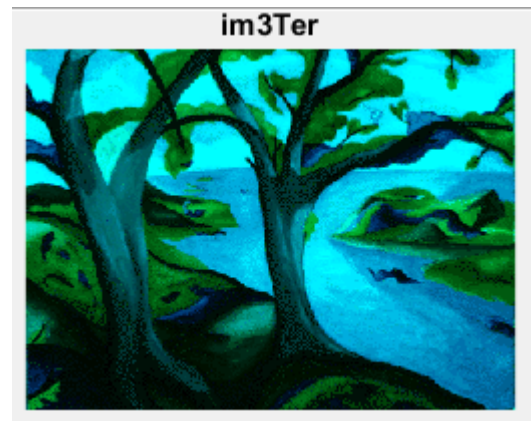
Workspace	
Name ▲	Value
ans	[256 3]
im3	258x350 uint8
im3Bis	258x350x3 double
map	256x3 double



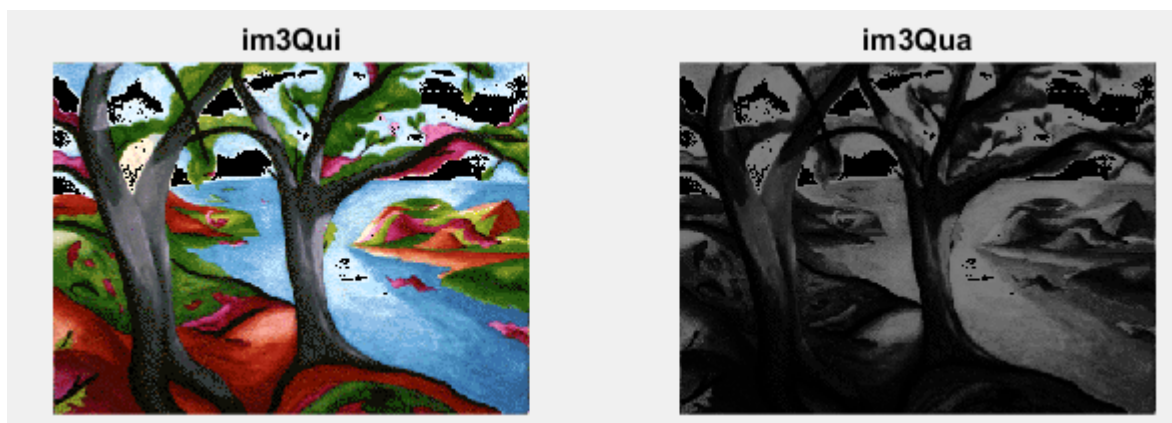
Annulation d'une composante couleur:

```
mapBis=map;
mapBis(:,1)=0;
im3Ter=ind2rgb(im3,mapBis);
figure(3);imshow(im3Ter);title('im3Ter');
%on donne 0 à tous les composantes rouge%

im3Qua=im3; im3Qua(im3==127)=0;
im3Qui=ind2rgb(im3Qua,map);
figure(4);subplot(1,2,1);
imshow(im3Qui);title('im3Qui');
subplot(1,2,2); imshow(im3Qua); title('im3Qua');
```



%on a annulé les niveaux de gris égaux à 127 ds l'image indexé im3Qua puis on a converti im3Qua en une image couleur im3Qui%



```
>> map(1,:)
```

```
ans =
```

```
0    0    0
```

```
>> map(128,:)
```

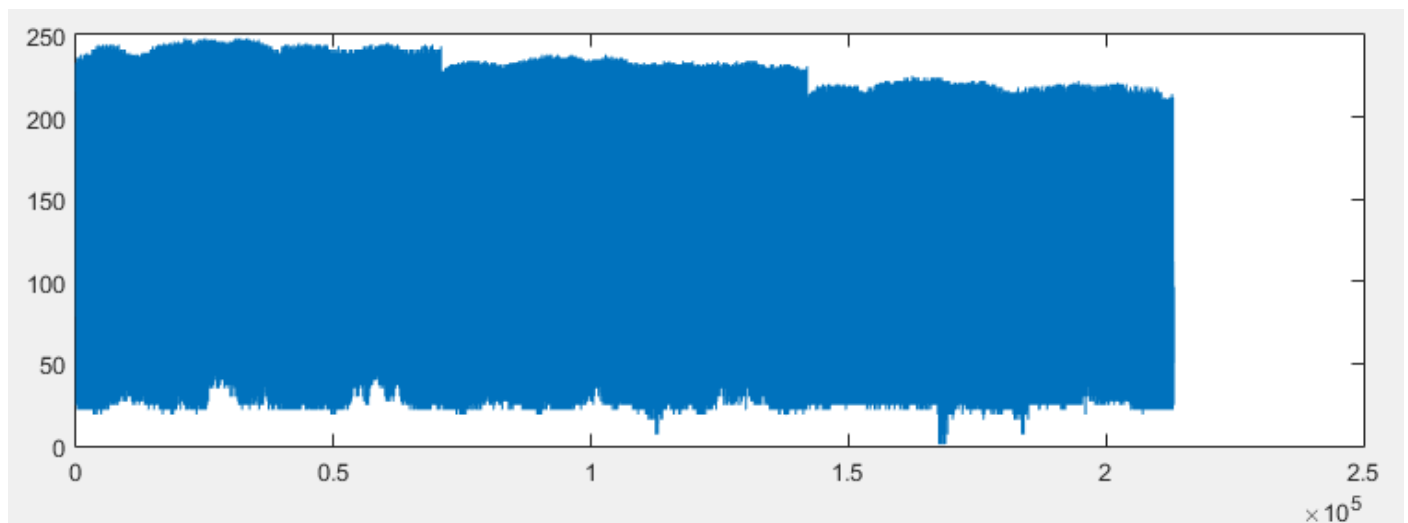
```
ans =
```

```
1    1    1
```

EXEMPLE 'autumn.tif':

```
clear all;  
im4=imread('autumn.tif');  
figure(1);imshow(im4);  
val4=im4(:);  
figure(2);  
plot(val4);  
size(im4) ;
```

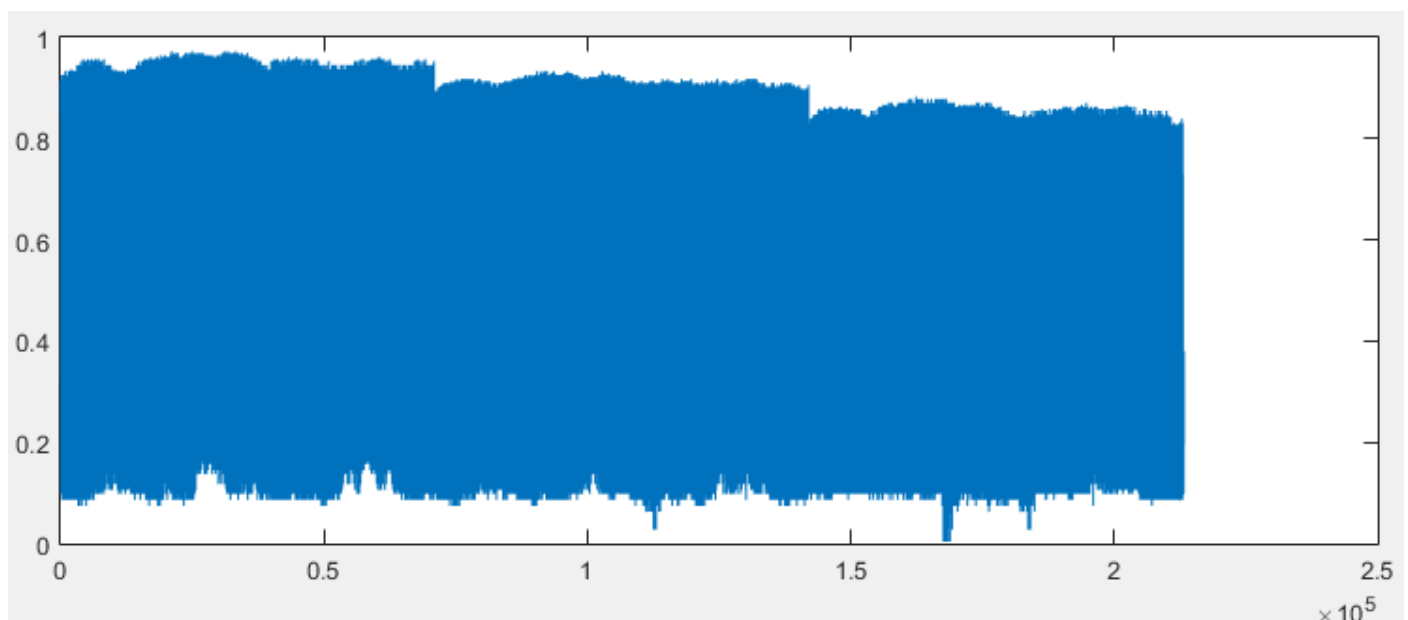
Workspace	
Name ▲	Value
ans	[206 345 3]
im4	206x345x3 uint8
val4	213210x1 uint8



```
im4Bis=double(im4)/255;  
figure(3); imshow(im4Bis); title('im4Bis');  
val5=im4Bis(:);  
figure(4); plot(val5);  
%on remarque bien que les niveaux de gris de  
im3Bis sont entre 0 et 1 puisque im3Bis est de  
Double%
```

Workspace	
Name ▲	Value
im4	206x345x3 uint8
im4Bis	206x345x3 double
val5	213210x1 double

type



Donne le max suivant la dimension 1 nbre de ligne
 puis suivant dimension 2 nbre de colonne puis
 suivant dimension 3 nbre de matrice

```
>> max(max(max(im4)))

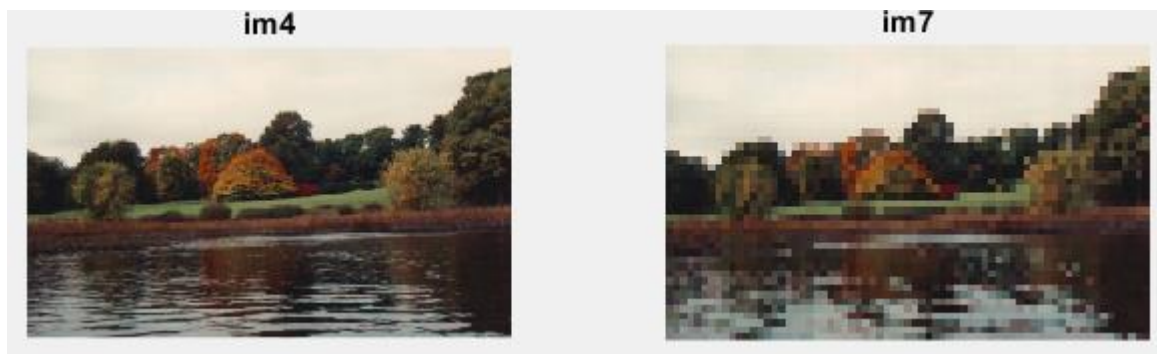
ans =

    248
```

Reduction de nbre de pixels :

```
im7=im4(1:5:end,1:5:end,:);
figure(1); subplot(1,2,1); imshow(im4);title('im4');
subplot(1,2,2); imshow(im7);title('im7');
%le nbre de pixels(resp lignes et colonnes) de im7 est plus
petit que celle de im4 cette transformation est bien
traduit dans l'image , on remarque im7 est plus floue
(résolution affaiblie)%
```

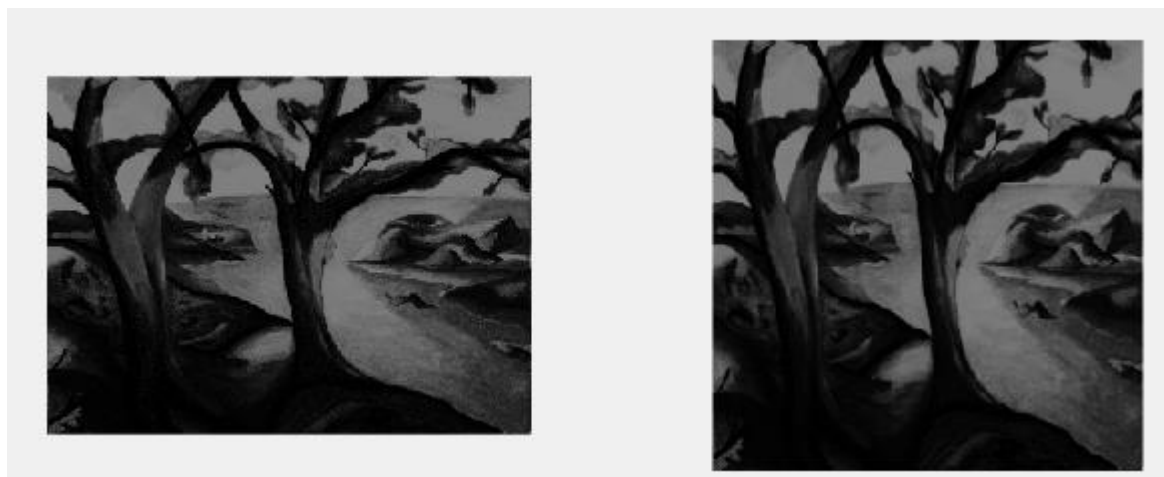
Workspace	
Name ▲	Value
im4	206x345x3 uint8
im7	42x69x3 uint8



Modification de taille de l'image :

```
im2=imread('trees.tif');
im8=imresize(im2,[256 256]);
figure(1);subplot(1,2,1); imshow(im2);
subplot(1,2,2); imshow(im8);
% la taille de im4 est réduite en im8(nbre de ligne
et colonnes plus petit que celles de im4)%
```

Workspace	
Name ▲	Value
im2	258x350 uint8
im8	256x256 uint8



La commande imresize ne fonctionne pas sur une image à 3 dimensions image couleur

```
im10=zéros(256,256,3);
im10(:,:,1)=imresize(im4Bis(:,:,2),[256 256]);%extraire la matrice rouge et ajuster sa
taille puis l'affecter à im10 en dimension1%
im10(:,:,2)=imresize(im4Bis(:,:,1),[256 256]);
im10(:,:,3)=imresize(im4Bis(:,:,3),[256 256]);%inverser le rouge en vert%
figure(1); subplot(1,2,1); imshow(im4); subplot(1,2,2); imshow(im10);
```

Workspace	
Name ▲	Value
im10	256x256x3 double
im4	206x345x3 uint8
im4Bis	206x345x3 double

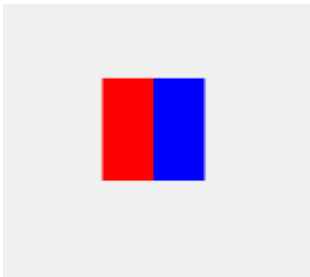


Creation d'une image:

```
im5=zeros(10,10,3);
im5(:,:,1)=[ones(10,5) zeros(10,5)];
im5(:,:,3)=[zeros(10,5) ones(10,5)];
figure;
imshow(im5);
```

Workspace	
Name ▲	Value
im5	10x10x3 double

%im5 image indexée initialisée en noir puis on lui affecte les couleurs en manipulant les matrices le rouge et le bleu%



```
im6= rgb2gray(im4);
figure;
subplot(1,2,1);
imshow(im4);
subplot(1,2,2);
imshow(im6);
```

