

# Computação Gráfica (IME 04-10842)

## Introdução ao Octave/Matlab

### Exercícios:

1. Confira os tipos de dados do Octave/Matlab através do comando:

```
help datatypes
```

2. Tipos de imagem:

#### RGB

- Crie 3 matrizes (RED, GREEN e BLUE), com as mesmas dimensões, contendo valores entre 0 e 1.

Exemplo (três matrizes 3x3):

```
RED = [1 0 0; 1 0 1; 1 0 0.5];  
GREEN = [0 0 1; 1 0 0; 1 1 0.5];  
BLUE = [0 0 0; 0 1 1; 1 1 0.5];
```

- Para ver o conteúdo das matrizes, digite seus nomes e, em seguida *Enter*. É muito importante notar que o Octave/Matlab é *case sensitive*.
- Concatenar as três matrizes para criar uma única matriz tridimensional (RGB) usando a função `cat` (usar a ajuda para aprender sobre a função: `help cat`).

Exemplo:

```
RGB = cat (3, RED, GREEN, BLUE);
```

- Visualize o conteúdo da matriz RGB. Os dois primeiros índices dos elementos da matriz definem, respectivamente, os números de linha e coluna. O terceiro índice refere-se a uma das três matrizes originais. Se considerarmos agora matriz RGB como uma imagem, os dois primeiros índices definem as coordenadas de um pixel, e o terceiro, um dos componentes RGB do pixel.

Exemplo (experimente os comandos abaixo):

```
RGB (:, :, 1)  
RGB (:, :, 2)  
RGB (:, :, 3)
```

- Visualize a matriz resultante com a função:

```
imshow (RGB);
```

#### Imagem indexada

- Crie uma matriz com o nome MAP, com três linhas e três colunas e valores entre 0 e 1, para representar um *mapa de cores*. Crie uma matriz bidimensional com o nome X, com valores inteiros entre 1 e 3. Visualize a imagem indexada usando:

```
imshow (X, MAP);
```

# Computação Gráfica (IME 04-10842)

## Introdução ao Octave/Matlab

### Imagem de Níveis de Cinza

- Crie uma matriz X com valores entre 0 e 1.
- Visualize a matriz X como uma imagem de níveis de cinza (intensidade). Experimente os comandos abaixo:

```
imshow (X);  
imshow (X, [0 1]);  
imshow (X, [0 2]);  
imshow (X, [0 16]);
```

### Imagem Binária

- Crie uma matriz X usando exclusivamente valores 0 ou 1 (imagem binária).
- Visualize a matriz X usando o comando:

```
imshow (X);
```

## 3. Leitura e gravação de imagens:

### Leitura de Imagens

- Estudar a função `imread` usando a ajuda do Octave/Matlab.
- Leia as imagens *arara\_full.png* e *arara\_full\_256.bmp* usando a função `imread`.
- Tente descobrir, com a ajuda da função `whos` se as imagens são RGB ou indexadas (para saber mais sobre a função use `help whos`).
- Visualize as imagens usando a função `imshow`.

### Escrita de Imagens

- Gravar em disco cópias das imagens lidas no exercício anterior com as extensões trocadas (*.bmp* ↔ *.png*) usando a função `imwrite`.
- Ler as imagens gravadas (`imread`) e visualize-as (`imshow`).

## 4. Conversão de Imagens:

- Aprenda a usar as funções `ind2gray`, `gray2ind`, `rgb2ind`, `ind2rgb`, `rgb2gray` e `gray2rgb`. Observe os tipos de dados (`double` ou `uint8`) dos elementos das matrizes resultantes.
- Transformar as imagens criadas nos exercícios anteriores em diferentes tipos de imagem e visualizar as imagens resultantes.

## 5. Visualização:

- Aprenda a usar a função `figure`. Mostrar duas imagens diferentes em diferentes janelas.

# Computação Gráfica (IME 04-10842)

## Introdução ao Octave/Matlab

- Aprenda a usar as funções `subplot` e `subimage`. Mostrar as duas imagens de exercícios anteriores em uma única janela usando estas funções.
- Aprenda a usar a função `title`. Coloque títulos nas janelas dos exercícios anteriores.

### 6. Matrizes:

- Criar a matriz `A` usando o seguinte comando:

```
A = [16 3 2 13 19; 5 10 11 8 3; 6 7 9 12 8; 4 15 14 1 13; 1 2 3 4 5]
```

- Calcular a soma das quatro células dos cantos da matriz, referenciando os respectivos índices.
- Verifique a saída dos seguintes comandos:

```
A
```

```
A(1,3) + A(3,1)
```

```
A(1:3,3)
```

```
A(1:4,2:4)
```

```
A(:,3)
```

```
A(1:3,:) 
```

Criar uma matriz 5x3 `B`, e aplicar o comando `A(:, [3 5 2]) = B(:, 1:3)`

```
A(:)
```

```
A'
```

```
A(:, [1 2 2 3 3 3 4 4 4 4])
```

```
A([1 2 2 3 3 3 4 4 4 4], :)
```

```
A(A>10) = 0
```

- Estude os comandos `zeros`, `ones`, `eye`, `size`.
- Use `help arith` para aprender sobre os operadores: `+`, `-`, `*`, `/`, `./`, `^`, `*`, `^.`, `./`.
- Experimente com os operadores: `*`, `.*`, `/`, `./`