

INTRODUÇÃO À PROGRAMAÇÃO

Aula 22

A função logical

```
>> x = -2:2
```

```
x =
```

```
-2 -1 0 1 2
```

```
>> k = (abs(x) > 1)
```

```
k =
```

```
1×5 logical array
```

```
1 0 0 0 1
```

```
>> z = x(k)
```

```
z =
```

```
-2 2
```

A função logical

```
>> x = -2:2
```

```
x =
```

```
    -2    -1     0     1     2
```

```
>> k = (abs(x) > 1)
```

```
k =
```

```
1x5 logical array
```

```
    1     0     0     0     1
```

```
>> z = x(k)
```

```
z =
```

```
    -2     2
```

A função logical

```
>> w = [1, 0, 0, 0, 1]
```

```
w =
```

```
1  0  0  0  1
```

```
>> v = x(w)
```

Subscript indices must either be real positive integers or logicals.

A função logical

” A função **logical** retorna um arranjo que pode ser utilizado para indexação lógica e testes lógicos

```
>> r = logical([1,0,0,0,1])
```

```
r =
```

```
1×5 logical array
```

```
1  0  0  0  1
```

```
>> v = x(r)
```

```
v =
```

```
-2   2
```

A função logical

```
>> y = logical(9)
```

```
y =
```

```
logical
```

```
1
```

```
>> y = logical([9,7])
```

```
y =
```

```
1×2 logical array
```

```
1 1
```

A função logical

” A função **double** converte um arranjo lógico em um arranjo da classe double

”Double - variável numérica de precisão dupla, 15 algarismos significativos.

```
>> clear
```

```
>> x = (5 > 3)
```

```
x =
```

```
logical
```

```
1
```

```
>> y = double(x)
```

```
y =
```

```
1
```

A função logical

“Algumas operações aritméticas convertem um arranjo lógico em um arranjo numérico.

```
>> clear
```

```
>> r = logical([1,0,0,0,1])
```

```
r =
```

1×5 logical array

```
1  0  0  0  1
```

```
>> r = r + 0
```

```
r =
```

```
1  0  0  0  1
```


Acessando arranjos utilizando arranjos lógicos

“ Quando um arranjo lógico é utilizado para endereçar outro arranjo, ele extrai daquele arranjo os elementos em que o arranjo lógico tiver 1s.

```
>> A = [5, 6, 7;8, 9, 10;11, 12, 13]
```

A =

```
5  6  7
8  9 10
11 12 13
```

```
>> eye(3)
```

ans =

```
1  0  0
0  1  0
0  0  1
```

```
>> B = logical(eye(3))
```

B =

3×3 logical array

```
1  0  0
0  1  0
0  0  1
```

```
>> C = A(B)
```

C =

```
5
9
13
```

Operadores Lógicos e funções lógicas

“ Há no MATLAB cinco operadores lógicos, que são algumas vezes chamados de operadores booleanos.

~ - NOT , & - AND, | - OR, && - Pequeno circuito AND

|| - Pequeno circuito OR

“ Esses operadores realizam operações elemento a elemento

“ Com exceção do operador NOT(~), eles tem um precedência menor que a dos operadores aritméticos e relacionais.

O Operador NOT(~)

```
>> clear
```

```
>> x = [0,3, 9];
```

```
>> y = [14, -2, 9];
```

```
>> u = x > y
```

u =

1×3 logical array

0 1 0

```
>> u = ~x > y
```

u =

1×3 logical array

0 1 0

O Operador NOT(~)

```
>> x = [0,3, 9];  
>> y = [14, -2, 9];
```

```
>> ~x
```

```
ans =
```

1×3 logical array

1 0 0

```
>> u = (~x) > y
```

```
u =
```

1×3 logical array

0 1 0

```
>> v = ~(x > y)
```

```
v =
```

1×3 logical array

1 0 1