



Programação em MATLAB

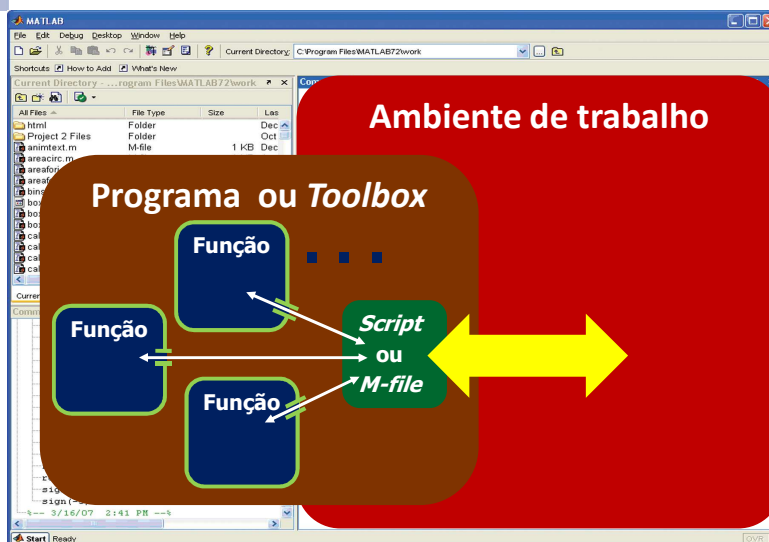
- **Desenvolvimento de software em MATLAB**

- *Scripts* ou *M-files*
- Funções definidas pelo programador
- **Estruturas de Selecção**

Instituto Superior Técnico,
Dep. de Engenharia Mecânica - ACCAII



Organização de programas no MATLAB



José Borges - Miguel Pedro Silva

Computação e Programação 2009 / 2010

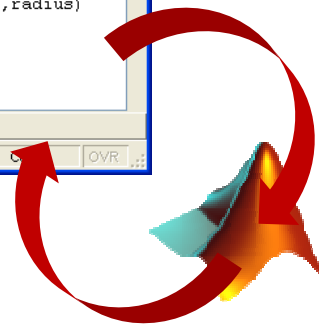
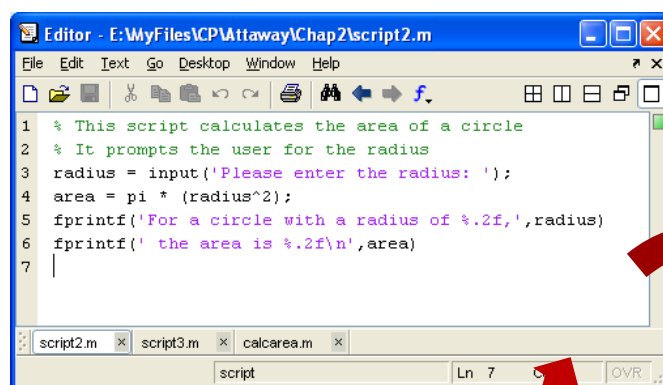
2

Scripts ou M-files

- Sequência de instruções que resolve um dado problema
- Pode chamar funções pré-definidas no ambiente ou funções definidas pelo programador
- Todas as variáveis são transparentes no interface, i.e. o script acede às variáveis do ambiente e vice-versa



Exemplo de script



Funções

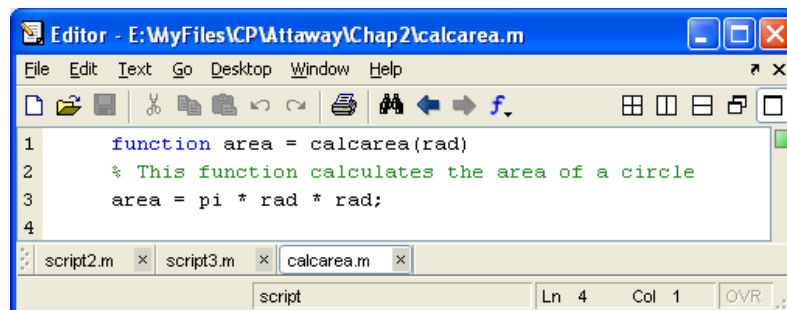
- Similar ao script em termos de funcionalidade, no entanto usa parâmetros de entrada e parâmetros de saída
- As variáveis são normalmente de âmbito local, com excepção das variáveis definidas no ambiente com a instrução **global**
- A passagem por valor das variáveis é feita no interface utilizando a instrução **function**



Função pré-definida **function**

Descrição: declaração da função

Sintaxe: `function [arg_out] = NomeFunção(arg_in)`

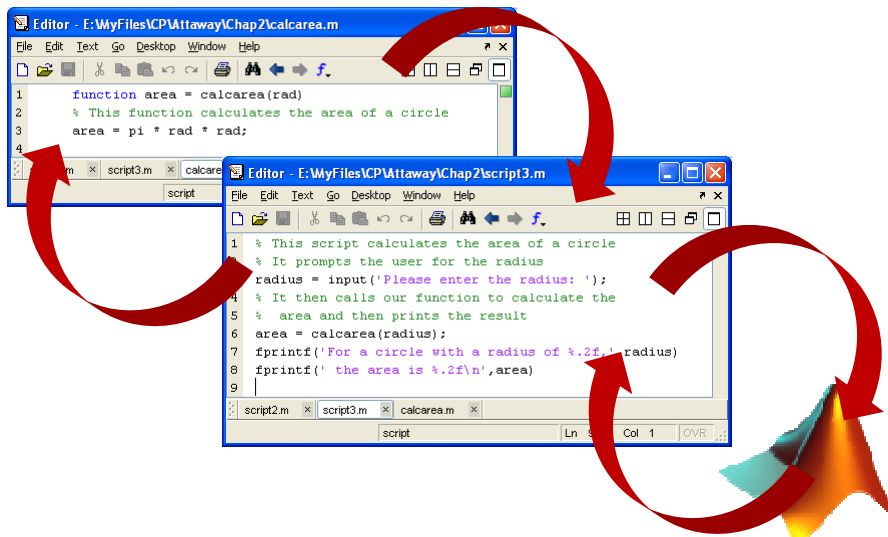


```

1 function area = calcarea(rad)
2 % This function calculates the area of a circle
3 area = pi * rad * rad;
4

```

Exemplo de função



José Borges - Miguel Pedro Silva

Computação e Programação 2009 / 2010

7

Âmbito das variáveis

- As variáveis podem ser **globais** ou **locais**
- Variáveis globais são **transparentes** a **todas as entidades**
- Variáveis locais **só são visíveis** dentro do âmbito em que estão **definidas**
- A **permuta do valor** das variáveis é feita via **interface**

Ambiente de trabalho

X = 1
y = 5
a = ?

X = 1
y = ?
a = ?

y ≡ b

a = 10
b = ?
X = ?

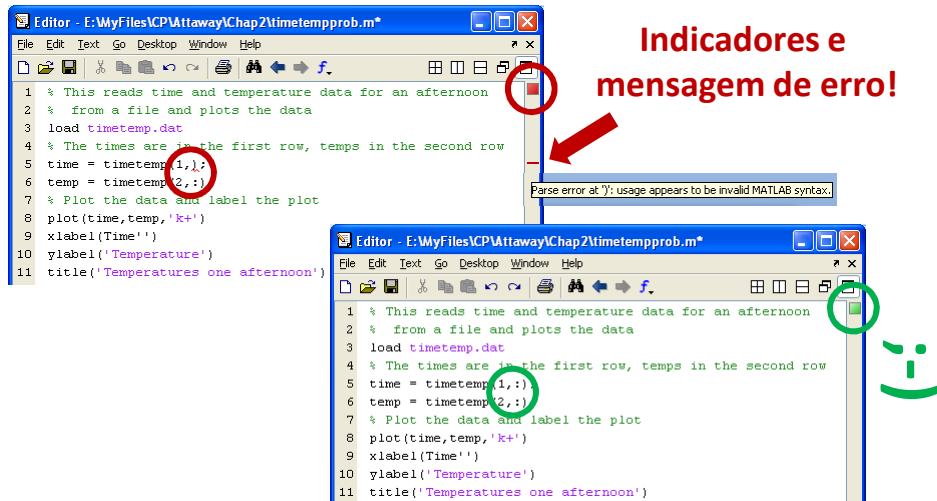
a = ?
X = ?
b = ?

José Borges - Miguel Pedro Silva

Computação e Programação 2009 / 2010

8

Ferramentas de depuração (debug)



Estruturas de Selecção

- Os programas apresentados até ao momento consistem de um conjunto de instruções executadas sequencialmente
- Na prática é comum ocorrerem excepções, o que implica a existência de **Estruturas de Selecção**, que permitam utilizar:
 - mecanismos de escolha: **executa** ou **não executa**
 - critérios de decisão baseados em **Expressões Relacionais**: **a maior que b** ou **x igual a y**
- As estruturas de selecção são implementadas no MATLAB utilizando :
 - instruções **if**
 - instruções **switch**
 - funções **is**



Expressões Relacionais

- Expressões que traduzem condições de decisão
- O resultado é um uma expressão booleana do tipo:
 - valor lógico **verdade** (ou **true**) = 1
 - valor lógico **falso** (ou **false**) = 0
- As expressões relacionais são constituídas por:
 - **operadores relacionais**, que relacionam duas expressões de tipos compatíveis
 - **operadores lógicos**, que relacionam os operandos



Operadores

Operadores Relacionais		Operadores Lógicos	
Operador	Descrição	Operador	Descrição
>	maior que		ou (disjunção)
<	menor que	or	ou (disjunção)
>=	maior ou igual que	xor	ou exclusivo
<=	menor ou igual que	&&	e (conjunção)
==	igual a	and	e (conjunção)
~=	diferente de	~	negação
		not	negação



Exemplos de operadores relacionais

```
>> 3 < 5
```

```
>> 9 < 2
```

```
>> 5 < 7
```

```
>> ans + 3
```

```
>> 'a' < 'c'
```



Exemplos de operadores lógicos

```
>> and(2>3, 5<6)
```

```
>> or(2>3, 5<6)
```

```
>> xor(2>3, 5<6)
```

```
>> not(2>3)
```

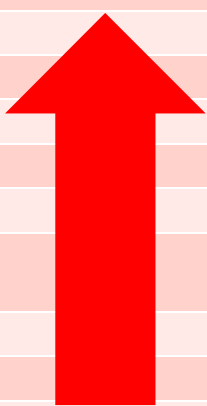


Tabelas de verdade

x	y	~x	x y	x&& y	xor(x,y)
true	true	false	true	true	false
true	false	false	true	false	true
false	false	true	false	false	false



Regras de precedência dos operadores

Operador	Símbolo	Precedência
parêntesis	()	
transposta, exponencial	' , ^	
negação	- , ~	
multiplicação, divisão	* , / , \	
adição, subtracção	+ , -	
operador coluna	:	
operadores relacionais	< , <= , > , >= , == , ~=	
conjunção	&&	
disjunção		
atribuição	=	
		Inferior



Exemplos

```
>> 4 > 3 + 1
```

```
>> 'e' == 'd' + 1
```

```
>> 3 < 9 - 2
```

```
>> 4 == 3 + 1 && 'd' > 'c'
```



Exemplos

```
>> 3 > 2 || 'x' == 'y'
```

```
>> xor(3 >= 2, 'x' == 'y')
```

```
>> xor(3 >= 2, 'x' ~= 'y')
```



Exemplos

Como implementar a expressão $3 < x < 5$ em MATLAB?

```
>> x = 1; , 3 < x < 5
ans =
     1
>> x = 4; , 3 < x < 5
ans =
     1
>> x = 7; , 3 < x < 5
ans =
     1
```

José Borges - Miguel Pedro Silva

Computação e Programação 2009 / 2010

19



Exemplos

Como implementar a expressão $3 < x < 5$ em MATLAB?

```
>> x = 1; , x > 3 && x < 5
ans =
     0
>> x = 1; , x > 3 && x < 5
ans =
     1
>> x = 1; , x > 3 && x < 5
ans =
     0
```

- $3 < x \ \&\& \ x < 5$
- `and(3 < x, x < 5)`
- `and(x > 3, x < 5)`

José Borges - Miguel Pedro Silva

Computação e Programação 2009 / 2010

20



Estrutura de selecção simples

- Escolhe se uma instrução, ou conjunto de instruções, deve, ou não, ser executada:

se **condição**
acção
terminar

- **condição:**
 - expressão relacional cujo resultado é um valor lógico
- **acção:**
 - instrução(ou conjunto de instruções) a ser executada(s)



Função pré-definida **if**

Descrição: escolhe se uma instrução, ou conjunto de instruções, deve, ou não, ser executada

Sintaxe: `if expression`
`statements`
`END`

```
>> num = -4.5;  
>> if num < 0  
    num = abs(num)  
end  
num =  
    4.5000
```

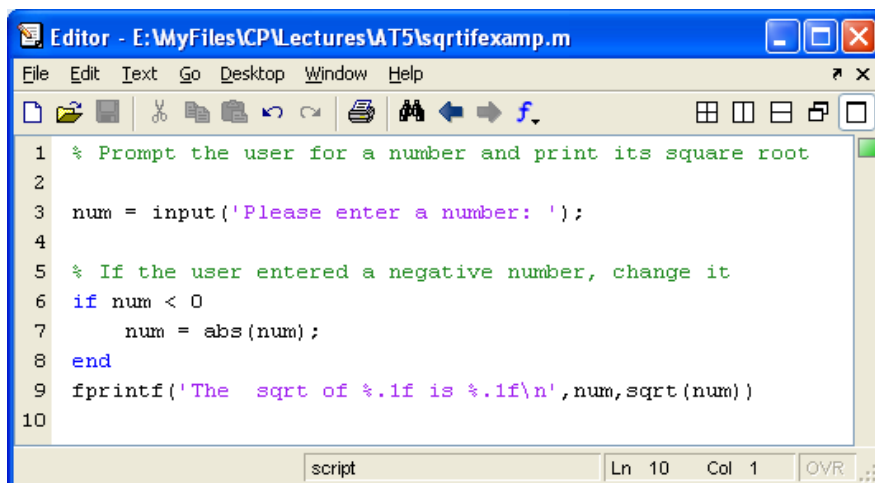


Função pré-definida **if**

```
>> num = -4;
>> if num < 0
    num = abs(num)
end
num =
    4
>> num = 5;
>> if num < 0
    num = abs(num)
end
>>
```



Função pré-definida **if**



```
Editor - E:\MyFiles\CPVLectures\AT5\sqrtifexamp.m
File Edit Text Go Desktop Window Help
1 % Prompt the user for a number and print its square root
2
3 num = input('Please enter a number: ');
4
5 % If the user entered a negative number, change it
6 if num < 0
7     num = abs(num);
8 end
9 fprintf('The sqrt of %.1f is %.1f\n', num, sqrt(num))
10
script Ln 10 Col 1 OVR
```

Função pré-definida **if**

```

>> sqrtifexamp
Please enter a number: -4.2
The sqrt of 4.2 is 2.0
>> sqrtifexamp
Please enter a number: 1.44
The sqrt of 1.4 is 1.2
  
```

```

2
3 num = input('Please enter a number: ');
4
5 % If the user entered a negative number, change it
6 if num < 0
7     num = abs(num);
8 end
9 fprintf('The sqrt of %.1f is %.1f\n', num, sqrt(num))
10
  
```

Função pré-definida **if**

```

Editor - E:\MyFiles\CP\Lectures\AT5\sqrtifexampii.m
File Edit Text Go Desktop Window Help
1 % Prompt the user for a number and print its square root
2
3 num = input('Please enter a number: ');
4
5 % If the user entered a negative number, tell
6 % the user and change it
7 if num < 0
8     disp('OK, we'll use the absolute value')
9     num = abs(num);
10 end
11 fprintf('The sqrt of %.1f is %.1f\n', num, sqrt(num))
  
```

Função pré-definida **if**

```
>> sqrtifexampii
```

```
Please enter a number: -25
```

```
OK, we'll use the absolute value
```

```
The sqrt of 25.0 is 5.0
```

```
1 % Prompt the user for a number and print its square root
2
3 num = input('Please enter a number: ');
4
5 % If the user entered a negative number, tell
6 % the user and change it
7 if num < 0
8     disp('OK, we'll use the absolute value')
9     num = abs(num);
10 end
11 fprintf('The sqrt of %.1f is %.1f\n', num, sqrt(num))
```

Valores lógicos no MATLAB

- Os valores lógicos no MATLAB têm a seguinte representação:
 - false** ⇔ valor inteiro 0
 - true** ⇔ qualquer valor inteiro diferente de 0

```
>> if 0
    disp('Yes, this is true!')
end
>>
>> if -1
    disp('Yes, this is true!')
end
Yes, this is true!
>> if 5
    disp('Yes, this is true!')
end
Yes, this is true!
```

Valores lógicos no MATLAB

```

1 - letter = input('Escolha uma tecla: ', 's');
2
3 - if letter == 'S' || letter == 's'
4 -     disp('Teste 1: carregou na tecla S')
5 - end
6
7 - if letter == 'S' || 's'
8 -     disp('Teste 2: carregou na tecla S')
9 - end
10
11 - if letter == ('S' || 's')
12 -     disp('Teste 2: carregou na tecla S')
13 - end

```

José Borges - Miguel Pedro Silva

Computação e Programação 2009 / 2010

29

Valores lógicos no MATLAB

```

>> testlogical
Escolha uma tecla: N
Teste 2: carregou na tecla S
>>

```

```

1 - letter = input('Escolha uma tecla: ', 's');
2
3 - if letter == 'S' || letter == 's'
4 -     disp('Teste 1: carregou na tecla S')
5 - end
6
7 - if letter == 'S' || 's'
8 -     disp('Teste 2: carregou na tecla S')
9 - end
10
11 - if letter == ('S' || 's')
12 -     disp('Teste 2: carregou na tecla S')
13 - end

```

José Borges - Miguel Pedro Silva

Computação e Programação 2009 / 2010

30



Referências

- Capítulos 2 e 3 de Stormy Attaway (2009), *“Matlab: A Practical Introduction to Programming and Problem Solving”*, Elsevier.