

# Comandos de entrada

Laura Goulart

UESB

16 de Fevereiro de 2016

Há duas formas de criarmos arquivos contendo comandos que serão executados posteriormente dentro do ambiente do Scilab:

Há duas formas de criarmos arquivos contendo comandos que serão executados posteriormente dentro do ambiente do Scilab:

- Scripts

Há duas formas de criarmos arquivos contendo comandos que serão executados posteriormente dentro do ambiente do Scilab:

- Scripts
- Funções

Por convenção, um script possui a extensão `.sce`. São arquivos criados no editor de texto do Scilab. Os comandos do arquivo de um script são executados automaticamente pelo Scilab.

**Exemplo:** Vamos fazer um script que calcula a área de uma circunferência de raio 3.

Uma função é um arquivo com a extensão `.sci`. O que distingue uma função de um script é que a primeira possui um ambiente local, mas se comunica com o prompt através de comandos de entrada e saída.

# Estrutura de uma função

```
function[ y1, y2, ... yn ]=nome( x1, x2, ... , xn )  
    sequência de comandos  
endfunction
```

Uma vez implementada uma função, é possível executar a qualquer momento a partir do nome dado a ela.



Um comando de entrada é utilizado para receber dados digitados pelo usuário. Esses dados recebidos são armazenados em variáveis. No Scilab teremos três comandos de entrada: `input`, `x_dialog`, `getvalue`. Todas as variáveis de um comando de entrada precisam de um texto.

O comando input é o comando de entrada com a sintaxe mais fácil e é o mais usado. Ele possui duas formas básicas:

O comando input é o comando de entrada com a sintaxe mais fácil e é o mais usado. Ele possui duas formas básicas:

- Variável numérica:  
`<variável>=input('texto')`

O comando input é o comando de entrada com a sintaxe mais fácil e é o mais usado. Ele possui duas formas básicas:

- Variável numérica:  
`<variável>=input('texto')`
- Variável não numérica(strings):  
`<variável>=input('texto','s')`

O comando `x_dialog` abre uma janela de diálogo para que o usuário entre com apenas um dado.

Sua sintaxe é da seguinte forma:

```
<variável>=evstr(x_dialog('texto','valor inicial'))
```

A função **evstr** é usada para converter os dados digitados pelo usuário para o formato numérico.

O comando `getvalue` abre uma janela de diálogo para que o usuário entre com dois ou mais dados. Sua sintaxe é da seguinte forma:

```
labels=['texto 1';...;'texto n']; <enter>  
[ok,<variável 1,...,variável n>]=getvalue('texto principal',labels,list(tipo 1,  
dimensão 1, ..., tipo n, dimensão n), ['valor inicial 1';...'valor inicial n'])
```

# Tipo de variáveis

- 'mat': matrizes



# Tipo de variáveis

- 'mat': matrizes
- 'col': vetores colunas

# Tipo de variáveis

- 'mat': matrizes
- 'col': vetores colunas
- 'row': vetores linhas

# Tipo de variáveis

- 'mat': matrizes
- 'col': vetores colunas
- 'row': vetores linhas
- 'vec': vetores

# Tipo de variáveis

- 'mat': matrizes
- 'col': vetores colunas
- 'row': vetores linhas
- 'vec': vetores
- 'str': string

**OBS1:** Um número real pode ser visto como um vetor de dimensão 1.

**OBS2:** É atribuído o valor -1 quando a dimensão é indefinida.