



UFOP



Aula Teórica 04

Material Didático

Conteúdos da Aula

- **Comando de Desvio de Fluxo**
- **Operadores Lógicos**
- **Exercícios**

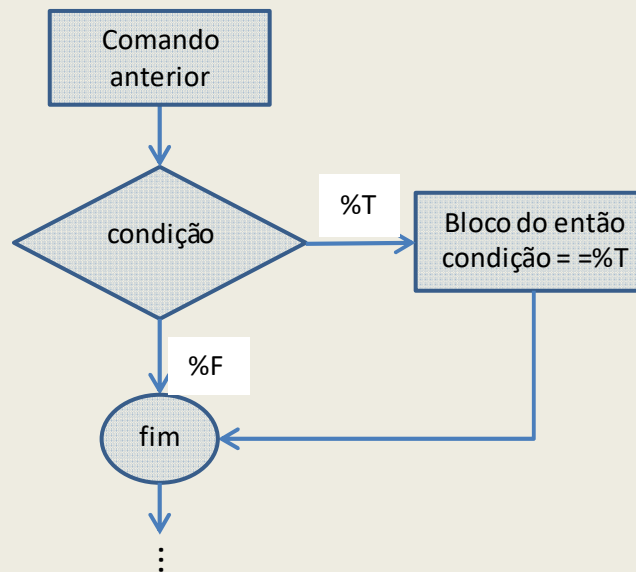
Comando de Desvio de Fluxo

Comando de Desvio de Fluxo

O comando `if` é um comando de desvio do fluxo de execução.

Duas estruturas para o comando `if` são:

`if` <condição> `then` <comandos> `end`

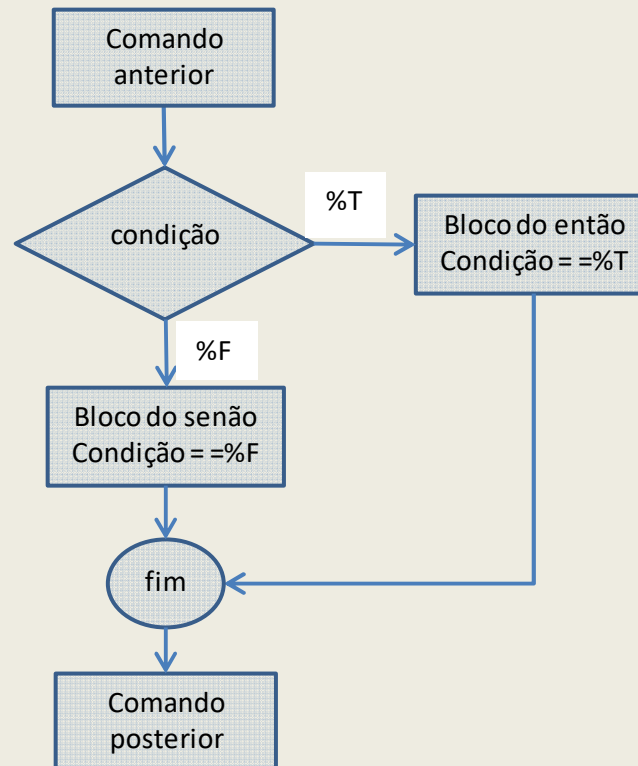


Se a condição for verdadeira, executa o Bloco do então e vai para o primeiro comando após o fim (end).

Se a condição for falsa, não faz nada e segue para o primeiro comando após o fim (end)

Comando de Desvio de Fluxo

if <condição> then <Bloco1> else <Bloco2> end



Se a condição for verdadeira então Executa o **Bloco do então** e vai para o comando posterior

Caso contrário, **se a condição for falsa**, executa o **Bloco do senão** e vai para o comando posterior

Comando de Desvio de Fluxo

if's aninhados

Quando existe este tipo de disposição do comando **if**, ocorre o que denominamos de **if's aninhados**

um **end** para cada **if** correspondente

```
if <condição_a> then
```

```
<comando_1>;
```

```
• • •
```

```
<comando_n>;
```

```
else
```

```
  if <condição_b> then
```

```
    <comando_1>;
```

```
    • • •
```

```
    <comando_m>;
```

```
  else
```

```
    <comando_1>;
```

```
    • • •
```

```
    <comando_p>;
```

```
  end
```

```
end
```

Operadores Lógicos

Operadores Lógicos

- As **expressões lógicas** quando avaliadas resultam em *True/False*
- Elas podem ser combinadas através de **operadores lógicos**.
- Eles são usados para construir expressões mais complexas, que combinam condições múltiplas, resultando em *True/False*.

`<exprLógica1> <operadorLógico> <exprLógica2>`

- Ou para negar uma condição:

`<operador_de_negação> <exprLógica>`

Operadores lógicos definidos no Scilab:

Operador	Descrição
&	E (conjunção).
	OU (disjunção não exclusiva).
~	Não (negação).

- **E (Conjunção):**
 - Resulta verdadeiro se as duas expressões forem verdadeiras;
- **OU (disjunção não exclusiva):**
 - Resulta verdadeiro se pelo menos uma das duas expressões forem verdadeiras;
- **Não (negação):**
 - Inverte o valor lógico da expressão, se ela for verdadeira o resultado é falso, se ela for falsa o resultado é verdadeiro.

Operadores Lógicos

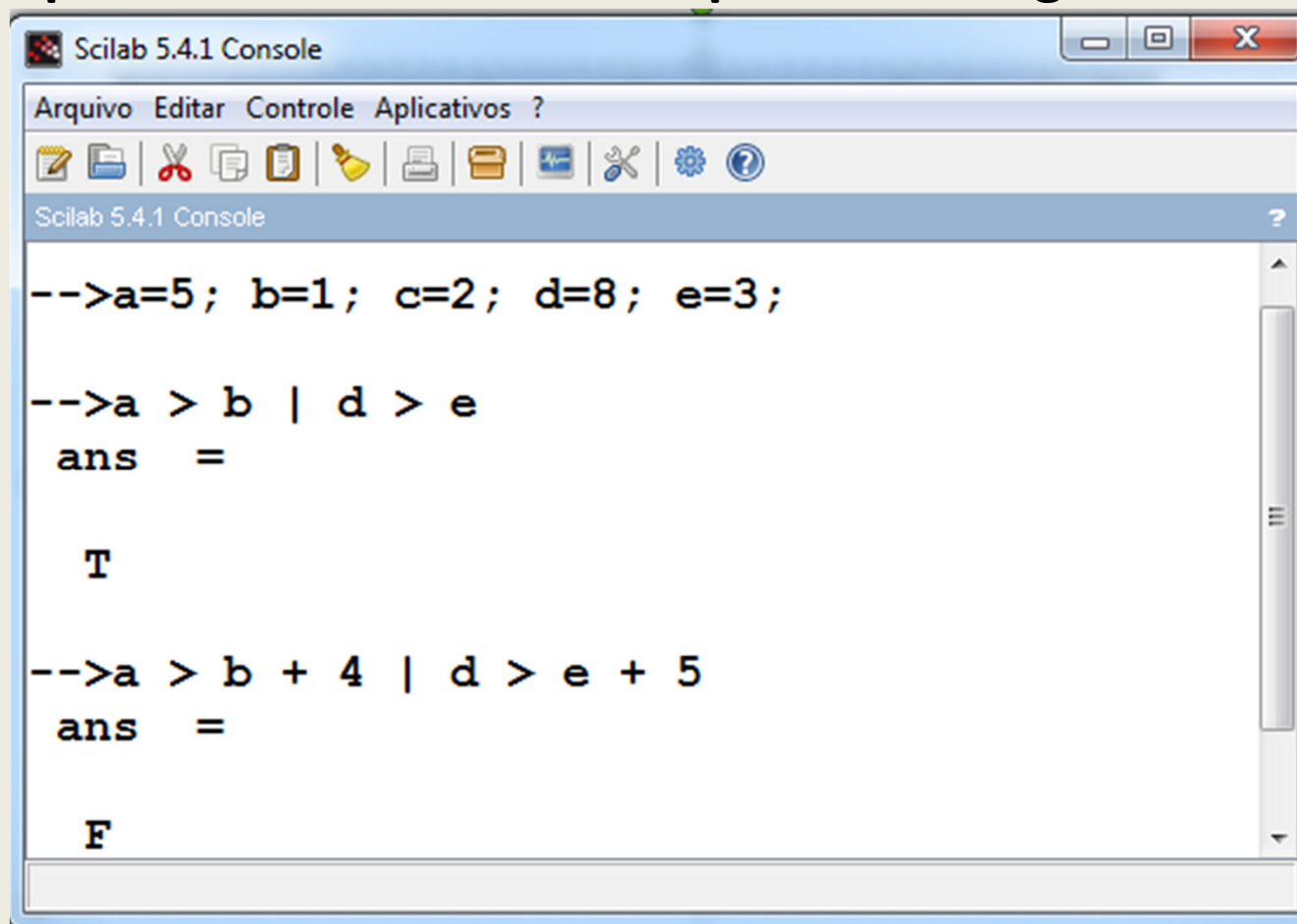
Operadores lógicos podem ser definidos por Tabelas Verdade

A	B	A & B	A B	~A
%t	%t	%t	%t	%f
%t	%f	%f	%t	%f
%f	%t	%f	%t	%t
%f	%f	%f	%f	%t

Operadores Lógicos

Usando Operadores Lógicos, Relacionais e Aritméticos

- Primeiro os operadores aritméticos e relacionais são avaliados, para depois serem avaliados os operadores lógicos.



```

Scilab 5.4.1 Console
Arquivo  Editar  Controle  Aplicativos  ?
[Icons]
Scilab 5.4.1 Console
-->a=5; b=1; c=2; d=8; e=3;

-->a > b | d > e
ans =

    T

-->a > b + 4 | d > e + 5
ans =

    F
  
```

Operadores Lógicos

Usando Operadores Lógicos, Relacionais e Aritméticos

- A ordem de precedência dos operadores lógicos é: negação (\sim), E ($\&$) e OU ($|$), ou seja $\sim X \& Y | Z$, corresponde a $((\sim X) \& Y) | Z$

```
Scilab 5.4.1 Console
-->A = 5; B = 1; C = 2; D = 8; E = 3;

-->A+3 ~= A+E & %t
ans =

F

-->sqrt(2) == sqrt(2.0000001)
ans =

F

-->C/2 + 1 < B+E & ~(%f)
ans =

T
```

Operadores Lógicos

Exemplo

- Em um parque de diversões apenas pessoas maiores que 18 anos e menores que 50 anos podem andar na montanha russa. Considerando a existência de uma variável chamada idade, a expressão lógica para permitir a entrada no brinquedo seria:
 - $(idade \geq 18) \& (idade \leq 50)$
 - **Mais uma vez, o uso dos parênteses é bom para facilitar a leitura.**
- **Atenção:** a expressão: $idade \geq 18 \& \leq 50$, está incorreta pela sintaxe do Scilab.
- A expressão $18 \leq idade \leq 50$ também está incorreta.

Exercícios

Exercício 1. Validação dos dados de entrada

Segundo uma tabela médica, o peso ideal está relacionado com a altura e o sexo de uma pessoa. Fazer um programa que receba como entradas a altura H e o sexo. A seguir verificar a validade dos dados lidos:

1. $H > 0$
2. Sexo igual à letra m ou igual à letra f

Se os dados forem válidos, calcular e informar o peso ideal dessa pessoa, utilizando as seguintes fórmulas:

para homens: $(72.7 * H) - 58$

para mulheres: $(62.1 * H) - 44.7$

Caso contrário, ou seja, se algum dado não for válido, informa:
“Dados inválidos”

Sexo = input("Digite o sexo da pessoa (m ou f): ", "s");

Exercício 2

Um terno pitagórico ou trio pitagórico, é formado por três números naturais a , b e c tais que $a^2 + b^2 = c^2$

O nome vem do teorema de Pitágoras que afirma que se as medidas dos lados de um triângulo retângulo são números inteiros, então são um terno pitagórico.

Codifique um programa que leia 3 números naturais e:

1. verifique se eles são positivos e inteiros
2. verifique se representam um terno pitagórico.

...

```
a = input("Digite o valor de a: ");
```

```
b = input("Digite o valor de b: ");
```

```
c = input("Digite o valor de c: ");
```

...

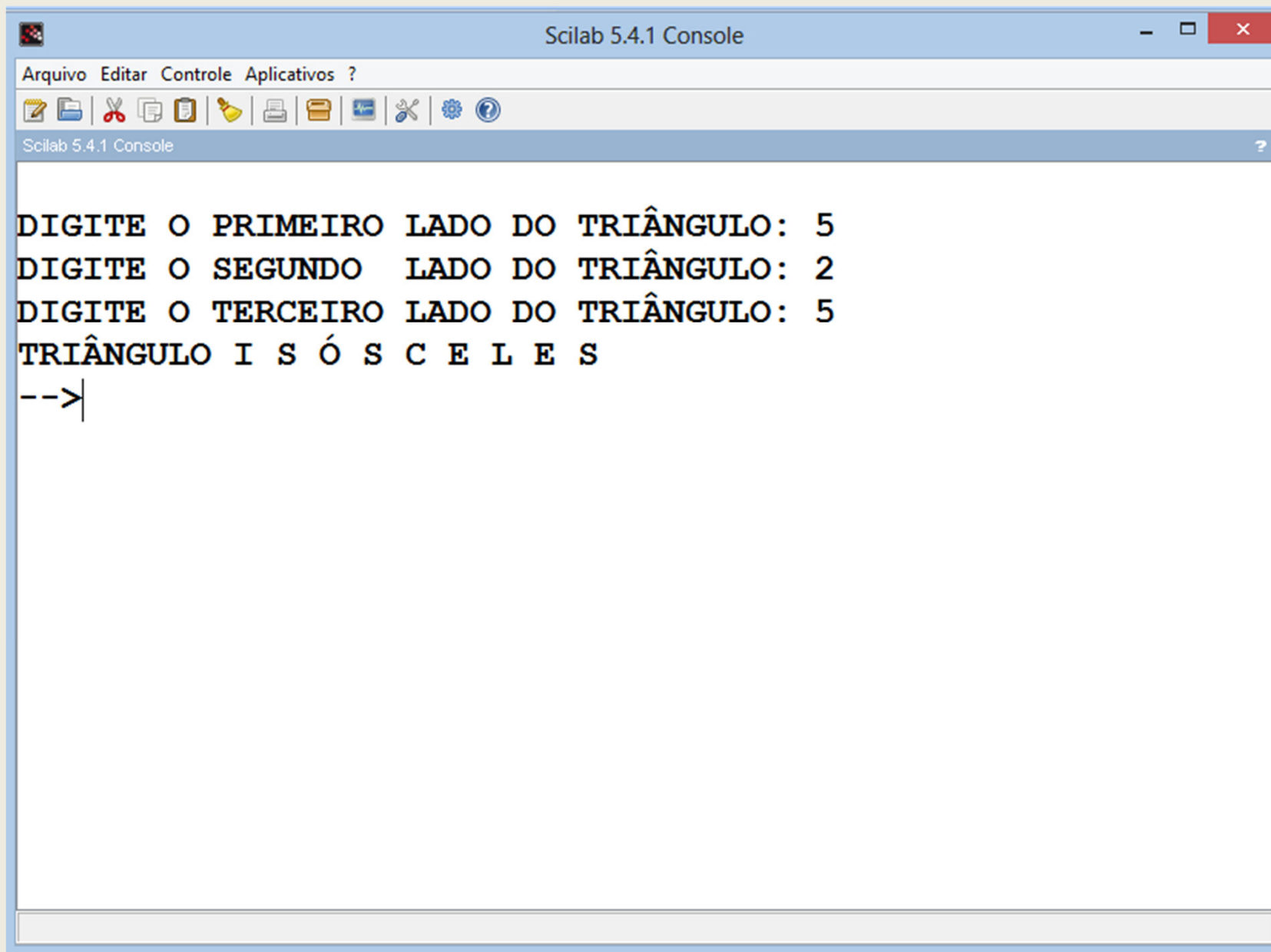
Exercício 3

A) Codifique um programa em Scilab que leia 3 valores inteiros a partir do teclado e imprimir “triângulo existente” ou “o triângulo não existe”.

B) Melhorar o E1, classificando o triângulo de acordo com seus lados: equilátero, ou isósceles, ou escaleno.

A seguir uma ilustração da execução do programa.

Exercício 3 - Execução



Scilab 5.4.1 Console

Arquivo Editar Controle Aplicativos ?

Scilab 5.4.1 Console

```
DIGITE O PRIMEIRO LADO DO TRIÂNGULO: 5  
DIGITE O SEGUNDO LADO DO TRIÂNGULO: 2  
DIGITE O TERCEIRO LADO DO TRIÂNGULO: 5  
TRIÂNGULO I S Ó S C E L E S  
-->
```

```
clc; clear;
// Entrada dos lados do triângulo
a = input("DIGITE O PRIMEIRO LADO DO TRIÂNGULO: ")
b = input("DIGITE O SEGUNDO LADO DO TRIÂNGULO: ")
c = input("DIGITE O TERCEIRO LADO DO TRIÂNGULO: ") //
Início dos cálculos
if (a == b) & (b == c) then
    printf("TRIÂNGULO E Q U I L Á T E R O")
else
    if (a == b) | (a == c) | (b == c) then
        printf("TRIÂNGULO I S Ó S C E L E S")
    else
        printf("TRIÂNGULO E S C A L E N O")
    end
end
end
```

Exercício 4

Escreva um programa que leia um valor real positivo e calcule o valor de $f(x)$ dado por:

$$\begin{aligned} F(x) &= (x^2 + \frac{1}{2})^3 \text{ se } x \leq 2 \\ &= 1/(x^2 - 4) \text{ se } 2 < x \leq 10 \text{ e} \\ &= 2\text{sen}(x) + \text{cos}(4x) \text{ se } x > 10 \end{aligned}$$

$F(x)$ não está definida para $x \leq 0$.

```
clc; clear;
x = input("DIGITE UM VALOR PARA x: ");
if x <= 0 then
    printf("\nERRO: O VALOR DE x ");
    printf("NÃO PERTENCE AOS REAIS POSITIVOS\n");
else
    if x <= 2 then
        fx = (x^2 + 0.5)^3;
    else // x já é maior que 2
        if x <= 10 then // x está entre 2 e 10
            fx = 1 / (x^2 - 4);
        else // x é maior que 10
            fx = 2 * sin(x) + cos(4*x);
        end
    end
end
printf("\n-----");
printf("\nIMPRESSÃO DO RESULTADO:")
printf("\n-----");
printf("\nfx(%4.1f) = %7.2f\n", x, fx);
end
```

Exercício 5

De acordo com o site Wikipedia:

- **São bissextos todos os anos múltiplos de 400, p. ex.: 1600, 2000, 2400, 2800...**
- **São bissextos todos os múltiplos de 4 e que não são múltiplos de 100, p. ex.: 1996, 2004, 2008, 2012, 2016...**
- **Não são bissextos todos os demais anos.**

Codifique um programa que tenha como entrada um determinado ano.

O programa imprime uma mensagem dizendo se o ano é ou não bissexto.

Exercício 5 - Código

```
clc; clear;
ano = input("DIGITE O ANO: ");
if (modulo(ano, 400) == 0) then
    printf("O ANO %g É BISSEXTO\n", ano);
else
    if (modulo(ano, 4) == 0) & ~(modulo(ano, 100) == 0)
        printf("O ANO %g É BISSEXTO\n", ano);
    else
        printf("O ANO %g NÃO É BISSEXTO\n", ano);
    end
end
end
```