



Aula 05: Condicionais (if / else)

Introdução a Programação

Túlio Toffolo & Puca Huachi
<http://www.toffolo.com.br>

Aula Anterior

- Breve Revisão
- Bibliotecas `<math.h>`
- Comandos de Decisão
- Exercícios

Aula de Hoje

- 1 Exercícios da última aula
- 2 Breve revisão
- 3 Exemplos (if)
- 4 Condicionais (if - else)
- 5 Exemplos (if - else)
- 6 Próximas Aulas
- 7 Exercícios

Aula de Hoje

- 1 Exercícios da última aula
- 2 Breve revisão
- 3 Exemplos (if)
- 4 Condicionais (if - else)
- 5 Exemplos (if - else)
- 6 Próximas Aulas
- 7 Exercícios

Exercícios da última aula

Exercício 1: Maior número

Faça um programa que leia dois números inteiros e verifique qual deles é maior.

- Imprima uma mensagem informando qual deles é o maior.
- Exemplo de saída caso o usuário digite 10 e 20:

```
1 20 é o maior número
```

```
1 // Exercício 1
2 #include <stdio.h>
3
4 int main()
5 {
6     int numero1, numero2;
7     printf("Digite dois numeros: ");
8     scanf("%d %d", &numero1, &numero2);
9
10    if (numero1 > numero2)
11        printf("O maior numero é %d\n", numero1);
12    else if (numero1 < numero2)
13        printf("O maior numero é %d\n", numero2);
14    else
15        printf("Os números são iguais!\n");
16
17    return 0;
18 }
```

Aula de Hoje

- 1 Exercícios da última aula
- 2 Breve revisão**
- 3 Exemplos (if)
- 4 Condicionais (if - else)
- 5 Exemplos (if - else)
- 6 Próximas Aulas
- 7 Exercícios

Operadores

Operadores **aritméticos**:

- +, -, *, /, %

Operadores relacionais e de igualdade:

- <, <=, >, >=
- ==, !=

Operadores lógicos:

- !, &&, ||

Operadores

Operadores **aritméticos**:

- +, -, *, /, %

Operadores **relacionais e de igualdade**:

- <, <=, >, >=
- ==, !=

Operadores **lógicos**:

- !, &&, ||

Operadores

Operadores **aritméticos**:

- +, -, *, /, %

Operadores **relacionais e de igualdade**:

- <, <=, >, >=
- ==, !=

Operadores **lógicos**:

- !, &&, ||

Operadores – Precedência

Operator	Priority	Associativity
!, ~	highest	Right
/, *, %		Left
+, -		Left
<, >, <=, >=		Left
==, !=		Left
&		Left
^		Left
		Left
&&		Left
		Left
==, +=, *=, ...	lowest	Right

Tomada de decisão

- Comando `if`
 - consiste de uma palavra-chave `if` seguida de uma expressão de teste entre parênteses. A instrução será executada apenas se a expressão de teste for **verdadeira**.
 - O corpo de um comando `if` pode conter uma única instrução terminada por ponto-e-vírgula ou várias instruções entre chaves.

Tomada de decisão

- Comando `if`
 - consiste de uma palavra-chave `if` seguida de uma expressão de teste entre parênteses. A instrução será executada apenas se a expressão de teste for verdadeira.
 - O corpo de um comando `if` pode conter uma única instrução terminada por ponto-e-vírgula ou várias instruções entre chaves.

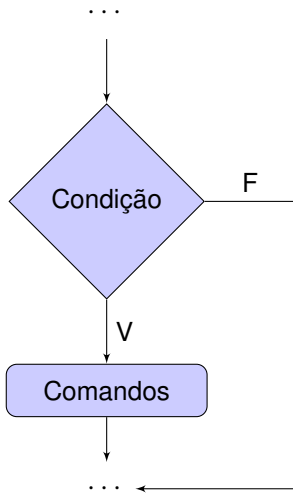
Tomada de decisão

- Comando `if`
 - consiste de uma palavra-chave `if` seguida de uma expressão de teste entre parênteses. A instrução será executada apenas se a expressão de teste for verdadeira.
 - O corpo de um comando `if` pode conter **uma única instrução terminada por ponto-e-vírgula** ou várias instruções entre chaves.

Tomada de decisão

- Comando `if`
 - consiste de uma palavra-chave `if` seguida de uma expressão de teste entre parênteses. A instrução será executada apenas se a expressão de teste for verdadeira.
 - O corpo de um comando `if` pode conter uma única instrução terminada por ponto-e-vírgula ou **várias instruções entre chaves**.

Tomada de decisão



Sintaxe do comando `if`

```
1 if ( <expressão_de_teste> )  
2     instrução_única;
```

OU

```
1 if ( <expressão_de_teste> )  
2 {  
3     instrução1;  
4     instrução2;  
5     instrução3;  
6     ...  
7 }
```

Sintaxe do comando `if`

```
1 if ( <expressão_de_teste> )  
2     instrução_única;
```

OU

```
1 if ( <expressão_de_teste> )  
2 {  
3     instrução1;  
4     instrução2;  
5     instrução3;  
6     ...  
7 }
```

Aula de Hoje

- 1 Exercícios da última aula
- 2 Breve revisão
- 3 Exemplos (if)**
- 4 Condicionais (if - else)
- 5 Exemplos (if - else)
- 6 Próximas Aulas
- 7 Exercícios

Exemplo usando o comando `if`

Exemplo 1 (resolvido anteriormente):

Faça um programa em C/C++, para calcular a área de um círculo. A área de um círculo é dada pela seguinte fórmula $a = \pi r^2$. O valor do raio r será digitado pelo usuário.

- Altere o programa anterior para calcular a área somente se o valor do raio for positivo.

Exemplo usando o comando `if`

Exemplo 1 (resolvido anteriormente):

Faça um programa em C/C++, para calcular a área de um círculo. A área de um círculo é dada pela seguinte fórmula $a = \pi r^2$. O valor do raio r será digitado pelo usuário.

- Altere o programa anterior para calcular a área somente se o valor do raio for positivo.

```
1  /* Programa que calcula a área de um círculo
2  */
3
4  #include <stdio.h>
5
6  int main()
7  {
8      // declaração da constante Pi
9      const double PI = 3.141592;
10     double raio;
11
12     printf("Digite o raio do círculo: ");
13     scanf("%lf", &raio);
14
15     // calculando e imprimindo a área do círculo
16     if (raio >= 0) {
17         double area = PI * raio * raio;
18         printf("\nÁrea do círculo: %lf\n", area);
19     }
20
21     return 0;
22 }
```

Exemplo 2

Codifique um programa que lê um número inteiro. A seguir o programa deve imprimir uma mensagem para o usuário dizendo se o número digitado é par. Se o número não for par, o programa não deve fazer nada.

```
1  /* Programa que verifica se um número é par
2     */
3  #include <stdio.h>
4
5  int main()
6  {
7     int numero; // variável para armazenar o número
8
9     printf("Digite um numero inteiro: ");
10    scanf("%d", &numero);
11
12    // Testa se o número é par
13    if (numero % 2 == 0) {
14        printf("O número %d é par!\n", numero);
15    }
16    return 0;
17 }
```


Exemplo 3

Faça um programa em C/C++, para calcular a área de um círculo. A área de um círculo é dada pela seguinte fórmula $a = \pi r^2$. O valor do raio r será digitado pelo usuário.

Verifique se o raio é positivo antes de efetuar cálculo, caso contrário imprima uma mensagem de erro ao usuário.

```
1  /* Programa que calcula a área de um círculo
2  */
3
4  #include <stdio.h>
5
6  int main()
7  {
8      // declaração da constante Pi
9      const double PI = 3.141592;
10     double raio;
11
12     printf("Digite o raio do círculo: ");
13     scanf("%lf", &raio);
14
15     // calculando e imprimindo a área do círculo
16     if (raio >= 0) {
17         double area = PI * raio * raio;
18         printf("\nÁrea do círculo: %lf\n", area);
19     }
20     // caso o raio seja negativo, imprime mensagem de erro
21     if (raio < 0)
22         printf("Erro: valor do raio é negativo...\n");
23
24     return 0;
25 }
```

Exemplo 4

Codifique um programa que lê um número inteiro positivo. A seguir o programa deve imprimir uma mensagem para o usuário dizendo se o número digitado é **par ou impar**.

```
1 // Programa que verifica se um no. é par ou impar
2 #include <stdio.h>
3
4 int main()
5 {
6     int numero; //variável para armazenar o número
7     printf("Digite um numero inteiro: ");
8     scanf("%d", &numero);
9
10    // testa se o número é par
11    if (numero % 2 == 0) {
12        printf("\n0 número %d é par.\n", numero);
13    }
14
15    // testa se o número é impar
16    if (numero % 2 != 0) {
17        printf("\n0 número %d é impar\n.", numero);
18    }
19
20    return 0;
21 }
```

Exemplo de execução

- Execução 1:

```
Digite um numero inteiro: 5
```

```
O número 5 é ímpar
```

- Execução 2:

```
Digite um numero inteiro: 8
```

```
O número 8 é par
```

Aula de Hoje

- 1 Exercícios da última aula
- 2 Breve revisão
- 3 Exemplos (if)
- 4 Condicionais (if - else)**
- 5 Exemplos (if - else)
- 6 Próximas Aulas
- 7 Exercícios

O Comando `if`

- Nos exercícios anteriores, o programa teve que tomar decisões baseadas nos dados fornecidos.
- Para tal, utilizamos dois comandos `if`.
- No entanto, podemos (e devemos) simplificar os programas com o uso do comando `if-else`.

O Comando `if-else`

- O comando `if` permite executar um ou mais comandos se a expressão relacional resultar em **verdadeiro**. Se desejarmos que algo seja executado se a expressão relacional resultar em **falso**, então devemos utilizar o comando `if-else`.
- Dessa forma, podemos executar um ou mais comandos se o teste for **verdadeiro**; ou executar um ou mais comandos distintos se o teste for **falso**.
- O comando `if-else` consiste no comando `if` seguido de uma instrução (ou um bloco de instruções), seguido da palavra-chave `else`, que por sua vez é seguida de uma instrução (ou um bloco de instruções).

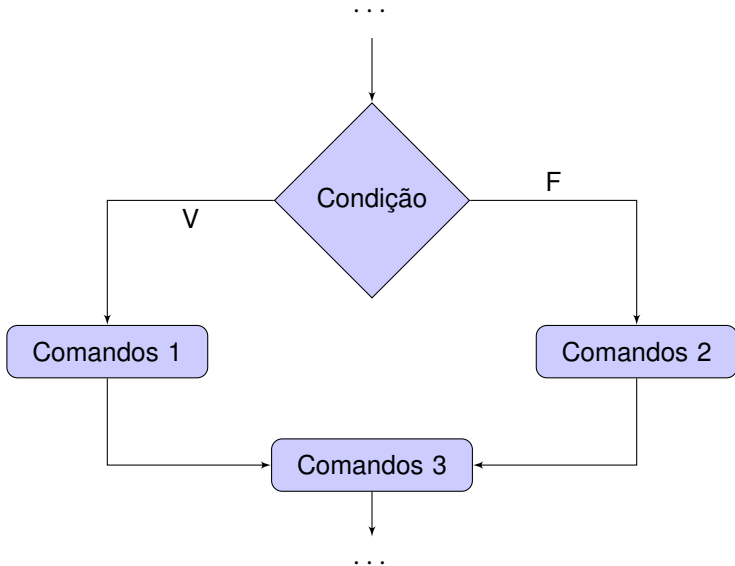
O Comando `if-else`

- O comando `if` permite executar um ou mais comandos se a expressão relacional resultar em **verdadeiro**. Se desejarmos que algo seja executado se a expressão relacional resultar em **falso**, então devemos utilizar o comando `if-else`.
- Dessa forma, podemos executar um ou mais comandos se o teste for **verdadeiro**; ou executar um ou mais comandos distintos se o teste for **falso**.
- O comando `if-else` consiste no comando `if` seguido de uma instrução (ou um bloco de instruções), seguido da palavra-chave `else`, que por sua vez é seguida de uma instrução (ou um bloco de instruções).

O Comando `if-else`

- O comando `if` permite executar um ou mais comandos se a expressão relacional resultar em **verdadeiro**. Se desejarmos que algo seja executado se a expressão relacional resultar em **falso**, então devemos utilizar o comando `if-else`.
- Dessa forma, podemos executar um ou mais comandos se o teste for **verdadeiro**; ou executar um ou mais comandos distintos se o teste for **falso**.
- O comando `if-else` consiste no comando `if` seguido de uma instrução (ou um bloco de instruções), seguido da palavra-chave `else`, que por sua vez é seguida de uma instrução (ou um bloco de instruções).

Tomada de decisão



Sintaxe do comando if-else

```
1 if ( <expressão_de_teste> )
2     instrução_única_V;
3 else
4     instrução_única_F;
```

OU

```
1 if ( <expressão_de_teste> )
2 {
3     instrução_V1;
4     ...
5     instrução_Vn;
6 }
7 else
8 {
9     instrução_F1;
10    ...
11    instrução_Fn;
12 }
```

Sintaxe do comando if-else

```
1  if ( <expressão_de_teste> )
2      instrução_única_V;
3  else
4      instrução_única_F;
```

OU

```
1  if ( <expressão_de_teste> )
2  {
3      instrução_V1;
4      ...
5      instrução_Vn;
6  }
7  else
8  {
9      instrução_F1;
10     ...
11     instrução_Fn;
12 }
```

Aula de Hoje

- 1 Exercícios da última aula
- 2 Breve revisão
- 3 Exemplos (if)
- 4 Condicionais (if - else)
- 5 Exemplos (if - else)**
- 6 Próximas Aulas
- 7 Exercícios

Exemplo 1

- Faça um programa em C/C++, para calcular a área de um círculo. A área de um círculo é dada pela seguinte fórmula $a = \pi r^2$. O valor do raio r será digitado pelo usuário.

Verifique se o raio é positivo antes de efetuar cálculo, caso contrário imprima uma mensagem de erro ao usuário. **Utilize if-else.**

```
1 #include <stdio.h>
2
3 int main()
4 {
5     // declaração da constante Pi
6     const double PI = 3.141592;
7     double raio;
8
9     printf("Digite o raio do círculo: ");
10    scanf("%lf", &raio);
11
12    // se raio for positivo, calcula e imprime a área do círculo
13    if (raio >= 0) {
14        double area = PI * raio * raio;
15        printf("\nÁrea do círculo: %lf\n", area);
16    }
17    // caso contrário, imprime mensagem de erro
18    else {
19        printf("Erro: valor do raio é negativo...\n");
20    }
21
22    return 0;
23 }
```


Exemplo 2

- Codifique um programa que leia um número inteiro positivo. A seguir o programa imprime uma mensagem para o usuário dizendo se o número digitado é par ou impar. Utilize `if-else`.

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int numero; //variável para armazenar o número
6
7     printf("Digite um numero inteiro: ");
8     scanf("%d", &numero);
9
10    // se o número for par...
11    if (numero % 2 == 0)
12        printf("\n0 número %d é par.\n", numero);
13
14    // caso contrário
15    else
16        printf("\n0 número %d é ímpar.\n", numero);
17
18    return 0;
19 }
```

Exemplo 3

- Escreva um programa para encontrar as raízes de uma equação do segundo grau:

$$ax^2 + bx + c = 0$$

- Os coeficientes da equação são reais. O programa faz a alocação de 3 posições de memória para esses coeficientes, inicializando-os com o valor zero.
- O programa efetua a leitura dos coeficientes através do teclado e, em seguida, calcula o valor das raízes existentes.
- Caso não existam raízes reais, o programa deve informar este fato ao usuário.

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     double a, b, c, delta;
7
8     printf("\nCoeficiente a: "); scanf("%lf", &a);
9     printf("\nCoeficiente b: "); scanf("%lf", &b);
10    printf("\nCoeficiente c: "); scanf("%lf", &c);
11    delta = pow(b, 2) - 4 * a * c;
12
13    if (delta < 0) {
14        printf("\nNão existem raízes reais\n");
15    }
16    else {
17        double x1 = (-b + sqrt(delta)) / (2 * a);
18        double x2 = (-b - sqrt(delta)) / (2 * a);
19        printf("\nX1 = %lf\n", x1);
20        printf("X2 = %lf\n", x2);
21    }
22    return 0;
23 }
```

Exemplo 4

- Como melhorar o programa anterior para informar o usuário as situações a seguir?
 - Não existem raízes reais ($\Delta < 0$);
 - Existem raízes reais idênticas ($\Delta = 0$);
 - Existem raízes reais distintas ($\Delta > 0$);

```
1 int main()
2 {
3     double a, b, c, delta;
4
5     printf("\nCoeficiente a: "); scanf("%lf", &a);
6     printf("\nCoeficiente b: "); scanf("%lf", &b);
7     printf("\nCoeficiente c: "); scanf("%lf", &c);
8     delta = pow(b, 2) - 4 * a * c;
9
10    if (delta < 0) {
11        printf("\nNão existem raízes reais\n");
12    }
13    else {
14        if (delta == 0) {
15            double x = (-b + sqrt(delta)) / (2 * a);
16            printf("\nRaízes idênticas: X1 = X2 = %lf\n", x);
17        }
18        else {
19            double x1 = (-b + sqrt(delta)) / (2 * a);
20            double x2 = (-b - sqrt(delta)) / (2 * a);
21            printf("\nRaízes distintas: X1 = %lf, X2 = %lf\n", x1, x2);
22        }
23    }
24    return 0;
25 }
```

```
1 int main()
2 {
3     double a, b, c, delta;
4
5     printf("\nCoeficiente a: "); scanf("%lf", &a);
6     printf("\nCoeficiente b: "); scanf("%lf", &b);
7     printf("\nCoeficiente c: "); scanf("%lf", &c);
8     delta = pow(b, 2) - 4 * a * c;
9
10    if (delta < 0) {
11        printf("\nNão existem raízes reais\n");
12    }
13    else if (delta == 0) {
14        double x = (-b + sqrt(delta)) / (2 * a);
15        printf("\nRaízes idênticas: X1 = X2 = %lf\n", x);
16    }
17    else {
18        double x1 = (-b + sqrt(delta)) / (2 * a);
19        double x2 = (-b - sqrt(delta)) / (2 * a);
20        printf("\nRaízes distintas: X1 = %lf, X2 = %lf\n", x1, x2);
21    }
22    return 0;
23 }
```

Aula de Hoje

- 1 Exercícios da última aula
- 2 Breve revisão
- 3 Exemplos (if)
- 4 Condicionais (if - else)
- 5 Exemplos (if - else)
- 6 Próximas Aulas**
- 7 Exercícios

Próxima Aula

Aula prática

- Comandos condicionais `if - else`

Aula teórica

- Comandos de decisão aninhados
- Macros e constantes
- Exercícios

Aula de Hoje

- 1 Exercícios da última aula
- 2 Breve revisão
- 3 Exemplos (if)
- 4 Condicionais (if - else)
- 5 Exemplos (if - else)
- 6 Próximas Aulas
- 7 Exercícios**

Exercícios

Exercício 1

- Codifique um programa que faça a leitura de dois número reais. A seguir o programa lê um caractere, que deve ser `+`, `-`, `*` ou `/`, e realiza a operação indicada pelo caractere sobre os valores lidos.
- O programa deve imprimir os valores e o resultado da operação realizada sobre eles, como mostra o exemplo a seguir (usando exatamente uma casa decimal):

```
1 Primeiro valor: 4.5
2 Segundo valor: 5
3 Operador: +
4
5 Resultado: 4.5 + 5.0 = 9.5
```



Perguntas?