

# **Expressões Lógicas e Comandos de Decisão IF/ELSE**

# Expressões Lógicas

São expressões que resultam em valores lógicos : **verdadeiro** ou **falso**.

Em **C**,

- **verdadeiro** é **1** e
- **falso** é **0**.

# Expressões Lógicas

Para compor as expressões lógicas utiliza-se os operadores:

- **relacionais:** comparam dois valores.

$>$ ,  $<$ ,  $>=$ ,  $<=$ ,  $!=$ ,  $==$

- **lógicos:** combinam expressões lógicas.

$\&\&$ ,  $\|\|$ ,  $!$

(e) (ou) (não)

# Expressões Lógicas

## Operadores

- **relacionais**

> maior que  
< menor que  
>= maior ou igual  
<= menor ou igual  
!= diferente  
== igual

Exemplos:

3>2    1>3    'a'>'d'  
3<3    5<10  
3>=3    3>=7  
4<=16    'A'<='A'  
4!=5    'a' != 'A'  
5==5    'c' == 'C'

**Obs: Strings não são comparados através de operadores relacionais.**

# Expressões Lógicas

Um erro muito freqüente em C é confundir a **atribuição**, representada por `=` com a **comparação**, representada por `==`. Exemplo: Supondo `x=2` e `y=3`:

- `x = y` : o valor **3** é atribuído à variável `x`;
- `x == y` : 2 é comparado a 3 resultando **0(F)**.

**Portanto, `x = y` é diferente de `x==y`!!**

# Expressões Lógicas

## Operadores

- **Lógicos**

!	NOT lógico	(não)
&&	AND lógico	(e)
	OR lógico	(ou)

# Expressões Lógicas

Considerando que A e B são duas expressões lógicas.

## Tabela Verdade

<b>A</b>	<b>B</b>	<b>A &amp;&amp; B</b>	<b>A    B</b>	<b>!A</b>
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

# Expressões Lógicas

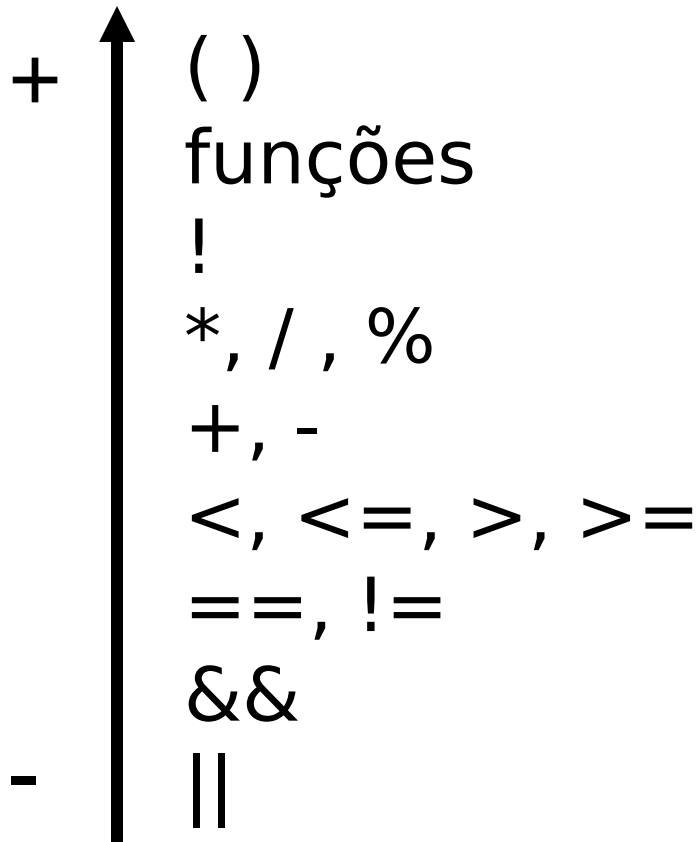
## Operadores Lógicos

- `!A` : inverte o valor lógico de A
- `A && B` : para resultar 1 (V) exige que as expressões A e B sejam verdadeiras ao mesmo tempo.
- `||` : para resultar 0 (F) exige que as expressões A e B sejam falsas ao mesmo tempo.



# Expressões Lógicas

## Prioridade dos Operadores



# Expressões Lógicas

Exemplos:

```
int num=2, val=3;
```

```
char op = 'C';
```

a) `num < val + 1 && num <= pow(val, 2) - 7`

b) `op < 'E' || op <= 'e' && op != 'c'`

c) `num > 1 && num < sqrt(64) && !(num != 2)`

d) `op == 'C' || op == 'c'`

# Decisão

- Permite criar alternativas para o fluxo de execução de um programa
- C prevê 2 tipos de comandos de decisão:
  - **if**
  - **switch**

# Comando if

A forma do comando ***if*** é a seguinte:

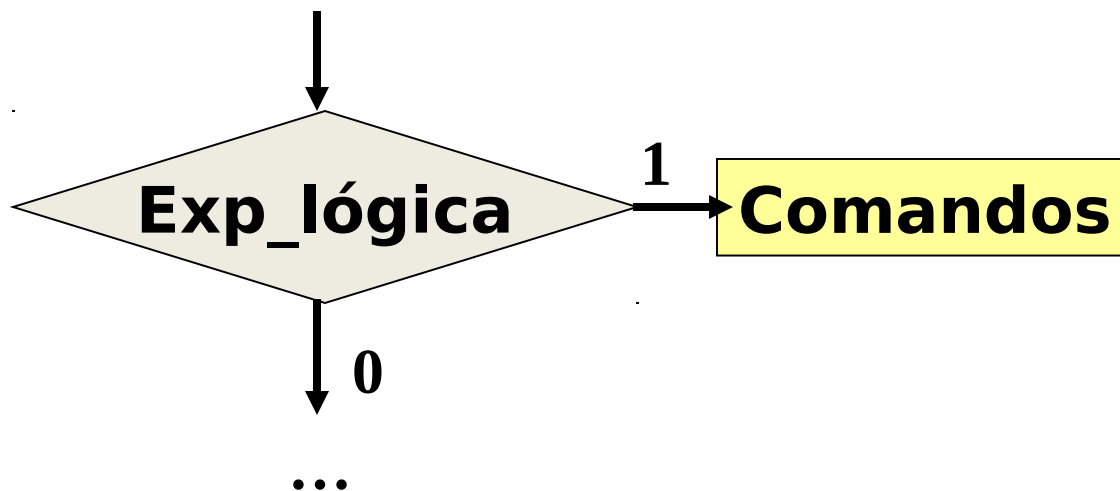
***if*** (*expressão\_lógica*) *comando*;

*ou*

***if*** (*expressão\_lógica*) {

*comandos*;

}



# Comando if

```
#include <stdio.h>
main ()
{
    int valor=10;

    printf("\nDigite um valor de 1 a 20");
    scanf("%d",&valor);
    if(valor==10) printf("\nAcertou!! ");
    if(valor!=10) printf("\nErrou!! ");
}
```

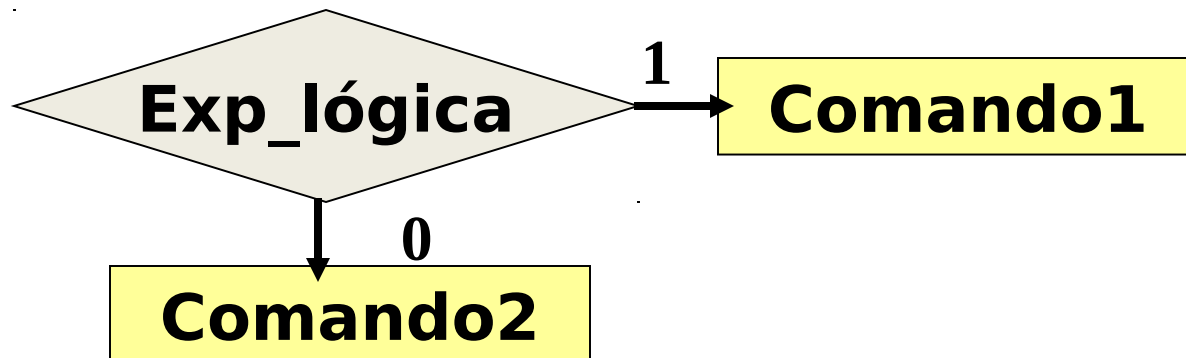
# Comando if

```
#include <stdio.h>
main ()
{
    int valor=10;
    printf("\nDigite um valor de 1 a 20");
    scanf("%d",&valor);
    if(valor==10) printf("\nAcertou!! ");
    if(valor!=10) {
        printf("\nVocê Errou!! ");
        if(valor==9 || valor ==11)
            printf("Chutou bem! Está quente!\n");
        if(valor <=5)
            printf("Chutou mal! Está frio!\n");
    }
}
```

# Comando if-else

Outra forma de usar o ***if*** é com o ***else***. Esse tipo de construção permite a definição de ações alternativas.

***if (expressão) comando1;***  
***else comando2;***



# Comando if-else

```
#include <stdio.h>
main ()
{
    int valor=10;

    printf("\nDigite um valor de 1 a 20");
    scanf("%d",&valor);
    if(valor==10) printf("\nAcertou!! ");
    else printf("\nErrou!! ");
}
```



# Comando if-else

***Blocos:***

```
if (expressão) {  
    comandos;  
}  
else {  
    comandos;  
}
```

# Comando if-else

```
#include <stdio.h>
main ()
{
    int valor=10;
    printf("\nDigite um valor de 1 a 20");
    scanf("%d",&valor);
    if(valor==10) printf("\nAcertou!! ");
    else {
        printf("\nVocê Errou!! ");
        if(valor==9 || valor ==11)
            printf("Chutou bem! Está quente!\n");
        if(valor <=5)
            printf("Chutou mal! Está frio!\n");
    }
}
```

# Exemplo:

Faça um programa que leia a idade de um nadador e classifica-o em uma das seguintes categorias:

5 a 7 anos: Infantil A

8 a 10 anos : Infantil B

11 a 13 anos : Juvenil A

14 a 17 anos: Juvenil B

maiores de 17 anos: Adulto

# 1ª Solução

```
#include <stdio.h>
main ()
{ int idade;
  printf("\nDigite a idade do nadador");
  scanf("%d",&idade);
  if(idade>=5 && idade<=7) printf("\nInfantil A");
  if(idade>=8 && idade<=10) printf("\nInfantil B");
  if(idade>=11 && idade<=13) printf("\nJuvenil A");
  if(idade>=14 && idade<=17) printf("\nJuvenil B");
  if(idade>=17) printf("\nAdulto");
}
```

**Funciona para qualquer entrada de dados ?**

# 2ª Solução

```
#include <stdio.h>

main ()
{ int idade;
  printf("\nDigite a idade do nadador");
  scanf("%d",&idade);
  if(idade <0 || idade>120)
    printf("\nIdade Inválida");
  if(idade>=0 && idade <5)
    printf("\nSem categoria");
  if(idade>=5 && idade<=7) printf("\nInfantil A");
  if(idade>=8 && idade<=10) printf("\nInfantil B");
  if(idade>=11 && idade<=13) printf("\nJuvenil A");
  if(idade>=14 && idade<=17) printf("\nJuvenil B");
  if(idade>=17 && idade<=120) printf("\nAdulto");
}
```

# 3ª Solução (está certa ?)

```
#include <stdio.h>

main ()
{ int idade;
  printf("\nDigite a idade do nadador");
  scanf("%d",&idade);
  if(idade <0 || idade>120)
    printf("\nIdade Inválida");
  if(idade>=0 && idade <5)
    printf("\nSem categoria");
  if(idade>=5 && idade<=7) printf("\nInfantil A");
  if(idade>=8 && idade<=10) printf("\nInfantil B");
  if(idade>=11 && idade<=13) printf("\nJuvenil A");
  if(idade>=14 && idade<=17) printf("\nJuvenil B");
  else printf("\nAdulto");
}
```

# Comandos if-else encadeados

```
if (expressão) comando1;  
else if (expressão) comando2;  
    else if(expressão) comando3;  
        else if(expressão) comando4;  
            else ...
```

# 4ª Solução

```
#include <stdio.h>
```

```
main ()
```

```
{ int idade;
```

```
    printf("\nDigite a idade do nadador");
```

```
    scanf("%d",&idade);
```

```
    if(idade <0 || idade>120)
```

```
        printf("\nIdade Inválida");
```

```
    else if(idade <5) printf("\nSem categoria");
```

```
        else if (idade<=7) printf("\nInfantil A");
```

```
            else if(idade<=10)
```

```
                printf("\nInfantil B");
```

```
            else if(idade<=13)
```

```
                printf("\nJuvenil A");
```

```
            else if(idade<=17)
```

```
                printf("\nJuvenil B");
```

```
            else printf("\nAdulto");
```

```
}
```



# Ambigüidade

```
x=2;  
if(x<2)  
    if(x>=-3)  
        printf("\n x está entre [-3;1]");  
else printf("x é maior que 1");
```

**Problema:**

**O else se refere ao 1º if ou ao 2º ?**

# Ambigüidade

**C resolve o problema da ambigüidade associando o `else` sempre ao último `if`.**

**Portanto, para que a construção funcione como desejado, pode-se resolver o problema usando *bloco*.**

```
x=2;
```

```
if(x<2) {
```

```
    if(x>=-3)
```

```
        printf("\n x está entre [-3;1]);
```

```
}
```

```
else printf("x é maior que 1");
```

# Exercícios:

1) Faça um programa que encontra o maior de 3 valores.

2) Altere o programa anterior para que ele encontre o menor de 4 valores.

3) Faça um programa que leia 3 valores *float* (n1,n2 e n3) e um valor inteiro (*opcao*). Se *opcao* for:

- 1 : escreve os valores em ordem crescente;
- 2: escreve os valores em ordem decrescente;
- 3 : escreve o maior dentre os demais.

Prever situações de erro.