

Linguagem de Programação

- Estruturas de controle de fluxo (seleção).
 - Material: LP_Aula03



Grupo de comandos Padrão ANSI

Seleção

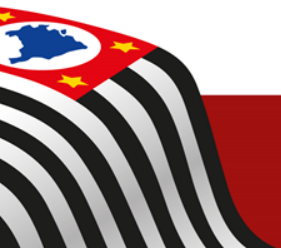
Iteração

Desvio

Rótulo

Expressão

Bloco

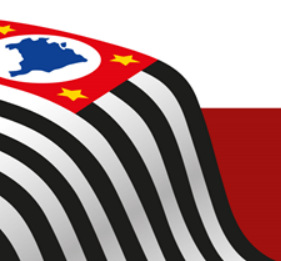


Verdadeiro e Falso em C

Em C, ao contrário de muitas linguagens, um valor é verdadeiro quando o valor for diferente de zero.

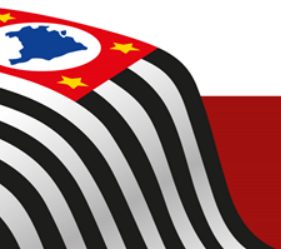
Um valor falso é \rightarrow 0.

Esse método simplifica a codificação de uma ampla gama de rotinas.



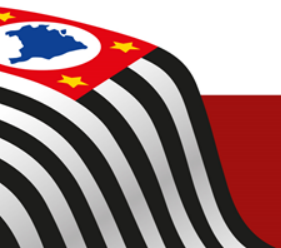
Operadores Relacionais em C

Operador	Nome	Exemplo	Significado do Exemplo
<code>==</code>	Igualdade	<code>a == b</code>	a é igual a b?
<code>></code>	Maior que	<code>a > b</code>	a é maior que b?
<code>>=</code>	Maior ou Igual que	<code>a >= b</code>	a é maior ou igual a b?
<code><</code>	Menor que	<code>a < b</code>	a é menor que b?
<code><=</code>	Menor ou Igual que	<code>a <= b</code>	a é menor ou igual a b?
<code>!=</code>	Diferente de	<code>a != b</code>	a é diferente de b?

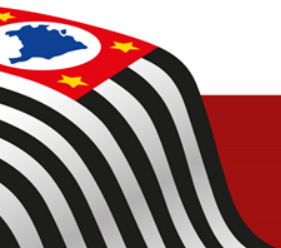


Operadores Lógicos usados em C

Operador	Significado	Exemplo
&&	AND (E lógico)	<code>x>=1 && x<=19</code>
 	OR (OU lógico)	<code>x==1 x ==2</code>
!	NOT (Negação lógica)	<code>! Continuar</code>



Comandos de seleção if e switch



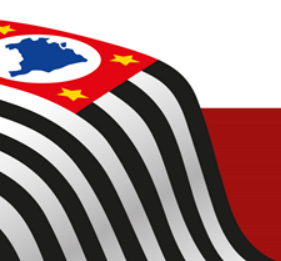
Comando if

A forma geral da sentença **if** é

```
if (expressão) comando;  
else comando;
```

Onde **comando** pode ser um único comando ou um bloco de comandos ou nada. A cláusula **else** é opcional.

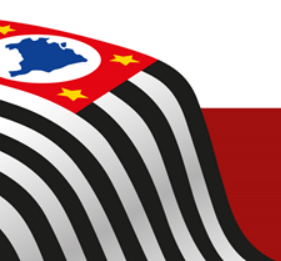
Se a expressão for diferente de **ZERO** é verdadeira o comando ou bloco do **if** é executado, senão executa-se o comando ou bloco do **else**.



Exemplo - 1

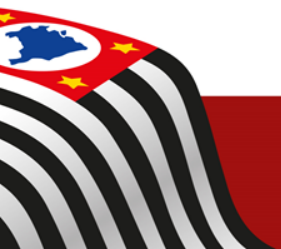
```
/* Programa de números mágicos #1. */  
#include <stdio.h>  
#include <stdlib.h>  
  
void main(void)  
{  
    int magic; /* número mágico */  
    int guess; /* palpite do usuário */  
  
    magic = rand(); /* gera o número mágico */  
  
    printf("adivinha o número mágico: ");  
    scanf("%d", &guess);  
  
    if(guess == magic) printf("*** Certo ***");  
}
```

Note a ausência do else (opcional).



Exemplo – 2 (com else)

```
/* Programa de números mágicos #2. */  
#include <stdio.h>  
#include <stdlib.h>  
  
void main(void)  
{  
    int magic; /* número mágico */  
    int guess; /* palpite do usuário */  
  
    magic = rand(); /* gera o número mágico */  
  
    printf("adivinha o número mágico: ");  
    scanf("%d", &guess);  
  
    if(guess == magic) printf("*** Certo ***");  
    else printf("Errado");  
}
```



ifs Aninhados

Um **if** aninhado é um comando **if** que é o objeto de outro **if** ou **else**.

São muito comuns em programação.

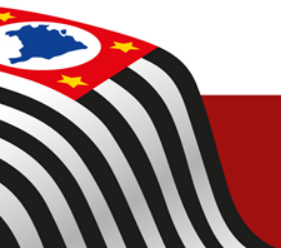
Um comando **else** sempre se refere ao comando **if** mais próximo, que está dentro do mesmo bloco de **else** e não está associado a outro **if**. Por exemplo:

```
if(i)
{
    if(j) comando 1;
    if(k) comando 2; /* este if */
    else comando 3; /* está associado a este else */
}
else comando 4; /* associado a if(i) */
```



Nota sobre níveis de aninhamento

O Padrão C ANSI especifica que pelo menos 15 níveis de aninhamentos devem ser suportados. Na prática a maioria dos compiladores permite substancialmente mais. Que tal melhorarmos o programa número mágico?



Exemplo - 3

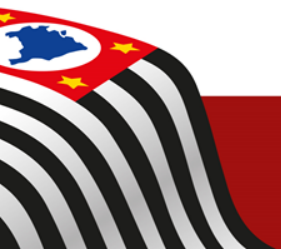
```
/* Programa de números mágicos #3. */
#include <stdio.h>
#include <stdlib.h>

void main(void)
{
    int magic; /* número mágico */
    int guess; /* palpite do usuário */

    magic = rand(); /* gera o número mágico */

    printf("Adivinhe o número mágico: ");
    scanf("%d", &guess);

    if(guess == magic) {
        printf("*** Certo ***");
        printf(" %d é o número mágico\n", magic);
    }
    else {
        printf("Errado, ");
        if(guess > magic) printf("muito alto\n");
        else printf("muito baixo\n");
    }
}
```



A Escada if-else-if

Uma construção comum em programação é a forma if-else-if. Sua forma geral:

```
if(expressão)comando;  
else  
    if(expressão)comando;  
    else  
        if(expressão)comando;  
        .  
        .  
        .  
    else comando;
```

As condições são avaliadas de cima para baixo. Assim que uma verdadeira é encontrada, o comando associado a ela é executado e desvia do resto da escada. Se nenhuma das condições for verdadeira desvia-se para o último else.



Exemplo - 4

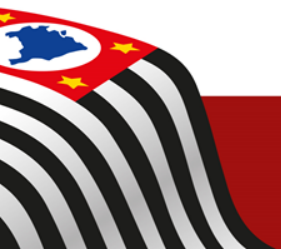
```
/* Programa de números mágicos #4. */
#include <stdio.h>
#include <stdlib.h>

void main(void)
{
    int magic; /* número mágico */
    int guess; /* palpite do usuário */

    magic = rand(); /* gera o número mágico */

    printf("Adivinhe o número mágico: ");
    scanf("%d", &guess);

    if(guess == magic) {
        printf("*** Certo ***");
        printf("%d é o número mágico", magic);
    }
    else if(guess > magic)
        printf("Errado, muito alto");
    else printf("Errado, muito baixo");
}
```



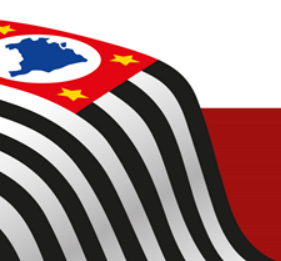
O ? (Alternativo) [Operador Ternário]

Você pode usar o operador ? Para substituir o comando if-else na sintaxe:

```
if(condição) expressão;  
else expressão;
```

- Contudo, os corpos de if e else não podem ser comandos de C. Forma geral do ?

```
Exp1 ? Exp2 : Exp3
```



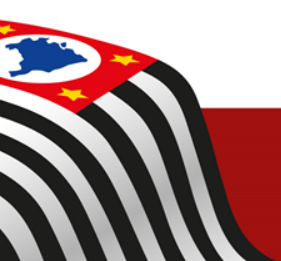
? [Exemplo]

```
x = 10;  
if (x>9) y = 100;  
else y = 200;
```

If-else

```
x = 10;  
y = x>9 ? 100 : 200;
```

Operador ?



Exemplo

O próximo programa usa o operador “?” para elevar ao quadrado uma valor inteiro digitado pelo usuário. Contudo, este programa preserva o sinal (10 ao quadrado = 100 e -10 ao quadrado = -100).



Exemplo

```
#include <stdio.h>

void main(void)
{
    int isqrd, i;

    printf("Digite um número: ");
    scanf("%d", &i);

    isqrd = i>0 ? i*i : -(i*i);

    printf("%d ao quadrado é %d", i, isqrd);
}
```



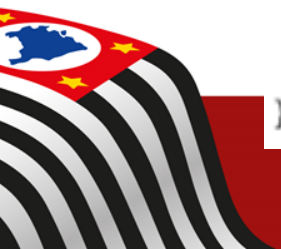
```
/* Número mágico programa #5. */
#include <stdio.h>
#include <stdlib.h>

void main(void)
{
    int magic;
    int guess;

    magic = rand(); /* gera o número mágico */

    printf("Adivinhe o número mágico: ");
    scanf("%d", &guess);

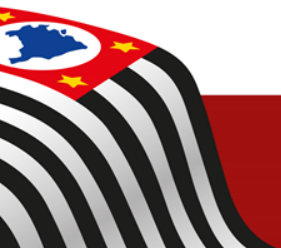
    if(guess == magic) {
        printf("*** Certo ***");
        printf("%d é o número mágico", magic);
    }
    else
        guess > magic ? printf("Alto") : printf("Baixo");
}
```



Expressão condicional

Algumas vezes, os iniciantes em C se confundem pelo fato de que você pode usar qualquer expressão válida para controlar um if ou um ?.

Isto é, você não fica restrito a expressões envolvendo operadores relacionais e lógicos (como em Pascal ou Basic).



O que acontece ao executarmos o programa abaixo?

```
/* Divide o primeiro número pelo segundo. */  
  
#include <stdio.h>  
  
void main(void)  
{  
    int a, b;  
  
    printf("Digite dois números: ");  
    scanf("%d%d", &a, &b);  
  
    if(b) printf("%d\n", a/b);  
    else printf("não pode dividir por zero\n");  
}
```

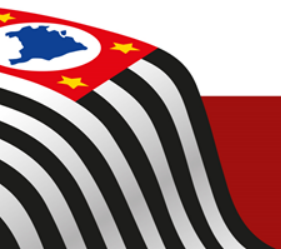


Comando switch

Testa sucessivamente o valor de uma expressão contra uma lista de constantes inteiras ou caractere. Quando o valor coincide, os comandos associados a àquela constante são executados.

Forma geral do switch é:

```
switch(expressão) {  
  case constante1:  
    seqüência de comandos  
    break;  
  case constante2:  
    seqüência de comandos  
    break;  
  case constante3:  
    seqüência de comandos  
    break;  
  .  
  .  
  .  
  default:  
    seqüência de comandos  
}
```

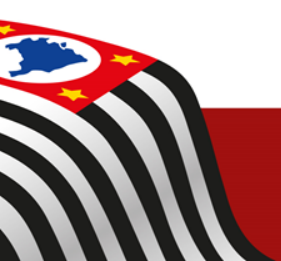


Comando switch

O **switch** testa a variável e executa a declaração cujo **case** corresponda ao valor atual da variável.

A declaração **default** é opcional e será executada apenas se a variável, que está sendo testada, não for igual a nenhuma das constantes.

O comando **break**, faz com que o **switch** seja interrompido assim que uma das declarações seja executada.



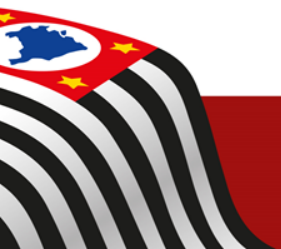
Exemplo

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
    int num;
    printf ("Digite um numero: ");
    scanf ("%d",&num);
    switch (num)
    {
        case 9:
            printf ("\n\nO numero eh igual a 9.\n");
            break;
        case 10:
            printf ("\n\nO numero eh igual a 10.\n");
            break;
        case 11:
            printf ("\n\nO numero eh igual a 11.\n");
            break;
        default:
            printf ("\n\nO numero nao eh nem 9 nem 10 nem 11.\n");
    }
    system("PAUSE");
    return 0;
}
```

Bloco de Comandos

Você pode criar um bloco de código como um dos comandos da sequência.

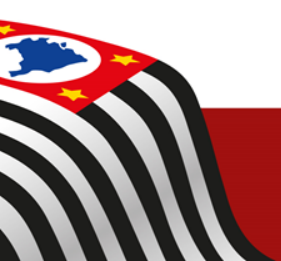
```
/* Isto é correto. */  
switch (c) {  
    case 1:  
        { /* Cria bloco */  
            int t;  
  
            .  
            .  
            .  
        }  
  
        .  
        .  
        .  
}
```



Comandos switch aninhados

Você pode ter um switch como parte de uma sequencia de comandos de um outro switch. Mesmo se as constantes dos cases dos switches interno e externo possuírem valores comuns, não ocorrerão conflitos.

```
switch(x) {  
    case 1:  
        switch(y) {  
            case 0: printf ("erro de divisão por zero");  
                    break;  
            case 1: process(x,y);  
        }  
        break;  
    case 2:  
        .  
        .  
        .  
}
```



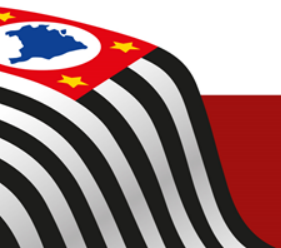
CENTRO PAULA SOUZA

Fatec

Mogi Mirim

Arthur de Azevedo

Exercícios para a prática



- 1) Escreva um programa que calcule e exiba
- o Salário Bruto,
 - o Salário Líquido e
 - o Imposto a Pagar
 - de acordo com a seguinte regra:

Salário	Taxa
<1000	5 %
≥ 1000 e < 5000	11 %
≥ 5000	35 %



2) Escreva um programa que leia um Salário Bruto e o Estado Civil de uma pessoa e calcule e mostre o Imposto a pagar obedecendo a seguinte regra: (s) solteiros pagam 10% e (c) casados pagam 9% de impostos.

3) Implemente um programa que calcule os aumentos de um ordenado para o corrente ano. Se o ordenado for > 1000 deve ser aumentado em 5%, se não deve ser aumentado em 7%. **Nota: use o operador ternário (?).**

4) Escreva um programa que calcule o imposto pago por mulheres e por homens, sabendo-se que mulheres pagam 10% de imposto sobre o salário e homens pagam mais 5% do que as mulheres. **Nota: use o switch.**



CENTRO PAULA SOUZA

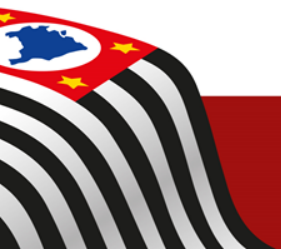
Fatec

Mogi Mirim

Arthur de Azevedo

Prof. Me. Marcos Roberto de Moraes, o Maromo

FIM



Referências Bibliográficas

DAMAS, L. M. D. Linguagem C. LTC, 2007.

HERBERT, S. C completo e total. 3a. ed. Pearson, 1997.

