



# Linguagem C - Tipos de Dados

Lógica de Programação

# Tipos de Dados

Caro(a) aluno(a),

Agora iremos conhecer os tipos de dados da Linguagem C, alguns você já conhece devido aos seus estudos em ILA.

Bom estudo!!!



# Tipos básicos

Tipo	Bits	Faixa de valores	Bytes
char	8	-128 a 127	1
Int	16	-32768 a 32767	2
Float	32	-3.4E-38 a 3.4E+38	4
Double	64	-1.7E-308 a 1.7E+308	8
void	0	Sem valor	0

A tabela exibe os cinco tipos de dados básicos que podem ser utilizados pelo programador para definir suas variáveis. São exibidos os **tipos básicos**, a quantidade de **bits**, a **faixa de valores** válida e o número de **bytes** que cada tipo de dados ocupa na memória RAM (memória principal) ou em disco (quando armazenados na memória secundária).



# Modificadores de Tipos

Modificador de tipo	Modificação	Descrição
signed	c/sinal	Números positivos e negativos
unsigned	s/sinal	Números positivos
long	longo	Aumenta o número de bytes do tipo
short	curto	Diminui o número de bytes do tipo

São utilizados para modificar os tipos de dados base, ou seja, adaptando-se às necessidades do programador. Os modificadores modificam a quantidade de bits e bytes dos tipos-base, alterando, desta forma, a faixa de valores destes novos tipos de dados.



# Operadores

Operador aritmético	Ação
-	Subtração
+	Adição
*	Multiplicação
/	Divisão
%	Resto inteiro da divisão
-- ++	Decremento / Incremento

Hierarquia	Operação
1	Parênteses
2	Funções
3	++ --
4	-
5	* / %
6	+ -



# Operadores

Operador Relacional	Ação
>	Maior que
>=	Maior ou igual a
<	Menor que
<=	Menor ou igual
==	Igual a
!=	Diferente de



# Operadores

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

Hierarquia Operadores Relacional e Lógico	Operação
1	!
2	> >= < <=
3	== !=
4	&&
5	





# Incremento e Decremento

São operadores aritméticos que permitem realizar operações de soma e subtração de forma simplificada.

- ++ adiciona (1) ao operando
- subtrai (1) ao operando

As seguintes operações são equivalentes:

```
x++; x = x + 1;  
x--; x = x - 1;
```



# Incremento e Decremento

## Exemplo:

```
x = 10;           /* y será 11 */  
y = ++x; /* x será 11 */
```

```
x = 10;           /* y será 10 */  
y = x++; /* x será 11 */
```

Os operadores (++ ou --) podem ser colocados antes ou depois do operando.

Quando precede seu operando, C efetua a **operação** de incremento ou decremento antes de utilizar o valor do operando.

Quando o operador vier depois do operando, C utiliza o **valor do operando** antes de incrementá-lo ou decrementá-lo.



# Operador de atribuição

O operador de atribuição é o símbolo de igual =. Uma atribuição pode ser escrita em uma das seguintes formas:

variável = constante;

variável = variável;

variável = expressão;

variável = função(x);

$x = 3;$

$x = y;$

$x = a + b;$

$x = \text{sqrt}(y);$



# Operador de atribuição - Exemplo

O programa calcula a idade de uma pessoa.

```
#include <stdio.h>
#include <conio.h>
void main (void)
{
int idade, ano_atual, ano_nasceu;
clrscr();
printf("Ano ATUAL: ");
scanf("%d",&ano_atual);
printf("Ano em que NASCEU: ");
scanf("%d",&ano_nasceu);
idade = ano_atual - ano_nasceu;
printf("Sua IDADE e\ ' %d\n",idade);
getch();
}
```

*/\* leitura do ano atual \*/*

*/\* leitura do ano de nascimento \*/*

*/\* atribuição - cálculo da idade \*/*



# Operador de atribuição

A linguagem de programação C permite utilizar o operador de atribuição em expressões, junto com operadores matemáticos, lógicos, relacionais, chamada de funções, e outros.

```
if ((produto = x * y) < 0)
```

**Funcionamento:** Primeiramente C atribui o valor  $x * y$  a variável **produto**, para depois avaliar a expressão, ou seja, comparar se o **produto** é menor (<) que zero.



# Operador *sizeof* (tamanho de)

Retorna o tamanho (em bytes) de uma variável ou de um tipo que está em seu operando.

**Exemplo:** O programa exibe a quantidade de bytes das variáveis e tipos.

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

void main (void)
{
  int x, y;
  clrscr();
  x = sizeof(float);
  printf("%d\n%d",x,sizeof(y));
  getch();
}
```

/\* x vale 4 \*/  
/\* sizeof(y) é 2 \*/



# Expressões

Uma expressão em C é qualquer combinação válida de operadores (aritméticos, relacionais ou lógicos), constantes, funções e variáveis.

**Exemplo:** `c = sqrt (a) + b / 3.4;`



# Conversão de tipos em expressões

Quando uma expressão é composta de tipos diferentes, C converte todos os operandos para o tipo do maior operando. Antes, os tipos abaixo são convertidos para:

**char** - convertido para **int**

**float** - convertido para **double**

## Exemplo:

```
char ch;
```

```
int i;
```

```
float f;
```

```
double d;
```

```
float result
```

