

# Tipos primitivos em C

## Introdução

Na geladeira temos recipientes específicos para produtos específicos (exemplo, copos, garrafas, bandeja de ovos, etc.). Na programação em C é necessário saber aquilo que queremos guardar e selecionar os melhores "recipientes" para a tarefa.

Estes recipientes são: **char**, **int**, **float**, **double**

## Variáveis

Uma **variável** é um nome associado a uma posição de memória. O tipo está associado ao número de bytes que serão utilizados para armazenar o valor da variável.

Os nomes de variáveis podem ter:

- Letras Maiúsculas e minúsculas, dígitos [0-9], \_
- O primeiro caractere não pode ser um dígito
- C é case sensitive
- Não pode ser uma palavra reservada
- Sem acentos ou 'ç'

Sempre que uma variável é declarada, estamos solicitando ao compilador que reserve espaço em memória para armazená-la. Esse espaço será referenciado pelo nome da variável.

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

```
int main(){
```

```
int x;  
float f;  
char c;
```

```
}
```

## Atribuição

```
int x;  
x=10;
```

Na mesma linha:

```
int x = 10;
```

## Tipo int

Armazenar valores que pertencem ao conjunto dos números inteiros. Qualquer operação entre inteiros irá retornar um número inteiro.

```
21%4=1
```

```
21/4=5
```

Exibir inteiro com *printf*:

```
In [8]: #include <stdio.h>
```

```
int main(){
    int num=21/4;
    int rest=21%4;

    printf("21/4=%d 21%%4=%d\n",num,rest);
}
```

21/4=5 21%4=1

Ler inteiro com *scanf*:

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

```
int main(){

    int num;
    printf("Introduza um número inteiro: ");
    scanf("%d",&num);
    printf("num=%d\n",num);

}
```

## Operador sizeof

Oferece o tamanho em bytes de um tipo de variável qualquer. Este operador também é útil para a alocação dinâmica com a função *malloc*. Sua utilização é feita da seguinte maneira:

```
sizeof(int);
```

```
sizeof(float);
```

```
sizeof(var);
```

## Prefixos do tipo int

Na matemática, podemos representar um número com uma quantidade arbitrária de algarismos. Porém, o computador possui memória limitada, e consequentemente possui um limite de quantos algarismos pode armazenar em uma variável do tipo *int*.

Para garantir que um programa sempre utilize determinado número de bytes para um inteiro, podemos utilizar 4 prefixos para melhor definição do tamanho do inteiro:

```
short // 2 bytes
long // 4 bytes
signed // com sinal
unsigned // sem sinal
```

O formato de leitura e escrita de *short* e *long* é %h e %l. O prefixo *signed* não é necessário, pois é o padrão. O formato de leitura e escrita de *unsigned* é %u.

Tipo da variável	Nº de bytes	Valor mínimo	Valor máximo
int	2	-32 768	32 767
short int	2	-32 768	32 767
long int	4	-2 147 483 648	2 147 483 647
unsigned int	2	0	65 535

Tipo da variável	Nº de bytes	Valor mínimo	Valor máximo
unsigned short int	2	0	65 535
unsigned long int	4	0	4 294 967 295