

Tipos de dados

emanoelim@utfpr.edu.br

Tipos de dados

- Os tipos de dados são utilizados para determinar a quantidade de memória necessária para uma variável:

```
int variavel;
```

a declaração acima determina que a variável “variavel” irá ocupar 4 bytes.

Tipos de dados

- Tipos básicos:

Tipo	Tamanho em bytes	Faixa
char	1	0 a 255
int	ao menos 2	-32.768 a 32.767
	geralmente 4	-2.147.483.648 a 2.147.483.647
float	4	$3.4 \cdot 10^{-38}$ a $3.4 \cdot 10^{38}$ - seis dígitos de precisão
double	8	$1.7 \cdot 10^{-308}$ a $1.7 \cdot 10^{308}$ - dez dígitos de precisão
void	0	Sem valor

Tipos de dados

- A faixa é definida pelo tamanho, ex.: int (2 bytes):

1 byte = 8 bits

2 bytes = 16 bits

$2^{16} = 65.536$

$65.536 / 2 = 32.768$:

-32.768

+32.767

Tipos de dados

- Tipo int: usado para representar números inteiros, que podem ser 0, números positivos ou números negativos.

```
int idade;  
int ano;  
int numero_alunos;
```

Tipos de dados

- Tipos float e double: representam números reais.

```
float salario = 5000.00;  
double preco = 2.50;  
double pi = 3.14;
```

- Eles diferem quanto a precisão (casas decimais).
- Para representar um número quebrado deve-se sempre usar **ponto**, não vírgula.

Tipos de dados

- Tipo char: usado para caracteres (uma letra, um número, um símbolo).
- Existem duas formas de declarar uma variável do tipo char.

Tipos de dados

- Declaração de char - 1ª forma: passando o caractere entre **aspas simples**:

```
char letra = 'c';
```


Tipos de dados

- Declaração de char - 2ª forma: todo caractere é representado por um código numérico na tabela ASCII. Se o código do caractere for conhecido, é possível fazer a variável receber o código numérico (sem aspas) em vez do símbolo:

```
#include <stdio.h>

int main(void) {
    char c = 99;
    printf("Símbolo correspondente ao código %d: %c\n", c, c);
}
```

Saída:

Símbolo correspondente ao código 99: c

Tipos de dados

- Tabela ASCII:
https://www.ufrgs.br/wiki-r/images/d/d9/Tipos_de_dados%2CTipos_primarios_-_Raw.png

Tipos de dados

- Tipo void: “sem tipo”. Não pode ser usado na declaração de variáveis, é utilizado em funções que não precisam de retorno.

Tipos de dados

- Verificando o tamanho de um tipo de dado em C com o operador *sizeof*:

```
#include <stdio.h>

main()
{
    printf("Tamanho do tipo int: %d bytes\n", sizeof(int));
    printf("Tamanho do tipo char: %d bytes\n", sizeof(char));
    printf("Tamanho do tipo float: %d bytes\n", sizeof(float));
    printf("Tamanho do tipo double: %d bytes\n", sizeof(double));
}
```

Tipos de dados

- Modificadores de tipo:

Modificador	Aplica-se a	Significa
short	int	Menor dimensão
long	int, double	Maior dimensão
signed	char, int	Com sinal
unsigned	char, int	Sem sinal

Tipos de dados

- Modificadores de tipo - exemplos:

```
#include <stdio.h>

int main(void) {
    printf("tamanho - int: %d \n", sizeof(int));
    printf("tamanho - long int: %d \n", sizeof(long int));
    printf("tamanho - short int: %d \n", sizeof(short int));
}
```

Saídas:

tamanho - int: 4

tamanho - long int: 8

tamanho - short int: 2