

# Laboratório de Programação

Prof. Ms. Hugo Régis

Adaptado de:

Profa. Ms. Valéria Pinheiro



**UNIVERSIDADE  
FEDERAL DO CEARÁ**  
CAMPUS DE RUSSAS

# Aula passada....Um primeiro programa em C

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  void main() {
4      printf("Ola Mundo!");
5  }
```

<https://petbcc.ufscar.br/stdlib/>

<https://petbcc.ufscar.br/stdio/>

# Aula passada.... Um primeiro programa em C

- O seu programa deve comunicar-se com o usuário através de comandos de **impressão** (na tela) e de **leitura** (pelo teclado).
  - Ex: Pedir que o usuário digite um número inteiro.
  - Ex: Fornecer respostas ao usuário via impressão na tela

**Escrita** de mensagens na tela

- comando ***printf***

**Leitura** de dados pelo teclado

- comando ***scanf***

# Aula passada....Variáveis em C

```
1  int idade;  
2  float peso;  
3  char genero;  
4  double rendimento;
```

# Aula passada.... Um segundo programa em C

## Exemplo

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  void main() {
4      int soma, num1, num2;
5      printf("Informe o primeiro numero:");
6      scanf("%d", &num1);
7      printf("Informe o segundo numero:");
8      scanf("%d", &num2);
9
10     soma = num1 + num2;
11
12     printf("Resultado da soma: %d", soma);
13 }
```

# Aula 03 – Tipos

# Tipo

Conjunto de valores

- Inteiros: -2, -1, 0, 1, 2
- Reais: 2.5, 3.0, -201.001
- Lógicos: verdadeiro, falso
- Caractere: 'a', 'G', '\*', '#'
- Cadeia de caracteres (String): "Ana Maria", "Bom dia", "Olá Mundo"

# Tipo em C

Na Linguagem C

**int** → inteiros

**float** → reais com 6 casas decimais

**double** → reais com 10 casas decimais

**char** → caracteres

```
int tam = sizeof(int);  
printf("Tamanho do int = %d", tam);
```



# Tipo em C

**char** → caracteres

- Ocupa um 1byte de espaço
- Valores são usados baseando-se nos caracteres ASCII

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

# Tipo em C

**char** → caracteres

- Ocupa um 1byte de espaço
- Valores são usados baseando-se nos caracteres ASCII

palavra	gato			
letra	g	a	t	o
número	103	97	116	111
binário	01100111	01100001	01110100	01101111

# Tipo em C

```
#include <stdio.h>
int main(){
    char nome[] = "gato";
    int i;
    for (i = 0; nome[i] != '\0'; ++i){
        printf("%c\n", nome[i]);
    }
    printf("\n");
    return 0;
}
```

Saída:

g  
a  
t  
o

# Tipo em C

```
#include <stdio.h>
int main(){
    char nome[] = "gato";
    int i;
    for (i = 0; nome[i] != '\0'; ++i){
        printf("%d\n", nome[i]);
    }
    printf("\n");
    return 0;
}
```

Saída:

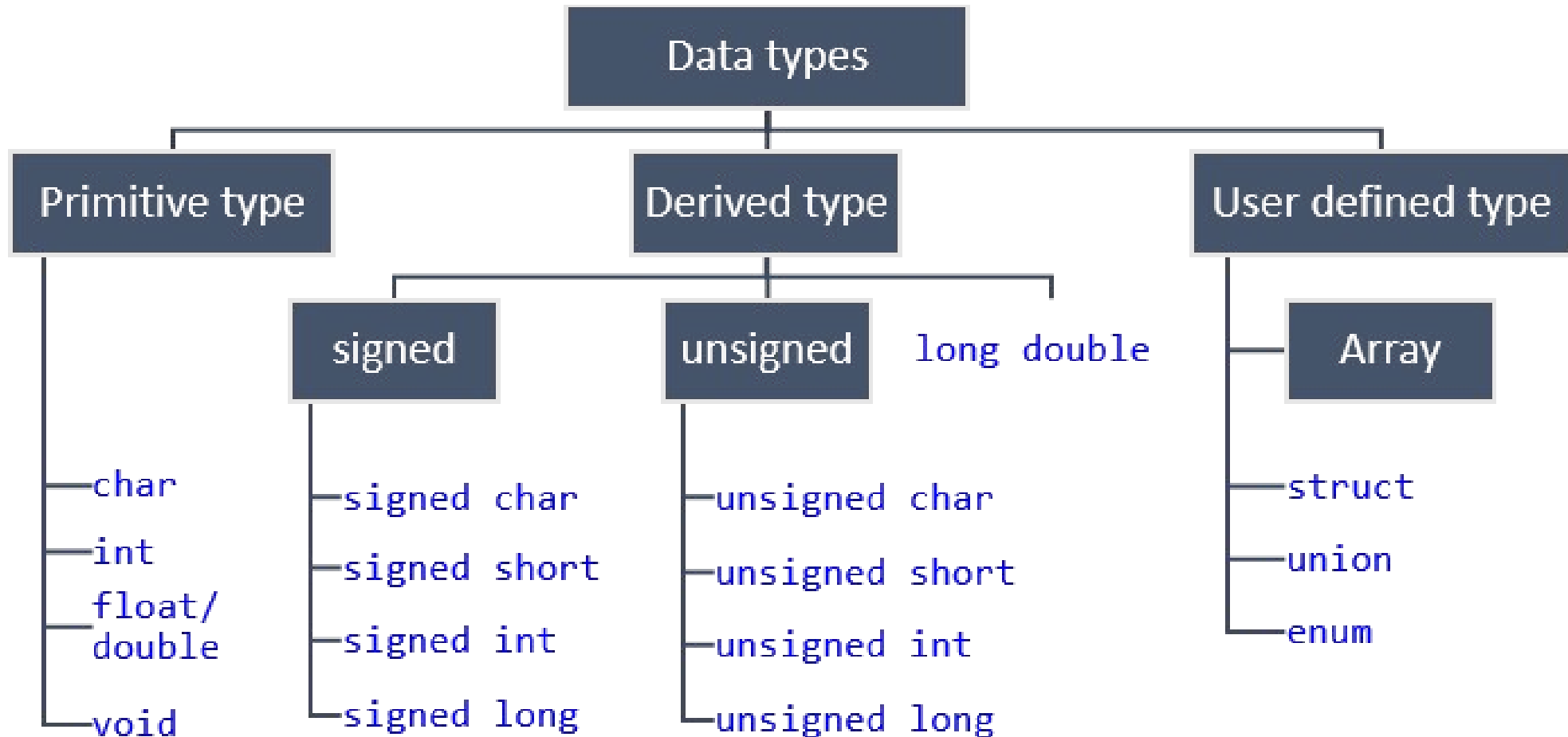
103

97

116

111

# Variações dos tipos básicos



# Tipo em C

Para usar uma variável, é necessário criá-la antes declarando o nome e o tipo da variável

```
int x;  
float n1, n2, soma;  
char nome;
```

Ao declarar uma variável, estamos reservando um espaço de memória para armazenar valores de um determinado tipo.

# Tipo em C

```
scanf("%d", &x);      --- inteiros  
scanf("%f", &x);      --- reais  
scanf("%c", &letra);  --- caracteres  
scanf("%s", &nome);   --- string
```

```
printf("%d", x);  
printf("Sexo: %c", sexo);  
printf("A soma de %f e %f eh igual a %f", n1, n2, soma);  
printf("Meu nome eh %s", nome);
```

# Atribuição

Comando que vai alterar o valor de uma variável com o resultado de uma expressão

Na linguagem C, o operador de atribuição é o símbolo igual =

```
X = 0;  
Y = 2.5;  
Z = 'm';
```





# Vamos praticar

- 1- Faça um programa em C que imprima o seu nome
- 2 - Faça um programa em C que leia e imprima um número inteiro
- 3- Faça um programa em C que leia dois números reais e os imprima
- 4- Faça um programa em C que imprima o produto dos valores 30 e 27
- 5- Faça um programa em C que imprima a média aritmética entre os números 5, 8, 12

# Vamos praticar

6 - Escreva um programa em C que declare e inicialize as seguintes variáveis: **idade**, **altura**, **letra**

7 - Escreva um programa que recebe 2 valores **a** e **b**, e troque esses valores, de modo que, no fim da execução, **b** possua o valor de **a** e vice-versa.

8 - Escreva um programa que recebe os valores de **altura** e **largura** para calcular a **área** de um retângulo.

9 - Faça um programa em C que leia dois números reais e calcule as quatro operações básicas entre estes dois números, adição, subtração, multiplicação e divisão. Ao final, o programa deve imprimir os resultados dos cálculos