

## 2 - TIPOS ALFANUMERICOS

- São formados por caracteres e strings
- Na linguagem C não existe o tipo de dados STRING
  - Na Programação tudo que estiver entre aspas duplas "dasd asdasd dasd" é uma string.

### - CARACTERE(unidade)

- Na programação em C, é tudo que possui aspas simples 'c'
- Declaração : char(%c)
- Tabela ASCII , diz respeito dos caracteres na computação, inicia do 0 a 127.

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

- O 0 em char é equivalente ao [null] e assim por diante.
- o a = 97 e o z = 122
- Isso significa para a gente que podemos gerar um alfabeto completo com um loop
- Vamos criar um programa chamado loop\_alfabeto.c

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_IDE\PROG_C\56(tipos_dados) > cd "C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_IDE\PROG_C\56(tipos_dados)"
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_IDE\PROG_C\56(tipos_dados) > cmd /c .\loop_alfabeto.exe
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_IDE\PROG_C\56(tipos_dados) >

1  /*
2     Alfabeto, tabela ASCII, com loop
3     97 = a | 122 = z
4  */
5
6  #include <stdio.h>
7  #include <stdlib.h>
8
9  int main()
10 {
11     //declarando_variaveis
12
13     //entrada_dados
14     for(int i = 97 ; i<=122;i++){
15         printf("%c\n",i);
16     }
17     //processamento_dados
18
19     //saida_dados
20
21
22
23     return 0;
24 }
```

- Quando mudamos para %d, aparece a contagem dos numeros ate o 122

```
> cd "C:\Users\Gabri\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\56(tipos_dados)"
PS C:\Users\Gabri\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\56(tipos_dados)
> cd /c .\loop_alfabeto.exe"
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
PS C:\Users\Gabri\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\56(tipos_dados)

#include <stdio.h>
/*
 * Alfabeto, tabela ASCII, com loop
 * 97 = a | 122 = z
 */
#include <stdio.h>
#include <stdio.h>
int main()
{
    //declarando_variaveis
    //entrada_dados
    for(int i = 97; i<=122;i++){
        printf("%d\n",i);
    }
    //processamento_dados
    //saida_dados
    return 0;
}
```

## STRINGS

- Para trabalharmos com strings em formas de variaveis

- Declaração : char nome[50]

- Recebimento do dado: gets(nome).

- Saida de dados : printf("%s",nome);

char nome[50] -> array do tipo char que suporta ate 49 caractere mais o (/0) caractere finalizador da string