

ASCII

ASCII (acrônimo para *American Standard Code for Information Interchange*, que em português significa "Código Padrão Americano para o Intercâmbio de Informação") é uma codificação de caracteres de oito bits baseada no alfabeto inglês. Os códigos ASCII representam texto em computadores, equipamentos de comunicação, entre outros dispositivos que trabalham com texto. Desenvolvida a partir de 1960, grande parte das codificações de caracteres modernas a herdaram como base.

A codificação define 128 caracteres, preenchendo completamente os sete bits disponíveis. Desses, 33 não são imprimíveis, como caracteres de controle atualmente não utilizáveis para edição de texto, porém amplamente utilizados em dispositivos de comunicação, que afetam o processamento do texto. Exceto pelo caractere de espaço, o restante é composto por caracteres imprimíveis.

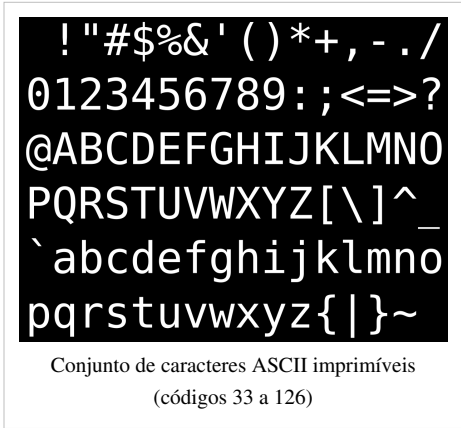


Tabela ASCII

Caracteres não imprimíveis

Representados como a parte não imprimível da tabela ASCII, os caracteres de controle tiveram sua origem nos primórdios da computação, quando se usavam máquinas teletipo e fitas de papel perfurado.

Binário	Decimal	Hexa	Controle	Abreviação	Descrição
0000 0000	00	00	^@	NUL	<i>Null</i> - Nulo
0000 0001	01	01	^A	SOH	<i>Start of Header</i> - Início do cabeçalho
0000 0010	02	02	^B	STX	<i>Start of Text</i> - Início do texto
0000 0011	03	03	^C	ETX	<i>End of Text</i> - Fim do texto
0000 0100	04	04	^D	EOT	<i>End of Tape</i> - Fim de fita
0000 0101	05	05	^E	ENQ	<i>Enquire</i> - Interroga identidade do terminal
0000 0110	06	06	^F	ACK	<i>Acknowledge</i> - Reconhecimento
0000 0111	07	07	^G	BEL	<i>Bell</i> - Campainha
0000 1000	08	08	^H	BS	<i>Back-space</i> - Espaço atrás
0000 1001	09	09	^I	HT	<i>Horizontal Tabulation</i> - Tabulação horizontal
0000 1010	10	0A	^J	LF	<i>Line-Feed</i> - Alimenta linha
0000 1011	11	0B	^K	VT	<i>Vertical Tabulation</i> - Tabulação vertical
0000 1100	12	0C	^L	FF	<i>Form-Feed</i> - Alimenta formulário
0000 1101	13	0D	^M	CR	<i>Carriage-Return</i> - Retorno do carro (enter)
0000 1110	14	0E	^N	SO	<i>Shift-Out</i> - Saída do <i>shift</i> (passa a usar caracteres de baixo da tecla - minúsculas, etc.)
0000 1111	15	0F	^O	SI	<i>Shift-In</i> - Entrada no <i>shift</i> (passa a usar caracteres de cima da tecla: maiúsculas, caracteres especiais, etc.)
0001 0000	16	10	^P	DLE	Data-Link Escape
0001 0001	17	11	^Q	DC1	Device-Control 1

0001 0010	18	12	^R	DC2	Device-Control 2
0001 0011	19	13	^S	DC3	Device-Control 3
0001 0100	20	14	^T	DC4	Device-Control 4
0001 0101	21	15	^U	NAK	<i>Neg-Acknowledge</i> - Não-reconhecimento
0001 0110	22	16	^V	SYN	Synchronous Idle
0001 0111	23	17	^W	ETB	End-of-Transmission Block
0001 1000	24	18	^X	CAN	Cancel
0001 1001	25	19	^Y	EM	End-Of-Medium
0001 1010	26	1A	^Z	SUB	Substitute
0001 1011	27	1B	^[ESC	Escape
0001 1100	28	1C	^\	FS	File Separator
0001 1101	29	1D	^]	GS	Group Separator
0001 1110	30	1E	^^	RS	Record Separator
0001 1111	31	1F	^_	US	Unit Separator
0111 1111	127	7F	^?	DEL	Delete

Caracteres imprimíveis

Binário	Decimal	Hexa	Glifo	
0010 0000	32	20		
0010 0001	33	21	[[Ponto de exclamação]]
0010 0010	34	22	"	
0010 0011	35	23	#	
0010 0100	36	24	\$	
0010 0101	37	25	%	
0010 0110	38	26	&	
0010 0111	39	27	'	
0010 1000	40	28	(
0010 1001	41	29)	
0010 1010	42	2A	*	
0010 1011	43	2B	+	
0010 1100	44	2C	,	
0010 1101	45	2D	-	
0010 1110	46	2E	.	
0010 1111	47	2F	/	
0011 0000	48	30	0	
0011 0001	49	31	1	
0011 0010	50	32	2	
0011 0011	51	33	3	
0011 0100	52	34	4	

0011 0101	53	35	5	
0011 0110	54	36	6	
0011 0111	55	37	7	
0011 1000	56	38	8	
0011 1001	57	39	9	
0011 1010	58	3A	:	
0011 1011	59	3B	;	
0011 1100	60	3C	<	
0011 1101	61	3D	=	
0011 1110	62	3E	>	
0011 1111	63	3F	?	

Binário	Decimal	Hexa	Glifo
0100 0000	64	40	@
0100 0001	65	41	A
0100 0010	66	42	B
0100 0011	67	43	C
0100 0100	68	44	D
0100 0101	69	45	E
0100 0110	70	46	F
0100 0111	71	47	G
0100 1000	72	48	H
0100 1001	73	49	I
0100 1010	74	4A	J
0100 1011	75	4B	K
0100 1100	76	4C	L
0100 1101	77	4D	M
0100 1110	78	4E	N
0100 1111	79	4F	O
0101 0000	80	50	P
0101 0001	81	51	Q
0101 0010	82	52	R
0101 0011	83	53	S
0101 0100	84	54	T
0101 0101	85	55	U
0101 0110	86	56	V
0101 0111	87	57	W
0101 1000	88	58	X
0101 1001	89	59	Y
0101 1010	90	5A	Z

0101 1011	91	5B	[
0101 1100	92	5C	\
0101 1101	93	5D]
0101 1110	94	5E	^
0101 1111	95	5F	_

Binário	Decimal	Hexa	Glifo
0110 0000	96	60	`
0110 0001	97	61	a
0110 0010	98	62	b
0110 0011	99	63	c
0110 0100	100	64	d
0110 0101	101	65	e
0110 0110	102	66	f
0110 0111	103	67	g
0110 1000	104	68	h
0110 1001	105	69	i
0110 1010	106	6A	j
0110 1011	107	6B	k
0110 1100	108	6C	l
0110 1101	109	6D	m
0110 1110	110	6E	n
0110 1111	111	6F	o
0111 0000	112	70	p
0111 0001	113	71	q
0111 0010	114	72	r
0111 0011	115	73	s
0111 0100	116	74	t
0111 0101	117	75	u
0111 0110	118	76	v
0111 0111	119	77	w
0111 1000	120	78	x
0111 1001	121	79	y
0111 1010	122	7A	z
0111 1011	123	7B	{
0111 1100	124	7C	
0111 1101	125	7D	}
0111 1110	126	7E	~

Exemplo de código para gerar os caracteres acima em diferentes linguagens de programação

Assembly

```
..start:
    mov     cx,95          ;número de caracteres
    mov     dl,32          ;número decimal do primeiro caractere, equivalente a 20H
    mov     ah,02          ;função que imprime caractere na tela
ASCII:
    int     21h
    inc     dl             ;próximo caractere
    loop    ASCII          ;repete mostra caractere

    mov     ax,4c00h        ;encerra o programa e retorna ao sistema operacional
    int     21h
```

Basic

```
10 FOR I = 32 TO 126
20 PRINT CHR$(I)
30 NEXT I
```

C

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int i=32;
    while(i < 127)
    {
        printf("%c[%d]\n",i,i);
        i++;
    }
}
```

C++

```
#include <iostream>
int main() {
    for ( char i = 32; i < 127; i++ ) {
        std::cout << i << std::endl;
    }
}
```

C#

```
using System;
static void Main(string[] args){
    for(int i=32; i<=126; i++)
        Console.WriteLine(i+" : "+(char) i);
}
```

Calc

```
32
m+
<<
repeat 94
    m
    sum
    1
    equ
    m+
    m
    out
    printc
    <<
endrepeat
```

Java

```
public class Ascii {
    public static void main(String args[]) {
        for (int i = 32; i <= 126; i++)
            System.out.println(i + " : " + (char) i);
    }
}
```

Javascript

```
for ( i = 0; i < 127; i++ ) {
    var Chars = String.fromCharCode( i );
    document.write( Chars );
}
```

Mumps

```
F i=32:1:126 W !,"Caractere ",i,":",$,C(i)
```

Pascal

```
program ASCII;
var i:integer;
begin
    for i := 32 to 126 do
        writeln('Caractere ',i,':',chr(i));
end.
```

Perl

```
print chr for 32..126;
```

Python

```
for x in range(32, 126): print (chr(x))
```

OpenEdge (Progress)

```
def var cont as integer.

def var texto as character format "x(10)" EXTENT 255.

do cont = 1 to 255:
    assign
        texto[cont] = string(cont) + " - " + chr(cont).
end.

repeat.
    disp texto
        with no-labels title "The book is on the Table ASCII "
        scrollbar-vertical.
end.
```

Visual FoxPro

```
FOR i=32 TO 126 STEP 1
    ? "Caracter " + CHR(i)
ENDFOR
```

Fontes e Editores da Página

ASCII *Fonte:* <http://pt.wikipedia.org/w/index.php?oldid=27132310> *Contribuidores:* !Silent, 200.185.142.xxx, 555, Adailton, Adrian Nicolaiev, Al Lemos, Alchimista, AlexSP, Anacarolgramatiqueira, Armagedon, Beria, Carla Cristiana Carvalho, Cesarius, David71rj, E2m, Edinopereira, Erwin, FML, Filipedavila, GOE, GRS73, Gauss, Get It, Hennet, Herr Klugbeisser, Jic, JoniFili, Jorge, JulioGold, Kimbrasil, Leonardo.stabile, LeonardoG, LeonardoRob0t, Lijealso, Luís Felipe Braga, Lvella, Marcio Henrique do Monte Furtado, MarioM, Master, MatheusMK3, Mocohome, OS2Warp, OriginalKratos, PRFS, Patirslm, Phillipe Israel, Pica83mail, Polemaco, ProfValente, Raafael, Rafaelsdm, Raphaelzinhoalves, Rei-artur, Reynaldo, Riverflow, Rjclaudio, Rodrigo.dst, Romulogs, Rui Malheiro, Ruy Pugliesi, SallesNeto BR, Sistema428, Stego, Tpsouza, Willianloco 2009, 223 edições anónimas

Fontes, Licenças e Editores da Imagem

Ficheiro:ASCII full.svg *Fonte:* http://pt.wikipedia.org/w/index.php?title=Ficheiro:ASCII_full.svg *Licença:* GNU Free Documentation License *Contribuidores:* Arite

Licença

Creative Commons Attribution-Share Alike 3.0 Unported
[//creativecommons.org/licenses/by-sa/3.0/](https://creativecommons.org/licenses/by-sa/3.0/)