

# Langage C

## Caractères et textes

# Syntaxe



## Caractère

Guillemet simple

'Z'

'x'

'8'

' '

...



## Chaîne de caractères

Guillemet double

"Hello"

"Why ?"

"15"

""

# Saisir et afficher / Caractère

```
#include <stdio.h>

int main() {
    char car;

    printf("Etes-vous d'accord ?\n");
    printf("o ... oui\n");
    printf("n ... non\n");
    car = getchar();
    printf("Reponse = %c ", car);
    return 0;
}
```

```
Etes-vous d'accord ?
o ... oui
n ... non
o
Reponse = o
```

# Codage ASCII

000	(nul)	016	► (dle)	032	sp	048	0	064	@	080	P	096	`	112	p
001	Ⓢ (soh)	017	◄ (dcl)	033	!	049	1	065	A	081	Q	097	a	113	q
002	Ⓢ (stx)	018	↑ (dc2)	034	"	050	2	066	B	082	R	098	b	114	r
003	▼ (etx)	019	!! (dc3)	035	#	051	3	067	C	083	S	099	c	115	s
004	Ⓢ (eot)	020	¶ (dc4)	036	\$	052	4	068	D	084	T	100	d	116	t
005	♣ (enq)	021	§ (nak)	037	%	053	5	069	E	085	U	101	e	117	u
006	♣ (ack)	022	— (syn)	038	&	054	6	070	F	086	V	102	f	118	v
007	▪ (bel)	023	‡ (etb)	039	'	055	7	071	G	087	W	103	g	119	w
008	■ (bs)	024	↑ (can)	040	(	056	8	072	H	088	X	104	h	120	x
009	(tab)	025	↓ (em)	041	)	057	9	073	I	089	Y	105	i	121	y
010	(lf)	026	(eof)	042	*	058	:	074	J	090	Z	106	j	122	z
011	♂ (vt)	027	← (esc)	043	+	059	;	075	K	091	[	107	k	123	{
012	Ⓢ (np)	028	L (fs)	044	,	060	<	076	L	092	\	108	l	124	
013	(cr)	029	↔ (gs)	045	-	061	=	077	M	093	]	109	m	125	}
014	♂ (so)	030	▲ (rs)	046	.	062	>	078	N	094	^	110	n	126	~
015	♂ (si)	031	▼ (us)	047	/	063	?	079	O	095	_	111	o	127	o

Le caractère '**M**' a pour code ASCII **77**

Le code ASCII **77** représente le caractère '**M**'

# Coder

```
#include <stdio.h>

int main() {
    char car = 'M';

    printf("Code ASCII = %d", car);

    return 0;
}
```

Code ASCII = 77

Le caractère 'M' a pour code ASCII 77

# Décoder

```
#include <stdio.h>

int main() {
    int code = 77;

    printf("Caractere = %c", code);

    return 0;
}
```

Caractere = M

Le code ASCII **77** représente le caractère '**M**'

# Un caractère est-il une lettre majuscule ?

000 (nul)	016 ► (dle)	032 sp	048 0	064 @	080 P	096 `	112 p
001 ☉ (soh)	017 ◄ (dcl)	033 !	049 1	065 A	081 Q	097 a	113 q
002 ☉ (stx)	018 ↑ (dc2)	034 "	050 2	066 B	082 R	098 b	114 r
003 ♥ (etx)	019 !! (dc3)	035 #	051 3	067 C	083 S	099 c	115 s
004 + (eot)	020 ¶ (dc4)	036 \$	052 4	068 D	084 T	100 d	116 t
005 ♣ (enq)	021 § (nak)	037 %	053 5	069 E	085 U	101 e	117 u
006 ♣ (ack)	022 − (syn)	038 &	054 6	070 F	086 V	102 f	118 v
007 • (bel)	023 † (etb)	039 '	055 7	071 G	087 W	103 g	119 w
008 ■ (bs)	024 † (can)	040 (	056 8	072 H	088 X	104 h	120 x
009 (tab)	025 ↓ (em)	041 )	057 9	073 I	089 Y	105 i	121 y
010 (lf)	026 (eof)	042 *	058 :	074 J	090 Z	106 j	122 z
011 ♂ (vt)	027 ← (esc)	043 +	059 ;	075 K	091 [	107 k	123 {
012 ♣ (np)	028 L (fs)	044 ,	060 <	076 L	092 \	108 l	124
013 (cr)	029 ↔ (gs)	045 -	061 =	077 M	093 ]	109 m	125 }
014 ♂ (so)	030 ▲ (rs)	046 .	062 >	078 N	094 ^	110 n	126 ~
015 ♂ (si)	031 ▼ (us)	047 /	063 ?	079 O	095 _	111 o	127 ò

```
char car;  
printf("Votre caractere : ");  
car = getchar();  
if (car >= 'A' && car <= 'Z')  
    printf("Majuscule", car);
```

# Un caractère est-il une lettre minuscule ?

000 (nul)	016 ► (dle)	032 sp	048 0	064 @	080 P	096 `	112 p
001 ☉ (soh)	017 ◄ (dcl)	033 !	049 1	065 A	081 Q	097 a	113 q
002 ☉ (stx)	018 ↑ (dc2)	034 "	050 2	066 B	082 R	098 b	114 r
003 ♥ (etx)	019 !! (dc3)	035 #	051 3	067 C	083 S	099 c	115 s
004 + (eot)	020 ¶ (dc4)	036 \$	052 4	068 D	084 T	100 d	116 t
005 ♣ (enq)	021 § (nak)	037 %	053 5	069 E	085 U	101 e	117 u
006 + (ack)	022 − (syn)	038 &	054 6	070 F	086 V	102 f	118 v
007 • (bel)	023 † (etb)	039 '	055 7	071 G	087 W	103 g	119 w
008 ■ (bs)	024 † (can)	040 (	056 8	072 H	088 X	104 h	120 x
009 (tab)	025 ↓ (em)	041 )	057 9	073 I	089 Y	105 i	121 y
010 (lf)	026 (eof)	042 *	058 :	074 J	090 Z	106 j	122 z
011 ♂ (vt)	027 ← (esc)	043 +	059 ;	075 K	091 [	107 k	123 {
012 ♣ (np)	028 L (fs)	044 ,	060 <	076 L	092 \	108 l	124
013 (cr)	029 ↔ (gs)	045 -	061 =	077 M	093 ]	109 m	125 }
014 ♂ (so)	030 ▲ (rs)	046 .	062 >	078 N	094 ^	110 n	126 ~
015 ♂ (si)	031 ▼ (us)	047 /	063 ?	079 O	095 _	111 o	127 ò

```
char car;  
printf("Votre caractere : ");  
car = getchar();  
if (car >= 'a' && car <= 'z')  
    printf("Minuscule", car);
```



# Opérations arithmétiques sur un code ASCII

```
#include <stdio.h>

int main() {
    char car = 'A';

    printf("car = %c\n", car);
    printf("car plus un = %c\n", car + 1);
    printf("car plus deux = %c\n", car + 2);
    printf("car moins un = %c\n", car - 1);
    return 0;
}
```

	064 @
	065 A
Extrait	066 B
code ASCII	067 C
	068 D
	069 E

```
car = A
car plus un = B
car plus deux = C
car moins un = @
```

# Saisir et afficher / Chaîne de caractères

```
#include <stdio.h>

int main() {
    char prenom[100];

    printf("Ton prenom ?\n");
    fgets(prenom, 100, stdin);
    printf("Hello %s", prenom);
    printf("Debut = %c\n", prenom[0]);
    return 0;
}
```

```
Ton prenom ?
Solveig
Hello Solveig
Debut = S
```

# Coder et décoder

```
#include <stdio.h>
int main() {
    char txt[100] = "Why ?";

    printf("%d\n", txt[0]);
    printf("%d\n", txt[1]);
    printf("%d\n", txt[2]);
    printf("%d\n", txt[3]);
    printf("%d\n", txt[4]);
    printf("%d\n", txt[5]);
    return 0;
}
```

87  
104  
121  
32  
63  
0

txt	'W'	'h'	'y'	' '	'?'	'\0'	?	?	?	
	0	1	2	3	4	5	6	7	8	

Le caractère '`\0`' est le caractère fin de chaîne de caractères.

# Modifier une chaîne

txt	'G'	'u'	'n'	'n'	'a'	'r'	's'	'o'	'n'	'\0'
	0	1	2	3	4	5	6	7	8	9

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

int main() {
    char txt[100] = "Gunnarson";

    txt[6] = '\0';
    printf("%s", txt);
    return 0;
}
```

Gunnar

txt	'G'	'u'	'n'	'n'	'a'	'r'	'\0'	'o'	'n'	'\0'
	0	1	2	3	4	5	6	7	8	9

# Manipuler une chaîne de caractères

```
#include <stdio.h>

int main() {
    char txt[100] = "Gunnar";

    printf("%s\n", txt);
    txt = "Erik";
    printf("%s\n", txt);
    return 0;
}
```

En bleu, la déclaration de txt  
et son initialisation



En rouge,  
le fail !!!

Error : assignment to expression with array type

# Manipuler une chaîne de caractères

txt	'G'	'u'	'n'	'n'	'a'	'r'	'\0'	'?'	'?'	
	0	1	2	3	4	5	6	7	8	

```
#include <stdio.h>
int main() {
    char txt[100] = "Gunnar";
    printf("%s\n", txt);
    txt[0] = 'E';
    txt[1] = 'r';
    txt[2] = 'i';
    txt[3] = 'k';
    txt[4] = '\0';
    printf("%s\n", txt);
    return 0;
}
```

Gunnar  
Erik

txt	'E'	'r'	'i'	'k'	'\0'	'r'	'\0'	'?'	'?'	
	0	1	2	3	4	5	6	7	8	

# Longueur d'une chaîne

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

int main() {
    char txt[100] = "Gunnarson";

    printf("%d\n", strlen(txt));
    printf("%s", txt);
    return 0;
}
```

9  
Gunnarson

txt	'G'	'u'	'n'	'n'	'a'	'r'	's'	'o'	'n'	'\0'
	0	1	2	3	4	5	6	7	8	9

# Parcourir une chaîne

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

int main() {
    char pre[100] = "GuNnArSoN";
    int cpt;

    for (cpt = 0; cpt < strlen(pre); cpt++) {
        printf("%c\n", pre[cpt]);
    }
    return 0;
}
```

G  
U  
N  
n  
A  
r  
S  
O  
N