



## Chap. 1: Éléments de langage

# Les caractères permis

- ▶ Lettres majuscules et minuscules
- ▶ Chiffres
- ▶ Caractères spéciaux  
! " # % & ' ( ) \* + , - . / : ; < = > ? [ \ ] ^ \_ { | }
- ▶ Séparateurs de mots  
(espace, tabulation, passage à la ligne, etc.)

# Les commentaires

- Balises /\* et \*/

```
/* commentaire sur  
   plusieurs lignes */
```

- Commentaires de fin de ligne  
(depuis la norme C99)

```
// commentaire de  
// fin de ligne
```

# 32 mots réservés

<b>auto</b>	<b>double</b>	<b>int</b>	<b>struct</b>
<b>break</b>	<b>else</b>	<b>long</b>	<b>switch</b>
<b>case</b>	<b>enum</b>	<b>register</b>	<b>typedef</b>
<b>char</b>	<b>extern</b>	<b>return</b>	<b>union</b>
<b>const</b>	<b>float</b>	<b>short</b>	<b>unsigned</b>
<b>continue</b>	<b>for</b>	<b>signed</b>	<b>void</b>
<b>default</b>	<b>goto</b>	<b>sizeof</b>	<b>volatile</b>
<b>do</b>	<b>if</b>	<b>static</b>	<b>while</b>

# Les types de base (1)

Type	Description	Taille	Valeur
void	Type générique		
char	caractère/octet	1 octets	comme signed ou unsigned char
unsigned char	caractère/octet non signé	1 octets	0 à 255
signed char	caractère/octet signé	1 octets	-128 à 127
short	entier court signé	2 octets	-32 768 à 32 767
unsigned short	entier non signé	2 octets	0 à 65 535
int	entier signé	2 ou 4 octets	(en fonction du compilateur)
unsigned	entier non signé	2 ou 4 octets	(en fonction du compilateur)
long	entier signé long	4 octets	-2 147 483 648 à 2 147 483 647
unsigned long	entier non signé long	4 octets	0 à 4 294 967 295 ( $2^{32}-1$ )
float	flottant	4 octets	Mantisse : +- 6 chiffres significatifs
double	flottant	8 octets	Mantisse : +- 12 chiffres significatifs
long double	flottant	16 octets	

# Les types de base (2)

- ▶ Les tailles et valeurs des différents types sont données dépendent de l'architecture informatique.
- ▶ Les limites de chaque type sont données dans le fichier d'include `limits.h` :

`CHAR_MIN, CHAR_MAX, INT_MIN, INT_MAX...`

# Le type booléen

- ▶ C ne définit pas de type booléen
- ▶ Généralement utilisation d'un type entier:
  - FAUX → valeur nulle 0
  - VRAI → toute autre valeur
- ▶ Pour plus de lisibilité, utilisation de la bibliothèque standard ***stdbool.h*** qui définit le type **bool** avec deux valeurs:
  - true** = 1
  - false** = 0

# Les identificateurs

- ▶ composés de lettres, de chiffres et du caractère '\_'
- ▶ case-sensitive
- ▶ Définition de variable :  
spécification du type, de l'identificateur et éventuellement d'une valeur

```
int a, b, c=5, d;  
int taille  = 3 * 4;  
int tailleD = taille * 2;
```



# Les constantes

Deux manières de définir une constante:

- ▶ une macro via la directive au préprocesseur `#define` :

```
#define MAX 10
```

- ▶ une variable qualifiée de constante via le mot réservé `const` :

```
const int MAX = 10;
```

# Priorité des opérateurs

## Conversions de types

## Littéraux

► cf. syllabus