

## Types de données

### Les types numériques:

#### **Type entier**

**Domaine de valeurs :** sous-ensemble de  $\mathbb{Z}$ . Max=32767

#### **Remarque:**

Type	Domaine de def	Nombre de bits
SHORTINT	-128..127	Signé 8 bits
INTEGER	-32768..32767	Signé 16 bits
LONGINT	-2147483648..2147483647	Signé 32 bits
BYTE	0..255	Non signé 8 bits
WORD	0..65535	Non signé 16 bits

#### **Opérateurs arithmétiques :**

$+$ ,  $-$ ,  $*$ ,  $\div$  et  $\text{mod}$

**div:** donne le quotient dans la division entière.

**mod:** donne le reste de la division entière.

$$\begin{array}{r|l} \text{reste} & 23 \quad | \quad 7 \quad \text{quotient} \\ & \downarrow \quad | \quad \uparrow \\ & 2 \quad | \quad 3 \\ & \hline & 23 \text{ div } 7 = \\ & 23 \text{ mod } 7 = \end{array}$$

**En Pascal:** var <nom\_variable> : integer;

**Exemple:** var m, i, j : integer ;

#### **Type réel**

**Domaine de valeurs :** sous ensemble des nombres réels  $\mathbb{R}$ .

$2.6 \cdot 10^2$   $2.6E+2$  mantisse et exposant

#### **Opérateurs arithmétiques:**

$+$ ,  $-$ ,  $*$ ,  $/$  (division réelle)

#### **Dans le langage pascal:**

var <nom\_variable> : real ;

**Exemple :** Var x , y : real ;

\*Voir les fonctions arithmétiques standards:

#### **Type booléen**

**Domaine de valeur :** deux valeurs logiques VRAI et FAUX (true, false)

#### **Opérateurs logiques sur les booléen:**

**NON:** négation NOT

**ET:** conjonction AND

**OU:** disjonction OR

**Ouex:** Ou exclusif XOR

#### **Tableau de vérité:**

x	y	NON(x)	x et y	x ou y	x ouex y
Faux	Faux	Vrai	Faux	Faux	Faux
Faux	Vrai	Vrai	Faux	Vrai	Vrai
Vrai	Vrai	Faux	Vrai	Vrai	Faux
Vrai	Faux	Faux	Faux	Vrai	Vrai

#### **Exemples:**

$(15 > 5)$  ET  $(3 < 6)$  = Vrai

Vrai ET Vrai

$(15 > 5)$  OU  $(2 > 5)$  = Vrai

Vrai OU Faux

$(10 > 5)$  OU  $(20 > 40)$  ET  $(3 < 2)$  = Vrai

Vrai OU (Faux ET Faux)

Vrai OU Faux

#### **Ordre de priorité:**

Entre parenthèses prioritaires.

Non>ET>ou=ouex

#### **Dans le langage pascal:**

var <nom\_variable> : boolean ;

#### **Exemple:**

var flag, test :boolean ;

begin

flag:=true ; test:=5<2 ;

end.

#### **Type caractère**

**Domaine de valeur :** tous caractère alphanumérique imprimable de l'alphabet latin sont admis (lettre majuscule, minuscule des chiffres, symboles spéciaux (des ponctuation, signes et autres) aussi les caractères non imprimable:retour chariot, l'échappe, le bip sonore.

Tout ses caractères sont ordonnés selon leurs codes ASCII (American Standard Code For Information Interchange). Norme utilisée pour le codage binaire des caractères.(256 caractères)

"0"<"1"< ... "A" < "B"<... "a" < "b"<...

Une variable de type caractère contient un seul caractère. L'espace est un caractère blanc.

#### **Opérateurs sur les caractères:**

Type caractère ordonné, donc : les opérateurs relationnels y sont définis.

<, >, =, <=, >=, <>

**Exemple:** "A" < "B" vrai

#### **Dans le langage pascal:**

var <nom\_variable> : char ;

**Exemple:** var c1,c2 : char ;

begin

c1:='A' ; c2:='B' ;

end.

\*Voir Les fonctions prédéfinies sur les caractères:

#### **Type chaîne de caractères**

**Définition :** une chaîne de caractères est une entité composée d'une suite de n caractères.  $0 \leq n \leq 255$ . si  $n=0$  on dit que la chaîne est vide.

Les valeurs de chaînes de caractères sont définies entre guillemets.

#### **Dans le langage pascal:**

Var <nom\_var> : String[<taille>] ;

#### **Exemple:**

Var

nom : string[10] ;

prenom : string; { indéfini  $\rightarrow$  max=255 }

**begin** nom:='Bac2013' ; **end.**

#### **Exemple:**

nom  $\leftarrow$  "" vide

prénom  $\leftarrow$  " " espace

ville  $\leftarrow$  "Sousse"

**Remarque:** On peut accéder en lecture et en écriture au  $i^{\text{ème}}$  caractère d'une chaîne CH en utilisant la notion CH[i] où  $1 \leq i \leq \text{long}(\text{ch})$ .

#### **Exemple:**

CH  $\leftarrow$  "Tunisie" c  $\leftarrow$  ch[1]

ch[7]  $\leftarrow$  "a" donc ch devient "Tunisia"

#### **Remarque:**

On peut utiliser les opérateurs relationnels sur les chaînes:

**Exemple:** "Mohamed" > "Ali"

"Said" < "Saleh"

**Rq:** Comparaison caractère par caractère.

\* Voir les fonctions et les procédures standard sur les chaînes de caractère: