

# 3. Les types scalaires

---

Type d'une variable = ensemble des valeurs qu'elle peut prendre . Ensemble muni des opérations possibles sur ces valeurs.

Objets scalaires : (scalaris = échelon) totalement ordonné.

# Les types de base

---

- Le type entier

Il est muni des opérations suivantes :

+   -   \*   div   mod

- Le type réel

Il est muni des opérations suivantes :

+   -   \*   /

- En mathématique, l'ensemble des entiers est inclus dans l'ensemble des réels.

En programmation, les entiers et les réels ont des représentations différentes.

# Exemple

---

```
Var i : entier  
      b : réel  
i ← 5  
b ← i
```

L'affectation respecte l'inclusion de  $\mathbb{N}$  dans  $\mathbb{R}$ , mais s'accompagne d'un changement de représentation interne : le contenu de  $b$  est 5, mais la représentation interne est de la forme  $0,5 \cdot 10^1$ .

L'affectation d'une variable réelle à une entière n'est pas possible directement même si cette variable contient une valeur entière.

```
b ← 3.0  
i ← trunc(b) {partie entière}
```

# Le type booléen (1/3)

---

- Une *proposition* a l'une des deux valeurs : vrai (V), faux (F).

*Exemple* : La proposition  $4 > 10$  a la valeur faux.

- Opérations sur les propositions : dans l'ensemble des deux propositions V et F, on définit des opérations logiques dont les plus importantes sont

conjonction (et)	$\wedge$
disjonction inclusive (ou)	$\vee$
négation (non)	$\neg$
implication	$\Rightarrow$
équivalence	$\Leftrightarrow$
disjonction exclusive (ou exclusif)	$\oplus$

Les opérateurs logiques sont appelés connecteurs.

# Le type booléen (2/3)

---

## Les connecteurs

p	q	$p \wedge q$	$p \vee q$	$p \Rightarrow q$	$p \Leftrightarrow q$	$p \text{w} q$
F	F	F	F	V	V	F
F	V	F	V	V	F	V
V	F	F	V	F	F	V
V	V	V	V	V	V	F

# Le type booléen (3/3)

---

□ Chacune de ces opérations peut s'exprimer à l'aide de  $\wedge$ ,  $\vee$ ,  $\neg$ .

$$p \Rightarrow q = \neg p \vee q$$

$$p \Leftrightarrow q = (\neg p \vee q) \wedge (\neg q \vee p)$$

$$\neg(p \vee q) = \neg p \wedge \neg q$$

$$\neg(p \wedge q) = \neg p \vee \neg q$$

$$p \oplus q = \neg(p \Leftrightarrow q) = \neg [(\neg p \vee q) \wedge (\neg q \vee p)] = \neg(\neg p \vee q) \vee \neg(\neg q \vee p) = (p \wedge \neg q) \vee (q \wedge \neg p)$$

□ Le type booléen

Il est constitué de deux éléments : vrai, faux.

Il est muni des opérations :  $\wedge$ ,  $\vee$ ,  $\neg$

Il est totalement ordonné par la relation : faux < vrai

# Le type caractère

---

- Ensemble des caractères d'imprimerie, noté car.  
Une variable de type car peut prendre toute valeur de ce type.  
Une constante de type car se distingue par son écriture  
'O', 'A', 'o', 'a', '+'.
- Codification des caractères (configuration binaire)  
→ code les plus utilisés = code ASCII, UTF8
- Fonctions définies  
ord : pour obtenir le code ASCII du caractère donné  
chr : pour obtenir le caractère du code ASCII donné  
prec : caractère précédent  
succ : caractère suivant

# Les autres types scalaires

---

- Type défini par énumération

    saison : (prin, été, aut, hiver)

        prin < été < aut < hiver

- Type intervalle

    0 .. 10                    pour    {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

    'a' .. 'f'            pour    {'a', 'b', 'c', 'd', 'e', 'f'}

    été .. hiv            pour    {été, aut, hiver}

# Ordres de priorité

---

1.	Négation	Non						
2.	Opérateurs multiplicatifs	*	/	div	mod	et		
3.	Opérateurs additifs	+	-	ou				
4.	Opérateurs relationnels	=	≠	<	≤	>	≥	dans