

# Java

## Les primitives

# Représentation d'éléments

- La représentation de n'importe quoi en informatique (chiffres, nombres, lettres, etc.) est composée de 0 et de 1 (binaire).
- 1 bit a 2 valeurs ( $2^1 = 2$ ) possibles: 0 ; 1
- 2 bits a 4 valeurs ( $2^2 = 4$ ) possibles: 00 ; 01 ; 10 ; 11
- 3 bits a 8 valeurs ( $2^3 = 8$ ) possibles
- n bits a  $2^n$  valeurs possibles

# Représentation des entiers

- Les entiers sont des nombres sans virgules. Ex: 1 ; 2 ; -5 ; 0
- En Java, les nombres ont toujours un signe positif ou négatif.
- Un bit est donc utilisé pour représenter le signe + ou - .
- Pour 1 octet (8 bits), il y a  $2^8 = 256$  valeurs possibles.
- Ce nombre est divisé par 2 puisqu'un bit est utilisé pour le signe. (Il reste donc  $2^7 = 128$  valeurs positives et valeurs négatives)
- $\text{Min} = -2^7 = -128$
- $\text{Max} = 2^7 - 1 = 127$ 
  - Le (-1) est pour compter le nombre « 0 » qui fait partie des positifs
- La représentation interne dépend de la JVM

# Types et grandeurs (entiers)

Type	Taille	Nb Valeurs	Min	Max
boolean	1 bit	$2^1 = 2$	0 (false)	1 (true)
byte	8 bits = 1 octet	$2^8 = 256$	-128	127
short	16 bits = 2 octets	$2^{16} = 65'536$	-32'768	32'767
char	16 bits = 2 octets	$2^{16} = 65'536$	0	65'535
int	32 bits = 4 octets	$2^{32} = 4'294'967'296$	-2'147'483'648	2'147'483'647
long	64 bits = 8 octets	$2^{64} = 1.8 * 10^{19} (*1)$	*2	*3

\*1: 18'446'744'073'709'551'616

\*2: -9'223'372'036'854'775'808

\*3: 9'223'372'036'854'775'807

## Représentation des nombres à virgules flottantes

- Les nombres à virgules flottantes sont utilisés pour les nombres rationnels. Ex: -4.5 ; 3.14 ; 104.67
- Utilisent le standard « IEEE 754 floating point ».
- Ils sont constitués d'un signe, d'une mantisse et d'un exposant. De plus, certaines valeurs spéciales existent comme -0, +0, Inf et NaN

# Exemple: notation scientifique

- Imaginons que nous avons 9 cases : 5 pour la mantisse et 4 pour l'exposant.
- Nous pouvons faire des nombres composés de
  - $MMMMM * 10^{EEEE}$
  - Le plus petit (non 0) :  $00001 * 10^{-9999} = 0.000...01$
  - Le plus grand :  $99999 * 10^{9999} = 10000...00$
- En binaire
  - $MMMMM * 2^{EEEE}$
  - Le plus petit (non 0) :  $00001 * 2^{-1111}$
  - Le plus grand :  $11111 * 2^{1111}$

# Types et grandeurs (point flottant)

Type	Taille	Bits mantisse	Bits exposant
float	32 bits = 4 octets	23 bits	8 bits
double	64 bits = 8 octets	52 bits	11 bits

N'oubliez pas qu'il y a un bit pour le signe:  $23 + 8 + 1 = 32$  bits.

Les JVM peuvent aussi supporter les formats étendus, mais elles doivent au moins supporter les formats de base.

# Code

- <https://github.com/foilen/foilen-samples/tree/master/java-primitives>



# Références

- <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
- <https://www.noslangues-ourlanguages.gc.ca/bien-well/fra-eng/typographie-typography/nmbrs-fra.html>
- [http://fr.wikipedia.org/wiki/IEEE\\_754](http://fr.wikipedia.org/wiki/IEEE_754)