



LES TYPES DE DONNEES

Types Primitifs

Nom	Taille	Valeurs	
int	4 octets (32 bits)	- 2^{31} à $2^{31} - 1$	-2147483648 à 2147483647
short	2 octets (16 bits)	- 2^{15} à $2^{15} - 1$	-32768 à 32767
long	8 octets (64bits)	- 2^{63} à $2^{63} - 1$	environ $-9.2 \cdot 10^{18}$ à $9.2 \cdot 10^{18}$
byte	1 octet (8 bits)	- 2^7 à $2^7 - 1$	-127 à +128
float	4 octets (32 bits)		environ +/- $1.4 \cdot 10^{-45}$ à +/- $3.4 \cdot 10^{38}$
double	8 octets (64 bits)		environ +/- $4.9 \cdot 10^{-324}$ à +/- $1.7 \cdot 10^{308}$
char	2 octets (16 bits)	Codage Unicode	\u0000 à \uFFFF
boolean	1 bit	true ou false	



LES TYPES DE DONNEES

Déclaration des Variables

Chaque variable est définie par un type et un identificateur.

ex: **short distanceParcourue;**

L'identificateur doit être représentatif de la fonction de la variable.

ex: **moyenneHoraire, nombreParticipants,...**

Conventions:

L'initiale est en minuscule, les initiales des autres mots composant l'identificateur sont en Majuscule.



LES TYPES DE DONNEES

Déclaration des Variables

Constitution des Identificateurs

- Le premier caractère ne doit pas être un chiffre
ex: le nom ~~1Caractere~~ n'est pas correct
- Aucun espace.
- Les majuscules sont différentes des minuscules :
ex: **mavaleur** est différent de **maValeur**
- Caractères interdits:
& ~ " # ' { } () [] - | ` \ ^ @ = % * ? : / § ! < > £ ,



LES TYPES DE DONNEES

Déclaration des Variables

Les identificateurs doivent être différents des mots clefs

abstract	default	goto ¹⁾	package	synchronized
assert ³⁾	do	if	private	this
boolean	double	implements	protected	throw
break	else	import	public	throws
byte	enum ⁴⁾	instanceof	return	transient
case	extends	int	short	true ²⁾
catch	false ²⁾	interface	static	try
char	final	long	strictfp	void
class	finally	native	super	volatile
const ¹⁾	float	new	switch	while
continue	for	null ²⁾		



LES TYPES DE DONNEES

Déclaration des Variables

Déclaration et Initialisation séparées

```
int maVariable;
```

.....

```
maVariable = 1;
```

Déclaration et Initialisation simultanées

```
int maVariable = 1;
```



LES TYPES DE DONNEES

Initialisation des nombres entiers

```
int maVariable = 458; // valeur en Décimal
```

```
int maVariable = 0x45; // valeur en Hexadécimal
```

```
int maVariable = 0256; //valeur en Octal
```

```
int maVariable = 0b1011; //valeur en Binaire
```

```
int maVariable = monAutreValeur /* Initialisation avec la  
valeur d'une autre variable*/
```



LES TYPES DE DONNEES

Initialisation des Caractères

Le caractère (type char) est codé en Unicode sur 4 octets (65536 combinaisons).

L'Unicode est une extension du codage ASCII qui permet de coder tous les caractères internationaux

-Un caractère Unicode, est représenté par un code d'échappement `\u` suivi de 4 chiffres hexadécimaux représentant le code du caractère:

ex: `\u0000` , `\u2122`, `\u0035`



LES TYPES DE DONNEES

Initialisation des Caractères

Table de caractères Unicode (Alphabet Latin de Base).

0020	0	0030	@	0040	P	0050	`	0060	p	0070		00A0	°	00B0	À	00C0	Ð	00D0	à	00E0	ð	00F0
!	1	0031	A	0041	Q	0051	a	0061	q	0071	i	00A1	±	00B1	Á	00C1	Ñ	00D1	á	00E1	ñ	00F1
"	2	0032	B	0042	R	0052	b	0062	r	0072	¢	00A2	²	00B2	Â	00C2	Ò	00D2	â	00E2	ò	00F2
#	3	0033	C	0043	S	0053	c	0063	s	0073	£	00A3	³	00B3	Ã	00C3	Ó	00D3	ã	00E3	ó	00F3
\$	4	0034	D	0044	T	0054	d	0064	t	0074	¤	00A4	´	00B4	Ä	00C4	Ô	00D4	ä	00E4	ô	00F4
%	5	0035	E	0045	U	0055	e	0065	u	0075	¥	00A5	µ	00B5	Å	00C5	Õ	00D5	å	00E5	õ	00F5
&	6	0036	F	0046	V	0056	f	0066	v	0076		00A6	¶	00B6	Æ	00C6	Ö	00D6	æ	00E6	ö	00F6
'	7	0037	G	0047	W	0057	g	0067	w	0077	§	00A7	·	00B7	Ç	00C7	×	00D7	ç	00E7	÷	00F7
(8	0038	H	0048	X	0058	h	0068	x	0078	"	00A8	,	00B8	È	00C8	Ø	00D8	è	00E8	ø	00F8
)	9	0039	I	0049	Y	0059	i	0069	y	0079	©	00A9	¹	00B9	É	00C9	Ù	00D9	é	00E9	ù	00F9
*	:	003A	J	004A	Z	005A	j	006A	z	007A	ª	00AA	º	00BA	Ê	00CA	Ú	00DA	ê	00EA	ú	00FA
+	;	003B	K	004B	[005B	k	006B	{	007B	«	00AB	»	00BB	Ë	00CB	Û	00DB	ë	00EB	û	00FB
,	<	003C	L	004C	\	005C	l	006C		007C	¬	00AC	¼	00BC	Ì	00CC	Ü	00DC	ì	00EC	ü	00FC
-	=	003D	M	004D]	005D	m	006D	}	007D	-	00AD	½	00BD	Í	00CD	Ý	00DD	í	00ED	ý	00FD
.	>	003E	N	004E	^	005E	n	006E	~	007E	®	00AE	¾	00BE	Î	00CE	Þ	00DE	î	00EE	þ	00FE
/	?	003F	O	004F	_	005F	o	006F		007F	™	00AF	¿	00BF	Ï	00CF	ß	00DF	ï	00EF	ÿ	00FF



LES TYPES DE DONNEES

Initialisation des Caractères

Un **char** peut être initialisé de 2 façons:

```
char monCaract = 'Z' // Initialisation entre simples quotes
```

```
char monCaract = '\u005A' // code unicode de 'Z'
```



LES TYPES DE DONNEES

Initialisation des Caractères

Quelques caractères d'échappement particuliers:

Caractère d'échappement	Fonction	Valeur Unicode
<code>\b</code>	Effacement arrière	<code>\u0008</code>
<code>\t</code>	Tabulation horizontale	<code>\u0009</code>
<code>\n</code>	Saut de ligne	<code>\u000a</code>
<code>\r</code>	Retour chariot	<code>\u000d</code>
<code>\ "</code>	Double apostrophe	<code>\u0022</code>
<code>\'</code>	Apostrophe	<code>\u0027</code>
<code>\\</code>	Antislash	<code>\u005C</code>



LES TYPES DE DONNEES

Initialisation des Booleens

```
boolean monBooleen = true;
```

```
boolean monBooleen = false;
```



LES TYPES DE DONNEES

Initialisation des Nombres réels

Exemples:

```
double monDouble = 12.35;
```

```
double monDouble = 45000;
```

```
float monFloat = 35.12f;
```

Le Séparateur décimal est le point .