A thick dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the text '2020/2021'.

2020/2021

Les chaînes de caractères

INFO 0101

An abstract graphic in the bottom left corner consisting of several thin, curved lines in shades of blue and grey, resembling stylized grass or reeds.

Quentin JUILLIARD
URCA

Table des matières

Chaîne de caractères	2
ASCII.....	2
ISO 8859-1	2
Unicode	3

Chaîne de caractères

Suite de caractères contenus dans une même variable.

À l'interne, les caractères peuvent être représentés selon divers codages comme ASCII, ISO 8859-1, Unicode.

ASCII

American Standard Code for Information Interchange

Le Codage sur 7 bits (128 caractères)

Il est Suffisant pour les américains, mais pas pour les français !

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	caractères de contrôle et divers non imprimables															
1x																
2x		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

ISO 8859-1

Extension de ASCII : codage sur 1 octet (8 bits) : 256 caractères.

Il encode les caractères de plusieurs langues européennes dont le français comme Espagnol, allemand, ...

D'autres extensions pour d'autres langues comme le 8859-2 (Tchèque, slovaque, polonais, ...), le 8859-3 (Turc, maltais, ...), etc...

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	caractères de contrôle et divers non imprimables															
1x																
2x		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8x	caractères de contrôle et divers non imprimables															
9x																
Ax		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯	
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Unicode

Le codage sur 4 octets (32 bits), l'UTF-8 : 1 suite de 1 à 4 octets, l'UTF-16 : 1 ou 2 suites de 2 octets et l'UTF-32 : 1 suite de 4 octets.

On peut coder tous les caractères de toutes les langues.

En voie de devenir le standard pour le codage des caractères comme son utilisation par Java.

Les chaînes de caractères en algorithmique

Notée entre guillemets par exemple "bonjour"

Contrairement à une variable caractère qui est noté entre apostrophes comme 'b'

Modifiables en contenu, mais pas en longueur

On dispose d'une fonction permettant d'obtenir la longueur d'une chaîne : longueur(chaine).

Les caractères dans une chaîne sont indicés de 0 à longueur(chaine) – 1, comme pour un tableau, accessibles par chaîne[ind]

Concaténation de chaînes par un opérateur + (comme en Java)

Pas d'opérateur de comparaison "="

Chaînes de caractères en java

Chaînes de caractères en Java

Type chaîne de caractères : String

Une chaîne de caractère n'est pas de type primitif (comme entier, réel, caractère et booléen) : c'est un objet

Déclaration d'une chaîne de caractères : `String ch; //Déclaration d'une référence à une chaîne`

Allocation et initialisation : `ch = "bonjour"; //Alloue et initialise une chaîne`

Équivalent à : `String ch = new String("Bonjour");`

Principe de fonctionnement

Par l'instruction : `ch = "bonjour" ;`

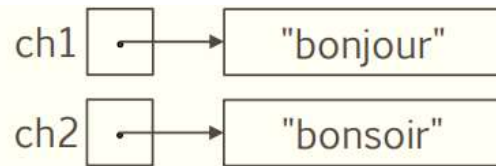
Un objet de type String est créé automatiquement par le compilateur :



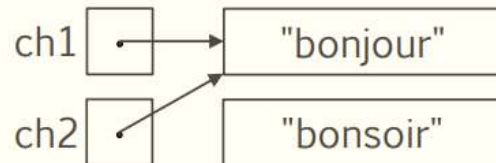
Un objet de type String n'est pas modifiable

Cependant, la référence sur la chaîne peut être modifiée :

```
String ch1 = "bonjour" ;
String ch2 = "bonsoir" ;
```



```
ch2 = ch1;
```



Entrées/sorties avec chaînes de caractères

Affichage

La méthode : `System.out.println("bonjour");`

Affiche une chaîne de caractères à l'écran. En réalité, la méthode reçoit une référence à une chaîne.

On peut donc passer une variable String à la méthode :

```
String ch = "bonjour";
System.out.println(ch);
```

Entrée Clavier

Classe Scanner => Méthode `nextLine()` : `String ch1 = clavier.nextLine()`

La méthode crée un objet de type chaîne. La référence obtenue est placée dans la variable ch1.

Longueur d'une chaîne

Méthode `length()` => Permet d'obtenir la longueur d'une chaîne, c'est-à-dire le nombre de

caractères :

```
String ch = "bonjour";
int longueur = ch.length(); //longueur contient 7
```

Remarque :

Cas des tableaux : `length` est un champ : `tab.length` (pas de parenthèses)

Cas des chaînes : c'est une méthode (fonction) : `ch.length()` (avec des parenthèses)

Concaténation de chaînes

Concaténation : fusion des caractères des deux chaînes.

Opérateur +

```
String ch1 = "Pierre";  
String ch2 = "Delisle";  
String ch3 = ch1 + ch2; //ch3 contient "PierreDelisle"
```

Lorsque l'opérateur + est utilisé dans la méthode System.out.println(), il désigne la concaténation et non l'addition arithmétique

Manipulation des caractères d'une chaîne

Accès aux caractères

Méthode charAt

Prend un entier en paramètre : indice du caractère voulu

```
String ch = "Pierre";  
System.out.println(ch.charAt(1)); //affiche 'i'
```

Recherche de caractères

Méthode indexOf prend un caractère ou une chaîne en paramètre et retourne l'indice de la première occurrence du caractère ou de la chaîne (-1 si non trouvé).

```
String ch = "Pierre";  
int n;  
n = ch.indexOf('i'); // n vaut 1  
n = ch.indexOf("erre"); // n vaut 2  
n = ch.indexOf('x'); // n vaut -1
```

Comparaison de chaînes

L'opérateur == compare les références, non les valeurs. 2 chaînes contenant la même valeur n'auront pas la même référence.

Il faut donc plutôt utiliser la méthode equals par exemple compare le contenu de 2 chaînes.

Prend une chaîne en paramètre

```
String ch1 = "Pierre";
String ch2 = "Pier";
ch2 = ch2 + "re";
if (ch1 == ch2)
    System.out.println("Chaines egales");
else
    System.out.println("Chaines differentes");
```

Affiche « chaines differentes »

```
String ch1 = "Pierre";
String ch2 = "Pier";
ch2 = ch2 + "re";
if (ch1.equals(ch2))
    System.out.println("Chaines egales");
else
    System.out.println("Chaines differentes");
```

Affiche « chaines egales »

La classe stringBuffer

StringBuffer

Pourquoi

Offre les fonctionnalités classiques de la classe String. Une chaîne codée en StringBuffer est toutefois modifiable. Il permet une gestion plus fine des chaînes de caractères.

Création d'une chaîne de type StringBuffer

En indiquant sa taille :

```
StringBuffer ch = new StringBuffer(20);
```

Crée une chaîne de 20 caractères

En indiquant une chaîne initiale :

```
StringBuffer ch = new StringBuffer("Pierre Delisle");
```

A partir d'un String :

```
String ch = "Pierre";
StringBuffer ch2 = new StringBuffer(ch);
```

Manipulation des chaînes de type StringBuffer

Ajouter des caractères dans la chaîne

méthode append()

- append(char caractère)
- append(String chaîne)
- append(StringBuffer chaîne)

```
StringBuffer ch = new StringBuffer(128);
ch.append("Pierre");
ch.append(' ');
String ch2 = "Delisle";
ch.append(ch2);
System.out.println(ch);
```

Connaître l'état d'une chaîne

`intcapacity()` qui retourne la capacité de la chaîne (le nombre de caractères qu'elle peut contenir)

`intlength()` qui retourne la longueur de la chaîne (le nombre de cases effectivement utilisées)

```
StringBuffer ch = new StringBuffer(128);
System.out.println(ch.length()); //retourne 0
ch.append("Pierre Delisle");
System.out.println(ch.capacity()); //retourne 128
System.out.println(ch.length()); //retourne 14
```

Accès aux caractères d'une chaîne de type StringBuffer

Récupérer un caractère

`char charAt(intindice)` qui retourne le caractère à la position *indice*

```
StringBuffer ch = new StringBuffer("Pierre");
char car = ch.charAt(1);
System.out.println(car); //affiche "i"
```

Modifier un caractère

`voidsetCharAt(intindice, char car)` qui modifie le caractère à la position *indice* par le caractère *car*

```
StringBuffer ch = new
StringBuffer("Pierret");
ch.setCharAt(5,'o');
System.out.println(ch); //affiche "Pierrot"
```


Insérer des caractères dans une chaîne

`insert(int indice, char car)` qui insère le caractère `car` à la position `indice`

`insert(int indice, String ch)` qui insère la chaîne `ch` à la position `indice`

```
StringBuffer ch = new StringBuffer("PD");  
ch.insert(1, "ierre");  
ch.insert(6, ' ');  
ch.insert(8, "elisle");  
System.out.println(ch);    //affiche "Pierre Delisle"
```