

Une vue sur les Objets Java et leur Organisation

type	
char	caractères
byte	Données binaires(Fichiers, images, flux d'entrée et de sortie du système)
short	Petits entiers
int	Entiers
long	Longs entiers
float	Nombres à virgules(flottante)
double	Nombres à virgules(flottante)double capacité
boolean	Vrai / faux

Mots clefs

Structures conditionnelles,
if,then, while, do,swith
Class, Interface, Abstract
Public, private, protected
Throw,throws,try, catch

Des outils primitifs uniquement sur les tableaux

```
int[][] tab = new int[2][3];
char[] tabChar = new char[5];
tabChar.
```

- clone(): char[] - char[]
- equals(Object obj): boolean - Object
- getClass(): Class<?> - Object
- hashCode(): int - Object
- notify(): void - Object
- notifyAll(): void - Object
- toString(): String - Object
- wait(): void - Object
- wait(long timeoutMillis): void - Object
- wait(long timeoutMillis, int nanos): void - Object
- length: int - char[]

Press 'Ctrl+Space' to show Template Proposals

Les classes enveloppes

type	classes enveloppes
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
boolean	Boolean

Package [java.lang](#)

Premier niveau d'outils pour manipuler nos données et variables.

```
char c = 'a';
c. aucune méthode outil accessible
Character c = 'a';
c.to
```

- toString(): String - Character
- toChars(int codePoint): char[] - Character
- toChars(int codePoint, char[] dst, int dstIndex): int - Character
- toCodePoint(char high, char low): int - Character
- toLowerCase(char ch): char - Character
- toLowerCase(int codePoint): int - Character
- toString(char c): String - Character
- toString(int codePoint): String - Character
- toUpperCase(char ch): char - Character
- toUpperCase(int codePoint): int - Character
- toUpperCase(char ch): char - Character

Press 'Ctrl+Space' to show Template Proposals

Converts the character argument to lowercase using case mapping information from the UnicodeData file.

Note that `Character.isLowerCase(Character.toLowerCase(ch))` does not always return true for some ranges of characters, particularly those that are symbols or ideographs.

In general, `String.toLowerCase()` should be used to map characters to lowercase. String case mapping methods have several benefits over Character case mapping methods. String case mapping methods can perform locale-sensitive mappings, context-sensitive mappings, and 1:M character mappings, whereas the Character case mapping methods cannot.

Note: This method cannot handle [supplementary characters](#). To support all Unicode characters, including supplementary characters, use the [toLowerCase](#)

Press 'Tab' from proposal table or click for focus

Des classes outils pour les MATHS

Math.

- round(float a): int - Math
- scalb(double d, int scaleFactor): double - Math
- scalb(float f, int scaleFactor): float - Math
- signum(double d): double - Math
- signum(float f): float - Math
- sin(double a): double - Math
- sinh(double x): double - Math

Collections

type *classes enveloppes*

ArrayList **Tableau linéaire**

HashMap **Clef-Valeur**

Stack **Pile**

Integer

etc

Des outils de gestions de groupes de données ou d'objets.
Manipulations simplifiées

ArrayList

```
add(String e) : boolean - ArrayList
add(int index, String element) : void - ArrayList
addAll(Collection<? extends String> c) : boolean - ArrayList
addAll(int index, Collection<? extends String> c) : boolean - ArrayList
clear() : void - ArrayList
clone() : Object - ArrayList
contains(Object o) : boolean - ArrayList
containsAll(Collection<?> c) : boolean - List
ensureCapacity(int minCapacity) : void - ArrayList
equals(Object o) : boolean - ArrayList
forEach(Consumer<? super String> action) : void - ArrayList
get(int index) : String - ArrayList
```

Press 'Ctrl+Space' to show Template Proposals

Performs the given action for each element of the Iterable until all element have been processed or the action throws an exception. Actions are performed in the order of iteration, if that order is specified. Exceptions thrown by the action are relayed to the caller.

The behavior of this method is unspecified if the action performs side-effects that modify the underlying source of elements, unless an overriding class has specified a concurrent modification policy.

Specified by: [forEach\(…\)](#) in [Iterable](#)

Parameters:

action The action to be performed for each element

Throws:

[NullPointerException](#) - if the specified action is null

Press 'Tab' from proposal table or click for focus

Boîtes à outils statiques des collections :

Arrays

```
sort(char[] a, int fromIndex, int toIndex) : void - java.util.Arrays
sort(double[] a, int fromIndex, int toIndex) : void - java.util.Arrays
sort(float[] a, int fromIndex, int toIndex) : void - java.util.Arrays
sort(int[] a, int fromIndex, int toIndex) : void - java.util.Arrays
sort(long[] a, int fromIndex, int toIndex) : void - java.util.Arrays
sort(Object[] a, int fromIndex, int toIndex) : void - java.util.Arrays
sort(short[] a, int fromIndex, int toIndex) : void - java.util.Arrays
sort(T[] a, int fromIndex, int toIndex, Comparator<? super T> c) : void - java.util.Arrays
parallelSort(byte[] a) : void - java.util.Arrays
parallelSort(char[] a) : void - java.util.Arrays
parallelSort(double[] a) : void - java.util.Arrays
```

Press 'Ctrl+Space' to show Template Proposals

Sorts the specified range of the array into ascending order. The range to be sorted extends from the index `fromIndex`, inclusive, to the index `toIndex`, exclusive. If `fromIndex == toIndex`, the range to be sorted is empty.

Parameters:

a the array to be sorted

`fromIndex` the index of the first element, inclusive, to be sorted

`toIndex` the index of the last element, exclusive, to be sorted

Throws:

[IllegalArgumentException](#) - if `fromIndex > toIndex`

[ArrayIndexOutOfBoundsException](#) - if `fromIndex < 0` or `toIndex > a.length`

Impl Note:

The sorting algorithm is a Dual-Pivot Quicksort by Vladimir Yaroslavskiy, Jon Bentley, and Joshua Bloch. This algorithm offers $O(n \log(n))$ performance on all data sets, and is typically faster than traditional

Press 'Tab' from proposal table or click for focus

Collections

```
list(Enumeration<T> e) : ArrayList<T> - java.util.Collections
max(Collection<? extends T> coll) : T - java.util.Collections
max(Collection<? extends T> coll, Comparator<? super T> comp) : T - java.util.Collections
min(Collection<? extends T> coll) : T - java.util.Collections
min(Collection<? extends T> coll, Comparator<? super T> comp) : T - java.util.Collections
nCopies(int n, T o) : List<T> - java.util.Collections
newSetFromMap(Map<E, Boolean> map) : Set<E> - java.util.Collections
replaceAll(List<T> list, T oldVal, T newVal) : boolean - java.util.Collections
reverse(List<T> list) : void - java.util.Collections
reverseOrder() : Comparator<T> - java.util.Collections
reverseOrder(Comparator<T> cmp) : Comparator<T> - java.util.Collections
```

Press 'Ctrl+Space' to show Template Proposals

Returns a comparator that imposes the reverse of the *natural ordering* on a collection of objects that implement the `Comparable` interface. (The natural ordering is the ordering imposed by the objects' own `compareTo` method.)

This enables a simple idiom for sorting (or maintaining) collections (or arrays) of objects that implement the `Comparable` interface in reverse-natural-order. For example, suppose `a` is an array of strings. Then:

```
Arrays.sort(a, Collections.reverseOrder());
```

sorts the array in reverse-lexicographic (alphabetical) order.

The returned comparator is serializable.

Type Parameters:

`T` the class of the objects compared by the comparator

Press 'Tab' from proposal table or click for focus

Package java.util

Des classes d'objets d'échanges avec le système et des données spécialisées

Class	
System	Flux entrées/sorties avec windows (clavier, console, paramètres...) Package java.lang
File	Flux entrées/sorties avec fichiers, texte ou image, bytes, mp3 etc Package java.io
Socket	Flux entrées/sorties à travers le réseau : Package java.net

Lectures séquentielles

Gestion par flux de données

Stream	Input /ouput
Buffer	Input /ouput
Printer	Input /ouput
Reader	Input /ouput

Package [java.util](#)

Des classes d'objets Spécialisées dans les flots d'executions des programmes

Class	Package java.lang
Thread	Au détails
ThreadGroup	Groupés
Executor	Des algorithmes de gestions de threads

Des classes d'objets qui s'apparentent à de petits logiciels

HttpServer	Package com.sun.net.httpserver Encapsulation des sockets, echange sur protocole http
	java.lang.instrument des class pour monitorer ses applications
Logger	Package java.util.logging gestion des logs dans une applications

JAVA des librairies différentes

Java SE

The Java Platform, Standard Edition (Java SE) APIs define the core Java platform for general-purpose computing.

JDK

The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be available in all implementations of the Java SE Platform. These APIs are in modules whose names start with `jdk`.

Bibliothèque

Une bibliothèque est une collection de routines ou de ressources précompilées fréquemment employées (classes, modèles, données de configuration, etc.), utilisable par un programme sur un ordinateur. Au lieu d'écrire le code de cette fonction à chaque exécution d'une tâche, vous pouvez utiliser des routines et les appeler lorsque la tâche doit être effectuée. Lorsqu'il a besoin d'une ressource, le programme appelle la bibliothèque correspondante.

Framework

Un framework peut être une collection de bibliothèques. À la différence des bibliothèques qui sont appelées par un programme, c'est le framework qui appelle une action. Le contrôle est inversé. Le framework contient déjà le modèle, ou la trame, des éléments nécessaires. Il peut donc appeler les ressources dont il a besoin (les trous à remplir dans le code) sur la base de ce qu'il contient déjà.

Serveur d'applications

Un serveur d'applications est un logiciel qui décrit un framework pour votre application. Il impose une structure pour vos fichiers (une arborescence de répertoires adaptée), des fichiers de configurations spécifiques, propose de l'outillage générique, spécifique, et permet l'utilisation en son sein d'autres framework (l'un spécialisé dans la communication avec les bases de données, l'autre sur la sécurité, etc etc etc.... les poupées russes encore et encore ! :D)

