

Apprenez à programmer en Java

40 heures 📊 Difficile

Mis à jour le 05/08/2019



Annexe A : liste des mots clés

Voici ici la liste des mots clés du langage Java, comme ceci vous ne serez pas tenté de nommez vos variables avec ces noms et ça vous permettra de retrouver facilement la définition d'un mot clé.

Les mots clés ▼

Cette annexe est une présentation des mots clés du langage Java et de leur signification. Ces mots clés ne peuvent être utilisés que pour quoi ils ont été créés. Ainsi, vous ne pouvez pas les utiliser comme noms de variables, noms de méthodes ou noms de classes.

À tout moment, cette liste de mots clés peut vous être utile. Si vous ne comprenez pas la signification de l'un d'eux, c'est que vous n'avez sans doute pas lu le chapitre dans lequel il se trouvait.

La signification de la liste de mots clés présentée dans cette annexe sera très courte pour chaque mot car il ne s'agit pas d'un chapitre de cours, mais plutôt de la présentation rapide de ceux-ci listés de façon à pouvoir retrouver rapidement celui que vous recherchez. Ainsi, vous n'aurez pas à rechercher dans tout le livre la signification d'un de ces mots.

Mots clés	Explications
abstract	Désigne une classe ou une méthode abstraite ce qui la rend impossible à instancier.
assert	Localise les erreurs internes du programme (à utiliser de préférence en phase de test).
boolean	Type booléen (valeurs possibles : <code>true</code> ou <code>false</code>).
break	Sort de la boucle en cours ou d'une instruction <code>switch</code> .
byte	Type entier (8 bits) : -128 à 127.

Mots clés	Explications
<code>case</code>	Clause d'une instruction <code>switch</code> .
<code>catch</code>	Clause d'un bloc <code>try/catch</code> interceptant un type d'exception.
<code>char</code>	Type caractère Unicode (16 bits).
<code>class</code>	Utilisé pour la définition d'une classe et ainsi créer vos propres objets.
<code>continue</code>	Passe à l'itération suivante d'une boucle sans effectuer les instructions situées après ce mot clé de l'itération en cours.
<code>default</code>	Cas par défaut d'une instruction <code>switch</code> .
<code>do</code>	Point d'entrée d'une boucle <code>do/while</code> (répéter / jusqu'à en algorithmique).
<code>double</code>	Type nombre flottant en double précision (64 bits).
<code>else</code>	Clause d'une instruction <code>if</code> .
<code>enum</code>	Nouveau type de classes depuis le JDK 1.5. Elle permet d'encapsuler une structure d'objet (comme une classe classique) mais aussi une liste d'instance limitant ainsi ses implémentation. Très utile pour des objets servants de paramètres pour des types de fonctionnement. (voir le chapitre consacré).
<code>extends</code>	Indique qu'une classe hérite d'une et une seule autre.
<code>false</code>	Valeur booléenne contraire à <code>true</code> .
<code>final</code>	<p>Appliqué à une variable, ce mot clé la transforme en constante, dont le contenu ne peut alors plus changer.</p> <p>Utilisé sur une méthode, celle-ci ne pourra plus être surchargée ou redéfinie dans une sous-classe.</p> <p>Sur une classe, elle ne pourra plus être dérivée par une autre classe (exemple la classe <code>String</code>).</p>
<code>finally</code>	Clause d'un bloc <code>try/catch</code> exécutée, qu'il y ait eu déclenchement d'une exception ou non.
<code>float</code>	Type nombre flottant en simple précision (32 bits).
<code>for</code>	Boucle dont le nombre de répétitions est connu (correspond à la boucle pour / fin pour en algorithmique).
<code>if</code>	Instruction conditionnelle (correspond au si / fin si en algorithmique).

Mots clés	Explications
<code>implements</code>	Indique la ou les classes implémentée(s) par une classe : s'utilise pour implémenter une interface.
<code>import</code>	Importe un package complet ou une classe présente dans un package.
<code>instanceof</code>	Permet de tester si un objet est une instance d'une classe.
<code>int</code>	Type entier (32 bits) : -2 147 483 648 à 2 147 483 647.
<code>interface</code>	Comparable à une classe mais de manière plus abstraite, une interface contient des méthodes devant être redéfinies obligatoirement par la classe implémentant l'interface.
<code>long</code>	Type entier (64 bits) : -9 223 372 036 854 775 808 à 9 223 372 036 854 775 807.
<code>native</code>	Indique que le code qui suit est écrit dans un autre langage que Java (solution non portable).
<code>new</code>	Instancie un nouvel objet ou un tableau.
<code>null</code>	Référence nulle : un objet ayant pour valeur <code>null</code> ne pointe vers aucun objet, il n'a aucune valeur.
<code>package</code>	Ensemble de classes et d'interfaces regroupés dans un ou plusieurs dossiers et sous-dossiers.
<code>private</code>	Utilisé sur une classe, il en restreint l'accès (uniquement utilisable pour les classes internes). Sur une méthode : on ne peut y accéder qu'à partir de la classe dans laquelle elle est définie. Appliqué à une variable : on ne peut y accéder qu'à partir de la classe dans laquelle elle est définie.
<code>protected</code>	Utilisé sur une classe, il en restreint l'accès à la classe contenant et ses classes filles (uniquement utilisable pour les classes internes). Sur une méthode : on ne peut y accéder qu'à partir de la classe dans laquelle elle est définie, par les classes filles ou le package dans lequel se trouve la classe contenant la méthode. Appliqué à une variable : on ne peut y accéder qu'à partir de la classe dans laquelle il est défini, par les classes filles ou le package dans lequel se trouve la classe contenant l'attribut.
<code>public</code>	Donne l'accessibilité à l'élément signé, méthode, variable ou classe par toutes les classes.

Mots clés

Explications

<code>return</code>	Utilisé dans une méthode pour retourner une valeur ou un objet.
<code>short</code>	Type entier (16 bits) : -32768 à 32767.
<code>static</code>	Appliqué à une variable : instancié une seule fois, l'attribut est commun à l'ensemble des instances d'une classe. Appliqué à une méthode : peut être appelée directement à partir du nom de la classe sans utiliser une instance de la classe.
<code>super</code>	Accède à la super-classe ou à son constructeur : permet d'exécuter le code présent dans la classe mère depuis la classe fille.
<code>switch</code>	Alternative à l'instruction <code>if</code> (les différents cas doivent être connus).
<code>synchronized</code>	Assure l'accès à une méthode ou à un bloc de code à un seul thread (les autres threads doivent alors attendre s'ils veulent accéder à cette partie du code).
<code>this</code>	Utilisé dans une classe pour indiquer l'objet courant.
<code>throw</code>	Lance une exception.
<code>throws</code>	Utilisé pour la déclaration des exceptions que peut lancer une méthode.
<code>transient</code>	Déclare une variable non sérialisable.
<code>true</code>	Valeur booléenne contraire à <code>false</code> .
<code>try</code>	Bloc de code susceptible d'intercepter une exception.
<code>void</code>	Utilisé pour les méthodes qui ne renvoient rien.
<code>volatile</code>	Permet de s'assurer que plusieurs <code>Thread</code> concurrent puissent accéder à la dernière valeur de la variable. Donc, une variable déclarée <code>volatile</code> est accessible par plusieurs <code>Thread</code> simultanément !
<code>while</code>	Boucle dans laquelle il est possible de ne jamais entrer (correspond à tant que / fin tant que en algorithmique).

☐ J'AI TERMINÉ CE CHAPITRE ET JE PASSE AU SUIVANT

◀ **ACTIVITÉ : RÉALISEZ UN GARAGE AVEC UNE
BASE DE DONNÉES EMBARQUÉE**

ANNEXE B : ECLIPSE

➤

Le professeur

Cyrille Herby

Spécialiste en développement Java et curieux insatiable d'informatique et de programmation web.
Actuellement auditeur en sécurité.

Découvrez aussi ce cours en...



Livre



PDF

OpenClassrooms

- L'entreprise
- Alternance
- Forum
- Blog
- Nous rejoindre

Entreprises

- Employeurs

En plus

- Devenez mentor
- Aide et FAQ
- Conditions Générales d'Utilisation
- Politique de Protection des Données Personnelles
- Nous contacter

 Français

▼



Télécharger dans
l'App Store