[**https://www.freecodecamp.org/news/java-string-to-int-how-to-convert-a-string-to-an-integer/**](https://www.freecodecamp.org/news/java-string-to-int-how-to-convert-a-string-to-an-integer/)

**Java String to Int – Comment convertir une chaîne en entier**

Les objets de chaîne sont représentés sous la forme d’une chaîne de caractères.

Si vous avez travaillé dans Java Swing, il a des composants tels que JTextField et JTextArea que nous utilisons pour obtenir notre entrée de l’interface graphique. Il prend notre entrée comme une chaîne.

Si nous voulons faire une calculatrice simple en utilisant Swing, nous devons comprendre comment convertir une chaîne en un entier. Cela nous amène à la question suivante: comment pouvons-nous convertir une chaîne en entier?

En Java, nous pouvons utiliser et convertir une chaîne en entier. Integer.valueOf()Integer.parseInt()

## 1. Utilisez Integer.parseInt() pour convertir une chaîne en entier

Cette méthode renvoie la chaîne sous la forme d’un **type primitif int**. Si la chaîne ne contient pas d’entier valide, elle lève une [Exception NumberFormatException](https://docs.oracle.com/javase/7/docs/api/java/lang/NumberFormatException.html).

Ainsi, chaque fois que nous convertissons une chaîne en int, nous devons prendre soin de cette exception en plaçant le code à l’intérieur du bloc try-catch.

Prenons un exemple de conversion d’une chaîne en int en utilisant :Integer.parseInt()

String str = "25";

try{

int number = Integer.parseInt(str);

System.out.println(number); // output = 25

}

catch (NumberFormatException ex){

ex.printStackTrace();

}

Essayons de casser ce code en entrant un entier non valide :

String str = "25T";

try{

int number = Integer.parseInt(str);

System.out.println(number);

}

catch (NumberFormatException ex){

ex.printStackTrace();

}

Comme vous pouvez le voir dans le code ci-dessus, nous avons essayé de convertir en entier. Il ne s’agit pas d’une entrée valide. Par conséquent, il doit lever une NumberFormatException. 25T

Voici la sortie du code ci-dessus:

java.lang.NumberFormatException: For input string: "25T"

at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)

at java.lang.Integer.parseInt(Integer.java:580)

at java.lang.Integer.parseInt(Integer.java:615)

at OOP.StringTest.main(StringTest.java:51)

Ensuite, nous allons examiner comment convertir une chaîne en entier à l’aide de la méthode. Integer.valueOf()

## 2. Utilisez Integer.valueOf() pour convertir une chaîne en entier

Cette méthode renvoie la chaîne sous la forme d’un **objet entier**. Si vous regardez la [documentation Java](https://docs.oracle.com/javase/7/docs/api/java/lang/Integer.html#valueOf(java.lang.String)), renvoie un objet entier qui est équivalent à un . Integer.valueOf()new Integer(Integer.parseInt(s))

Nous placerons notre code à l’intérieur du bloc try-catch lors de l’utilisation de cette méthode. Prenons un exemple en utilisant la méthode:Integer.valueOf()

String str = "25";

try{

Integer number = Integer.valueOf(str);

System.out.println(number); // output = 25

}

catch (NumberFormatException ex){

ex.printStackTrace();

}

Maintenant, essayons de casser le code ci-dessus en entrant un nombre entier non valide:

String str = "25TA";

try{

Integer number = Integer.valueOf(str);

System.out.println(number);

}

catch (NumberFormatException ex){

ex.printStackTrace();

}

Semblable à l’exemple précédent, le code ci-dessus lèvera une exception.

Voici la sortie du code ci-dessus:

java.lang.NumberFormatException: For input string: "25TA"

at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)

at java.lang.Integer.parseInt(Integer.java:580)

at java.lang.Integer.valueOf(Integer.java:766)

at OOP.StringTest.main(StringTest.java:42)

Nous pouvons également créer une méthode pour vérifier si la chaîne transmise est numérique ou non avant d’utiliser les méthodes mentionnées ci-dessus.

J’ai créé une méthode simple pour vérifier si la chaîne transmise est numérique ou non.

public class StringTest {

public static void main(String[] args) {

String str = "25";

String str1 = "25.06";

System.out.println(isNumeric(str));

System.out.println(isNumeric(str1));

}

private static boolean isNumeric(String str){

return str != null && str.matches("[0-9.]+");

}

}

La sortie est la suivante :

true

true

La méthode prend une chaîne comme argument. Tout d’abord, il vérifie si c’est le cas ou non. Après cela, nous utilisons la méthode pour vérifier si elle contient des chiffres de 0 à 9 et un caractère de point. isNumeric()nullmatches()

Il s’agit d’un moyen simple de vérifier les valeurs numériques. Vous pouvez écrire ou rechercher dans Google des expressions régulières plus avancées pour capturer des chiffres en fonction de votre cas d’utilisation.

Il est recommandé de vérifier si la chaîne transmise est numérique ou non avant d’essayer de la convertir en entier.

Merci d’avoir lu.

Publier l’image par [🇸🇮 Janko Ferlič](https://unsplash.com/@itfeelslikefilm?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText) sur [Unsplash](https://unsplash.com/collections/139346/soul-care?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText)

Vous pouvez vous connecter avec moi sur [Medium](https://mvthanoshan.medium.com/).

**Bon codage!**

J’ai créé une méthode simple pour vérifier si la chaîne transmise est numérique ou non.

public class StringTest {

public static void main(String[] args) {

String str = "25";

String str1 = "25.06";

System.out.println(isNumeric(str));

System.out.println(isNumeric(str1));

}

private static boolean isNumeric(String str){

return str != null && str.matches("[0-9.]+");

}

}

La sortie est la suivante :

true

true

La méthode prend une chaîne comme argument. Tout d’abord, il vérifie si c’est le cas ou non. Après cela, nous utilisons la méthode pour vérifier si elle contient des chiffres de 0 à 9 et un caractère de point. isNumeric()nullmatches()

Il s’agit d’un moyen simple de vérifier les valeurs numériques. Vous pouvez écrire ou rechercher dans Google des expressions régulières plus avancées pour capturer des chiffres en fonction de votre cas d’utilisation.

Il est recommandé de vérifier si la chaîne transmise est numérique ou non avant d’essayer de la convertir en entier.

Merci d’avoir lu.