

JAVA.LANG.BYTE.PARSEBYTE METHOD

http://www.tutorialspoint.com/java/lang/byte_parsebyte_radix.htm

Copyright © tutorialspoint.com

Description

The **java.lang.Byte.parseByte** method, *inradix* parses the string argument as a signed byte in the radix specified by the second argument.

The characters in the string must all be digits, of the specified radix *as determined by whether Character.digit(char, int) returns a nonnegative value*) except that the first character may be an ASCII minus sign '-' '\u002D' to indicate a negative value or an ASCII plus sign '+' '\u002B' to indicate a positive value. The resulting byte value is returned.

An exception of type **NumberFormatException** is thrown if any of the following situations occurs:

- The first argument is null or is a string of length zero.
- The radix is either smaller than **Character.MIN_RADIX** or larger than **Character.MAX_RADIX**.
- Any character of the string is not a digit of the specified radix, except that the first character may be a minus sign '-' '\u002D' or plus sign '+' '\u002B' provided that the string is longer than length 1.
- The value represented by the string is not a value of type byte.

Declaration

Following is the declaration for **java.lang.Byte.parseByte** method

```
public static byte parseByte(String s, int radix)
                        throws NumberFormatException
```

Parameters

- **s** - a String containing the byte representation to be parsed
- **radix** - the radix to be used while parsing s

Return Value

This method returns the byte value represented by the string argument in the specified radix.

Exception

- **NumberFormatException** - if the string does not contain a parsable byte.

Example

The following example shows the usage of **lang.Byte.parseByte** method.

```
package com.tutorialspoint;

import java.lang.*;

public class ByteDemo {

    public static void main(String[] args) {

        // create 2 byte primitives bt1, bt2
        byte bt1, bt2;

        // create and assign values to String's s1, s2
        String s1 = "123";
```

```

String s2 = "-1a";

// create and assign values to int r1, r2
int r1 = 8; // represents octal
int r2 = 16; // represents hexadecimal

/**
 * static method is called using class name. Assign parseByte
 * result on s1, s2 to bt1, bt2 using radix r1, r2
 */
bt1 = Byte.parseByte(s1, r1);
bt2 = Byte.parseByte(s2, r2);

String str1 = "Parse byte value of " + s1 + " is " + bt1;
String str2 = "Parse byte value of " + s2 + " is " + bt2;

// print bt1, bt2 values
System.out.println( str1 );
System.out.println( str2 );
}
}

```

Let us compile and run the above program, this will produce the following result:

```

Parse byte value of 123 is 83
Parse byte value of -1a is -26

```

Loading [MathJax]/jax/output/HTML-CSS/jax.js