

JAVA.LANG.CHARACTER.TOUPPERCASE METHOD

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

Copyright © tutorialspoint.com

Description

The **java.lang.Character.toUpperCaseintcodePoint** converts the character *Unicodecodepoint* argument to uppercase using case mapping information from the UnicodeData file.

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

Declaration

Following is the declaration for **java.lang.Character.toUpperCase** method

```
public static int toUpperCase(int codePoint)
```

Parameters

- **codePoint** - the character *Unicodecodepoint* to be converted

Return Value

This method returns the uppercase equivalent of the character, if any; otherwise, the character itself.

Exception

- **NA**

Example

The following example shows the usage of `lang.Character.toUpperCase` method.

```
package com.tutorialspoint;

import java.lang.*;

public class CharacterDemo {

    public static void main(String[] args) {

        // create 4 int primitives
        int cp1, cp2, cp3, cp4;

        // assign values to cp1, cp2
        cp1 = 0x0072; // represents r
        cp2 = 0x0569; // represents ARMENIAN SMALL LETTER TO

        // assign uppercase of cp1, cp2 to cp3, cp4
        cp3 = Character.toUpperCase(cp1);
        cp4 = Character.toUpperCase(cp2);

        String str1 = "Uppercase equivalent of " + cp1 + " is " + cp3;
        String str2 = "Uppercase equivalent of " + cp2 + " is " + cp4;

        // print cp3, cp4 values
        System.out.println( str1 );
        System.out.println( str2 );
    }
}
```

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

Uppercase equivalent of 114 is 82
Uppercase equivalent of 1385 is 1337

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