

JAVA.LANG.CHARACTER.DIGIT METHOD

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

Copyright © tutorialspoint.com

Description

The **java.lang.Character.digit**(*int* codePoint, *int* radix) returns the numeric value of the specified character *Unicodecodepoint* in the specified radix.

If the radix is not in the range $\text{MIN_RADIX} \leq \text{radix} \leq \text{MAX_RADIX}$ or if the character is not a valid digit in the specified radix, -1 is returned. A character is a valid digit if at least one of the following is true:

- The method *isDigitcodePoint* is true of the character and the Unicode decimal digit value of the character *oritssingle – characterdecomposition* is less than the specified radix. In this case the decimal digit value is returned.
- The character is one of the uppercase Latin letters 'A' through 'Z' and its code is less than $\text{radix} + \text{'A'}$. In this case, $\text{codePoint} - \text{'A'}$ is returned.
- The character is one of the lowercase Latin letters 'a' through 'z' and its code is less than $\text{radix} + \text{'a'}$. In this case, $\text{codePoint} - \text{'a'}$ is returned.
- The character is one of the fullwidth uppercase Latin letters A '\uFF21' through Z '\uFF3A' and its code is less than $\text{radix} + \text{'\uFF21'}$. In this case, $\text{codePoint} - \text{'\uFF21'}$ is returned.
- The character is one of the fullwidth lowercase Latin letters a '\uFF41' through z '\uFF5A' and its code is less than $\text{radix} + \text{'\uFF41'}$. In this case, $\text{codePoint} - \text{'\uFF41'}$ is returned.

Declaration

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

```
public static int digit(int codePoint, int radix)
```

Parameters

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

Return Value

This method returns the numeric value represented by the character in the specified radix.

Exception

- **NA**

Example

The following example shows the usage of lang.Character.digit method.

```
package com.tutorialspoint;

import java.lang.*;

public class CharacterDemo {

    public static void main(String[] args) {

        // create 2 int primitives cp1, cp2
```

```

int cp1, cp2;

// assign values to cp1, cp2
cp1 = 0x0034;
cp2 = 0x004a;

// create 2 int primitives i1, i2
int i1, i2;

// assign numeric value of cp1, cp2 to i1, i2 using radix
i1 = Character.digit(cp1, 8);
i2 = Character.digit(cp2, 8);

String str1 = "Numeric value of cp1 in radix 8 is " + i1;
String str2 = "Numeric value of cp2 in radix 8 is " + i2;

// print i1, i2 values
System.out.println( str1 );
System.out.println( str2 );
}
}

```

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

```

Numeric value of cp1 in radix 8 is 4
Numeric value of cp2 in radix 8 is -1

```

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