

JAVA.IO.PIPEDINPUTSTREAM.READ METHOD

http://www.tutorialspoint.com/java/io/pipedinputstream_read_byte.htm

Copyright © tutorialspoint.com

Description

The **java.io.PipedInputStream.read***byte[] b, int off, int len* method reads up to len bytes of data from this piped input stream into an array of bytes. Less than len bytes will be read if the end of the data stream is reached or if len exceeds the pipe's buffer size. If len is zero, then no bytes are read and 0 is returned; otherwise, the method blocks until at least 1 byte of input is available, end of the stream has been detected, or an exception is thrown.

Declaration

Following is the declaration for **java.io.PipedInputStream.read** method.

```
public int read(byte[] b, int off, int len)
```

Parameters

- **b** -- the buffer into which the data is read.
- **off** -- the start offset in the destination array b
- **len** -- the maximum number of bytes read.

Return Value

This method returns the total number of bytes read into the buffer, or -1 if there is no more data because the end of the stream has been reached.

Exception

- **NullPointerException** -- If b is null.
- **IndexOutOfBoundsException** -- If off is negative, len is negative, or len is greater than b.length - off
- **IOException** -- if the pipe is broken, unconnected, closed, or if an I/O error occurs.

Example

The following example shows the usage of **java.io.PipedInputStream.read** method.

```
package com.tutorialspoint;

import java.io.*;

public class PipedInputStreamDemo {

    public static void main(String[] args) {

        // create a new Piped input and Output Stream
        PipedOutputStream out = new PipedOutputStream();
        PipedInputStream in = new PipedInputStream();

        try {
            // connect input and output
            in.connect(out);

            // write something
            out.write(70);
            out.write(71);

            // read what we wrote into an array of bytes
```

```
byte[] b = new byte[2];
in.read(b, 0, 2);

// print the array as a string
String s = new String(b);
System.out.println("'" + s);

} catch (IOException ex) {
    ex.printStackTrace();
}

}
```

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

FG

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