| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PrintStream.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/io/PipedWriter.html)   [**NEXT CLASS**](http://docs.google.com/java/io/PrintWriter.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/PrintStream.html)    [**NO FRAMES**](http://docs.google.com/PrintStream.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#2jxsxqh) |

## **java.io**

Class PrintStream

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.io.OutputStream](http://docs.google.com/java/io/OutputStream.html)  
 [java.io.FilterOutputStream](http://docs.google.com/java/io/FilterOutputStream.html)  
 **java.io.PrintStream**

**All Implemented Interfaces:** [Closeable](http://docs.google.com/java/io/Closeable.html), [Flushable](http://docs.google.com/java/io/Flushable.html), [Appendable](http://docs.google.com/java/lang/Appendable.html) **Direct Known Subclasses:** [LogStream](http://docs.google.com/java/rmi/server/LogStream.html)

public class **PrintStream**extends [FilterOutputStream](http://docs.google.com/java/io/FilterOutputStream.html)implements [Appendable](http://docs.google.com/java/lang/Appendable.html), [Closeable](http://docs.google.com/java/io/Closeable.html)

A PrintStream adds functionality to another output stream, namely the ability to print representations of various data values conveniently. Two other features are provided as well. Unlike other output streams, a PrintStream never throws an IOException; instead, exceptional situations merely set an internal flag that can be tested via the checkError method. Optionally, a PrintStream can be created so as to flush automatically; this means that the flush method is automatically invoked after a byte array is written, one of the println methods is invoked, or a newline character or byte ('\n') is written.

All characters printed by a PrintStream are converted into bytes using the platform's default character encoding. The [PrintWriter](http://docs.google.com/java/io/PrintWriter.html) class should be used in situations that require writing characters rather than bytes.

**Since:** JDK1.0

| **Field Summary** | |
| --- | --- |

| **Fields inherited from class java.io.**[**FilterOutputStream**](http://docs.google.com/java/io/FilterOutputStream.html) |
| --- |
| [out](http://docs.google.com/java/io/FilterOutputStream.html#out) |

| **Constructor Summary** | |
| --- | --- |
| [**PrintStream**](http://docs.google.com/java/io/PrintStream.html#PrintStream(java.io.File))([File](http://docs.google.com/java/io/File.html) file)            Creates a new print stream, without automatic line flushing, with the specified file. |
| [**PrintStream**](http://docs.google.com/java/io/PrintStream.html#PrintStream(java.io.File,%20java.lang.String))([File](http://docs.google.com/java/io/File.html) file, [String](http://docs.google.com/java/lang/String.html) csn)            Creates a new print stream, without automatic line flushing, with the specified file and charset. |
| [**PrintStream**](http://docs.google.com/java/io/PrintStream.html#PrintStream(java.io.OutputStream))([OutputStream](http://docs.google.com/java/io/OutputStream.html) out)            Creates a new print stream. |
| [**PrintStream**](http://docs.google.com/java/io/PrintStream.html#PrintStream(java.io.OutputStream,%20boolean))([OutputStream](http://docs.google.com/java/io/OutputStream.html) out, boolean autoFlush)            Creates a new print stream. |
| [**PrintStream**](http://docs.google.com/java/io/PrintStream.html#PrintStream(java.io.OutputStream,%20boolean,%20java.lang.String))([OutputStream](http://docs.google.com/java/io/OutputStream.html) out, boolean autoFlush, [String](http://docs.google.com/java/lang/String.html) encoding)            Creates a new print stream. |
| [**PrintStream**](http://docs.google.com/java/io/PrintStream.html#PrintStream(java.lang.String))([String](http://docs.google.com/java/lang/String.html) fileName)            Creates a new print stream, without automatic line flushing, with the specified file name. |
| [**PrintStream**](http://docs.google.com/java/io/PrintStream.html#PrintStream(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) fileName, [String](http://docs.google.com/java/lang/String.html) csn)            Creates a new print stream, without automatic line flushing, with the specified file name and charset. |

| **Method Summary** | |
| --- | --- |
| [PrintStream](http://docs.google.com/java/io/PrintStream.html) | [**append**](http://docs.google.com/java/io/PrintStream.html#append(char))(char c)            Appends the specified character to this output stream. |
| [PrintStream](http://docs.google.com/java/io/PrintStream.html) | [**append**](http://docs.google.com/java/io/PrintStream.html#append(java.lang.CharSequence))([CharSequence](http://docs.google.com/java/lang/CharSequence.html) csq)            Appends the specified character sequence to this output stream. |
| [PrintStream](http://docs.google.com/java/io/PrintStream.html) | [**append**](http://docs.google.com/java/io/PrintStream.html#append(java.lang.CharSequence,%20int,%20int))([CharSequence](http://docs.google.com/java/lang/CharSequence.html) csq, int start, int end)            Appends a subsequence of the specified character sequence to this output stream. |
| boolean | [**checkError**](http://docs.google.com/java/io/PrintStream.html#checkError())()            Flushes the stream and checks its error state. |
| protected  void | [**clearError**](http://docs.google.com/java/io/PrintStream.html#clearError())()            Clears the internal error state of this stream. |
| void | [**close**](http://docs.google.com/java/io/PrintStream.html#close())()            Closes the stream. |
| void | [**flush**](http://docs.google.com/java/io/PrintStream.html#flush())()            Flushes the stream. |
| [PrintStream](http://docs.google.com/java/io/PrintStream.html) | [**format**](http://docs.google.com/java/io/PrintStream.html#format(java.util.Locale,%20java.lang.String,%20java.lang.Object...))([Locale](http://docs.google.com/java/util/Locale.html) l, [String](http://docs.google.com/java/lang/String.html) format, [Object](http://docs.google.com/java/lang/Object.html)... args)            Writes a formatted string to this output stream using the specified format string and arguments. |
| [PrintStream](http://docs.google.com/java/io/PrintStream.html) | [**format**](http://docs.google.com/java/io/PrintStream.html#format(java.lang.String,%20java.lang.Object...))([String](http://docs.google.com/java/lang/String.html) format, [Object](http://docs.google.com/java/lang/Object.html)... args)            Writes a formatted string to this output stream using the specified format string and arguments. |
| void | [**print**](http://docs.google.com/java/io/PrintStream.html#print(boolean))(boolean b)            Prints a boolean value. |
| void | [**print**](http://docs.google.com/java/io/PrintStream.html#print(char))(char c)            Prints a character. |
| void | [**print**](http://docs.google.com/java/io/PrintStream.html#print(char%5B%5D))(char[] s)            Prints an array of characters. |
| void | [**print**](http://docs.google.com/java/io/PrintStream.html#print(double))(double d)            Prints a double-precision floating-point number. |
| void | [**print**](http://docs.google.com/java/io/PrintStream.html#print(float))(float f)            Prints a floating-point number. |
| void | [**print**](http://docs.google.com/java/io/PrintStream.html#print(int))(int i)            Prints an integer. |
| void | [**print**](http://docs.google.com/java/io/PrintStream.html#print(long))(long l)            Prints a long integer. |
| void | [**print**](http://docs.google.com/java/io/PrintStream.html#print(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Prints an object. |
| void | [**print**](http://docs.google.com/java/io/PrintStream.html#print(java.lang.String))([String](http://docs.google.com/java/lang/String.html) s)            Prints a string. |
| [PrintStream](http://docs.google.com/java/io/PrintStream.html) | [**printf**](http://docs.google.com/java/io/PrintStream.html#printf(java.util.Locale,%20java.lang.String,%20java.lang.Object...))([Locale](http://docs.google.com/java/util/Locale.html) l, [String](http://docs.google.com/java/lang/String.html) format, [Object](http://docs.google.com/java/lang/Object.html)... args)            A convenience method to write a formatted string to this output stream using the specified format string and arguments. |
| [PrintStream](http://docs.google.com/java/io/PrintStream.html) | [**printf**](http://docs.google.com/java/io/PrintStream.html#printf(java.lang.String,%20java.lang.Object...))([String](http://docs.google.com/java/lang/String.html) format, [Object](http://docs.google.com/java/lang/Object.html)... args)            A convenience method to write a formatted string to this output stream using the specified format string and arguments. |
| void | [**println**](http://docs.google.com/java/io/PrintStream.html#println())()            Terminates the current line by writing the line separator string. |
| void | [**println**](http://docs.google.com/java/io/PrintStream.html#println(boolean))(boolean x)            Prints a boolean and then terminate the line. |
| void | [**println**](http://docs.google.com/java/io/PrintStream.html#println(char))(char x)            Prints a character and then terminate the line. |
| void | [**println**](http://docs.google.com/java/io/PrintStream.html#println(char%5B%5D))(char[] x)            Prints an array of characters and then terminate the line. |
| void | [**println**](http://docs.google.com/java/io/PrintStream.html#println(double))(double x)            Prints a double and then terminate the line. |
| void | [**println**](http://docs.google.com/java/io/PrintStream.html#println(float))(float x)            Prints a float and then terminate the line. |
| void | [**println**](http://docs.google.com/java/io/PrintStream.html#println(int))(int x)            Prints an integer and then terminate the line. |
| void | [**println**](http://docs.google.com/java/io/PrintStream.html#println(long))(long x)            Prints a long and then terminate the line. |
| void | [**println**](http://docs.google.com/java/io/PrintStream.html#println(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) x)            Prints an Object and then terminate the line. |
| void | [**println**](http://docs.google.com/java/io/PrintStream.html#println(java.lang.String))([String](http://docs.google.com/java/lang/String.html) x)            Prints a String and then terminate the line. |
| protected  void | [**setError**](http://docs.google.com/java/io/PrintStream.html#setError())()            Sets the error state of the stream to true. |
| void | [**write**](http://docs.google.com/java/io/PrintStream.html#write(byte%5B%5D,%20int,%20int))(byte[] buf, int off, int len)            Writes len bytes from the specified byte array starting at offset off to this stream. |
| void | [**write**](http://docs.google.com/java/io/PrintStream.html#write(int))(int b)            Writes the specified byte to this stream. |

| **Methods inherited from class java.io.**[**FilterOutputStream**](http://docs.google.com/java/io/FilterOutputStream.html) |
| --- |
| [write](http://docs.google.com/java/io/FilterOutputStream.html#write(byte%5B%5D)) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### PrintStream

public **PrintStream**([OutputStream](http://docs.google.com/java/io/OutputStream.html) out)

Creates a new print stream. This stream will not flush automatically.

**Parameters:**out - The output stream to which values and objects will be printed**See Also:**[PrintWriter.PrintWriter(java.io.OutputStream)](http://docs.google.com/java/io/PrintWriter.html#PrintWriter(java.io.OutputStream))

### PrintStream

public **PrintStream**([OutputStream](http://docs.google.com/java/io/OutputStream.html) out,  
 boolean autoFlush)

Creates a new print stream.

**Parameters:**out - The output stream to which values and objects will be printedautoFlush - A boolean; if true, the output buffer will be flushed whenever a byte array is written, one of the println methods is invoked, or a newline character or byte ('\n') is written**See Also:**[PrintWriter.PrintWriter(java.io.OutputStream, boolean)](http://docs.google.com/java/io/PrintWriter.html#PrintWriter(java.io.OutputStream,%20boolean))

### PrintStream

public **PrintStream**([OutputStream](http://docs.google.com/java/io/OutputStream.html) out,  
 boolean autoFlush,  
 [String](http://docs.google.com/java/lang/String.html) encoding)  
 throws [UnsupportedEncodingException](http://docs.google.com/java/io/UnsupportedEncodingException.html)

Creates a new print stream.

**Parameters:**out - The output stream to which values and objects will be printedautoFlush - A boolean; if true, the output buffer will be flushed whenever a byte array is written, one of the println methods is invoked, or a newline character or byte ('\n') is writtenencoding - The name of a supported  [character encoding](http://docs.google.com/lang/package-summary.html#charenc) **Throws:** [UnsupportedEncodingException](http://docs.google.com/java/io/UnsupportedEncodingException.html) - If the named encoding is not supported**Since:** 1.4

### PrintStream

public **PrintStream**([String](http://docs.google.com/java/lang/String.html) fileName)  
 throws [FileNotFoundException](http://docs.google.com/java/io/FileNotFoundException.html)

Creates a new print stream, without automatic line flushing, with the specified file name. This convenience constructor creates the necessary intermediate [OutputStreamWriter](http://docs.google.com/java/io/OutputStreamWriter.html), which will encode characters using the [default charset](http://docs.google.com/java/nio/charset/Charset.html#defaultCharset()) for this instance of the Java virtual machine.

**Parameters:**fileName - The name of the file to use as the destination of this print stream. If the file exists, then it will be truncated to zero size; otherwise, a new file will be created. The output will be written to the file and is buffered. **Throws:** [FileNotFoundException](http://docs.google.com/java/io/FileNotFoundException.html) - If the given file object does not denote an existing, writable regular file and a new regular file of that name cannot be created, or if some other error occurs while opening or creating the file [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - If a security manager is present and [checkWrite(fileName)](http://docs.google.com/java/lang/SecurityManager.html#checkWrite(java.io.FileDescriptor)) denies write access to the file**Since:** 1.5

### PrintStream

public **PrintStream**([String](http://docs.google.com/java/lang/String.html) fileName,  
 [String](http://docs.google.com/java/lang/String.html) csn)  
 throws [FileNotFoundException](http://docs.google.com/java/io/FileNotFoundException.html),  
 [UnsupportedEncodingException](http://docs.google.com/java/io/UnsupportedEncodingException.html)

Creates a new print stream, without automatic line flushing, with the specified file name and charset. This convenience constructor creates the necessary intermediate [OutputStreamWriter](http://docs.google.com/java/io/OutputStreamWriter.html), which will encode characters using the provided charset.

**Parameters:**fileName - The name of the file to use as the destination of this print stream. If the file exists, then it will be truncated to zero size; otherwise, a new file will be created. The output will be written to the file and is buffered.csn - The name of a supported [charset](http://docs.google.com/java/nio/charset/Charset.html) **Throws:** [FileNotFoundException](http://docs.google.com/java/io/FileNotFoundException.html) - If the given file object does not denote an existing, writable regular file and a new regular file of that name cannot be created, or if some other error occurs while opening or creating the file [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - If a security manager is present and [checkWrite(fileName)](http://docs.google.com/java/lang/SecurityManager.html#checkWrite(java.io.FileDescriptor)) denies write access to the file [UnsupportedEncodingException](http://docs.google.com/java/io/UnsupportedEncodingException.html) - If the named charset is not supported**Since:** 1.5

### PrintStream

public **PrintStream**([File](http://docs.google.com/java/io/File.html) file)  
 throws [FileNotFoundException](http://docs.google.com/java/io/FileNotFoundException.html)

Creates a new print stream, without automatic line flushing, with the specified file. This convenience constructor creates the necessary intermediate [OutputStreamWriter](http://docs.google.com/java/io/OutputStreamWriter.html), which will encode characters using the [default charset](http://docs.google.com/java/nio/charset/Charset.html#defaultCharset()) for this instance of the Java virtual machine.

**Parameters:**file - The file to use as the destination of this print stream. If the file exists, then it will be truncated to zero size; otherwise, a new file will be created. The output will be written to the file and is buffered. **Throws:** [FileNotFoundException](http://docs.google.com/java/io/FileNotFoundException.html) - If the given file object does not denote an existing, writable regular file and a new regular file of that name cannot be created, or if some other error occurs while opening or creating the file [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - If a security manager is present and [checkWrite(file.getPath())](http://docs.google.com/java/lang/SecurityManager.html#checkWrite(java.io.FileDescriptor)) denies write access to the file**Since:** 1.5

### PrintStream

public **PrintStream**([File](http://docs.google.com/java/io/File.html) file,  
 [String](http://docs.google.com/java/lang/String.html) csn)  
 throws [FileNotFoundException](http://docs.google.com/java/io/FileNotFoundException.html),  
 [UnsupportedEncodingException](http://docs.google.com/java/io/UnsupportedEncodingException.html)

Creates a new print stream, without automatic line flushing, with the specified file and charset. This convenience constructor creates the necessary intermediate [OutputStreamWriter](http://docs.google.com/java/io/OutputStreamWriter.html), which will encode characters using the provided charset.

**Parameters:**file - The file to use as the destination of this print stream. If the file exists, then it will be truncated to zero size; otherwise, a new file will be created. The output will be written to the file and is buffered.csn - The name of a supported [charset](http://docs.google.com/java/nio/charset/Charset.html) **Throws:** [FileNotFoundException](http://docs.google.com/java/io/FileNotFoundException.html) - If the given file object does not denote an existing, writable regular file and a new regular file of that name cannot be created, or if some other error occurs while opening or creating the file [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - If a security manager is presentand [checkWrite(file.getPath())](http://docs.google.com/java/lang/SecurityManager.html#checkWrite(java.io.FileDescriptor)) denies write access to the file [UnsupportedEncodingException](http://docs.google.com/java/io/UnsupportedEncodingException.html) - If the named charset is not supported**Since:** 1.5

| **Method Detail** |
| --- |

### flush

public void **flush**()

Flushes the stream. This is done by writing any buffered output bytes to the underlying output stream and then flushing that stream.

**Specified by:**[flush](http://docs.google.com/java/io/Flushable.html#flush()) in interface [Flushable](http://docs.google.com/java/io/Flushable.html)**Overrides:**[flush](http://docs.google.com/java/io/FilterOutputStream.html#flush()) in class [FilterOutputStream](http://docs.google.com/java/io/FilterOutputStream.html) **See Also:**[OutputStream.flush()](http://docs.google.com/java/io/OutputStream.html#flush())

### close

public void **close**()

Closes the stream. This is done by flushing the stream and then closing the underlying output stream.

**Specified by:**[close](http://docs.google.com/java/io/Closeable.html#close()) in interface [Closeable](http://docs.google.com/java/io/Closeable.html)**Overrides:**[close](http://docs.google.com/java/io/FilterOutputStream.html#close()) in class [FilterOutputStream](http://docs.google.com/java/io/FilterOutputStream.html) **See Also:**[OutputStream.close()](http://docs.google.com/java/io/OutputStream.html#close())

### checkError

public boolean **checkError**()

Flushes the stream and checks its error state. The internal error state is set to true when the underlying output stream throws an IOException other than InterruptedIOException, and when the setError method is invoked. If an operation on the underlying output stream throws an InterruptedIOException, then the PrintStream converts the exception back into an interrupt by doing:

Thread.currentThread().interrupt();

or the equivalent.

**Returns:**true if and only if this stream has encountered an IOException other than InterruptedIOException, or the setError method has been invoked

### setError

protected void **setError**()

Sets the error state of the stream to true.

This method will cause subsequent invocations of [checkError()](http://docs.google.com/java/io/PrintStream.html#checkError()) to return true until [clearError()](http://docs.google.com/java/io/PrintStream.html#clearError()) is invoked.

**Since:** JDK1.1

### clearError

protected void **clearError**()

Clears the internal error state of this stream.

This method will cause subsequent invocations of [checkError()](http://docs.google.com/java/io/PrintStream.html#checkError()) to return false until another write operation fails and invokes [setError()](http://docs.google.com/java/io/PrintStream.html#setError()).

**Since:** 1.6

### write

public void **write**(int b)

Writes the specified byte to this stream. If the byte is a newline and automatic flushing is enabled then the flush method will be invoked.

Note that the byte is written as given; to write a character that will be translated according to the platform's default character encoding, use the print(char) or println(char) methods.

**Overrides:**[write](http://docs.google.com/java/io/FilterOutputStream.html#write(int)) in class [FilterOutputStream](http://docs.google.com/java/io/FilterOutputStream.html) **Parameters:**b - The byte to be written**See Also:**[print(char)](http://docs.google.com/java/io/PrintStream.html#print(char)), [println(char)](http://docs.google.com/java/io/PrintStream.html#println(char))

### write

public void **write**(byte[] buf,  
 int off,  
 int len)

Writes len bytes from the specified byte array starting at offset off to this stream. If automatic flushing is enabled then the flush method will be invoked.

Note that the bytes will be written as given; to write characters that will be translated according to the platform's default character encoding, use the print(char) or println(char) methods.

**Overrides:**[write](http://docs.google.com/java/io/FilterOutputStream.html#write(byte%5B%5D,%20int,%20int)) in class [FilterOutputStream](http://docs.google.com/java/io/FilterOutputStream.html) **Parameters:**buf - A byte arrayoff - Offset from which to start taking byteslen - Number of bytes to write**See Also:**[FilterOutputStream.write(int)](http://docs.google.com/java/io/FilterOutputStream.html#write(int))

### print

public void **print**(boolean b)

Prints a boolean value. The string produced by [String.valueOf(boolean)](http://docs.google.com/java/lang/String.html#valueOf(boolean)) is translated into bytes according to the platform's default character encoding, and these bytes are written in exactly the manner of the [write(int)](http://docs.google.com/java/io/PrintStream.html#write(int)) method.

**Parameters:**b - The boolean to be printed

### print

public void **print**(char c)

Prints a character. The character is translated into one or more bytes according to the platform's default character encoding, and these bytes are written in exactly the manner of the [write(int)](http://docs.google.com/java/io/PrintStream.html#write(int)) method.

**Parameters:**c - The char to be printed

### print

public void **print**(int i)

Prints an integer. The string produced by [String.valueOf(int)](http://docs.google.com/java/lang/String.html#valueOf(int)) is translated into bytes according to the platform's default character encoding, and these bytes are written in exactly the manner of the [write(int)](http://docs.google.com/java/io/PrintStream.html#write(int)) method.

**Parameters:**i - The int to be printed**See Also:**[Integer.toString(int)](http://docs.google.com/java/lang/Integer.html#toString(int))

### print

public void **print**(long l)

Prints a long integer. The string produced by [String.valueOf(long)](http://docs.google.com/java/lang/String.html#valueOf(long)) is translated into bytes according to the platform's default character encoding, and these bytes are written in exactly the manner of the [write(int)](http://docs.google.com/java/io/PrintStream.html#write(int)) method.

**Parameters:**l - The long to be printed**See Also:**[Long.toString(long)](http://docs.google.com/java/lang/Long.html#toString(long))

### print

public void **print**(float f)

Prints a floating-point number. The string produced by [String.valueOf(float)](http://docs.google.com/java/lang/String.html#valueOf(float)) is translated into bytes according to the platform's default character encoding, and these bytes are written in exactly the manner of the [write(int)](http://docs.google.com/java/io/PrintStream.html#write(int)) method.

**Parameters:**f - The float to be printed**See Also:**[Float.toString(float)](http://docs.google.com/java/lang/Float.html#toString(float))

### print

public void **print**(double d)

Prints a double-precision floating-point number. The string produced by [String.valueOf(double)](http://docs.google.com/java/lang/String.html#valueOf(double)) is translated into bytes according to the platform's default character encoding, and these bytes are written in exactly the manner of the [write(int)](http://docs.google.com/java/io/PrintStream.html#write(int)) method.

**Parameters:**d - The double to be printed**See Also:**[Double.toString(double)](http://docs.google.com/java/lang/Double.html#toString(double))

### print

public void **print**(char[] s)

Prints an array of characters. The characters are converted into bytes according to the platform's default character encoding, and these bytes are written in exactly the manner of the [write(int)](http://docs.google.com/java/io/PrintStream.html#write(int)) method.

**Parameters:**s - The array of chars to be printed **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - If s is null

### print

public void **print**([String](http://docs.google.com/java/lang/String.html) s)

Prints a string. If the argument is null then the string "null" is printed. Otherwise, the string's characters are converted into bytes according to the platform's default character encoding, and these bytes are written in exactly the manner of the [write(int)](http://docs.google.com/java/io/PrintStream.html#write(int)) method.

**Parameters:**s - The String to be printed

### print

public void **print**([Object](http://docs.google.com/java/lang/Object.html) obj)

Prints an object. The string produced by the [String.valueOf(Object)](http://docs.google.com/java/lang/String.html#valueOf(java.lang.Object)) method is translated into bytes according to the platform's default character encoding, and these bytes are written in exactly the manner of the [write(int)](http://docs.google.com/java/io/PrintStream.html#write(int)) method.

**Parameters:**obj - The Object to be printed**See Also:**[Object.toString()](http://docs.google.com/java/lang/Object.html#toString())

### println

public void **println**()

Terminates the current line by writing the line separator string. The line separator string is defined by the system property line.separator, and is not necessarily a single newline character ('\n').

### println

public void **println**(boolean x)

Prints a boolean and then terminate the line. This method behaves as though it invokes [print(boolean)](http://docs.google.com/java/io/PrintStream.html#print(boolean)) and then [println()](http://docs.google.com/java/io/PrintStream.html#println()).

**Parameters:**x - The boolean to be printed

### println

public void **println**(char x)

Prints a character and then terminate the line. This method behaves as though it invokes [print(char)](http://docs.google.com/java/io/PrintStream.html#print(char)) and then [println()](http://docs.google.com/java/io/PrintStream.html#println()).

**Parameters:**x - The char to be printed.

### println

public void **println**(int x)

Prints an integer and then terminate the line. This method behaves as though it invokes [print(int)](http://docs.google.com/java/io/PrintStream.html#print(int)) and then [println()](http://docs.google.com/java/io/PrintStream.html#println()).

**Parameters:**x - The int to be printed.

### println

public void **println**(long x)

Prints a long and then terminate the line. This method behaves as though it invokes [print(long)](http://docs.google.com/java/io/PrintStream.html#print(long)) and then [println()](http://docs.google.com/java/io/PrintStream.html#println()).

**Parameters:**x - a The long to be printed.

### println

public void **println**(float x)

Prints a float and then terminate the line. This method behaves as though it invokes [print(float)](http://docs.google.com/java/io/PrintStream.html#print(float)) and then [println()](http://docs.google.com/java/io/PrintStream.html#println()).

**Parameters:**x - The float to be printed.

### println

public void **println**(double x)

Prints a double and then terminate the line. This method behaves as though it invokes [print(double)](http://docs.google.com/java/io/PrintStream.html#print(double)) and then [println()](http://docs.google.com/java/io/PrintStream.html#println()).

**Parameters:**x - The double to be printed.

### println

public void **println**(char[] x)

Prints an array of characters and then terminate the line. This method behaves as though it invokes [print(char[])](http://docs.google.com/java/io/PrintStream.html#print(char%5B%5D)) and then [println()](http://docs.google.com/java/io/PrintStream.html#println()).

**Parameters:**x - an array of chars to print.

### println

public void **println**([String](http://docs.google.com/java/lang/String.html) x)

Prints a String and then terminate the line. This method behaves as though it invokes [print(String)](http://docs.google.com/java/io/PrintStream.html#print(java.lang.String)) and then [println()](http://docs.google.com/java/io/PrintStream.html#println()).

**Parameters:**x - The String to be printed.

### println

public void **println**([Object](http://docs.google.com/java/lang/Object.html) x)

Prints an Object and then terminate the line. This method calls at first String.valueOf(x) to get the printed object's string value, then behaves as though it invokes [print(String)](http://docs.google.com/java/io/PrintStream.html#print(java.lang.String)) and then [println()](http://docs.google.com/java/io/PrintStream.html#println()).

**Parameters:**x - The Object to be printed.

### printf

public [PrintStream](http://docs.google.com/java/io/PrintStream.html) **printf**([String](http://docs.google.com/java/lang/String.html) format,  
 [Object](http://docs.google.com/java/lang/Object.html)... args)

A convenience method to write a formatted string to this output stream using the specified format string and arguments.

An invocation of this method of the form out.printf(format, args) behaves in exactly the same way as the invocation

out.format(format, args)

**Parameters:**format - A format string as described in [Format string syntax](http://docs.google.com/util/Formatter.html#syntax)args - Arguments referenced by the format specifiers in the format string. If there are more arguments than format specifiers, the extra arguments are ignored. The number of arguments is variable and may be zero. The maximum number of arguments is limited by the maximum dimension of a Java array as defined by the [Java Virtual Machine Specification](http://java.sun.com/docs/books/vmspec/). The behaviour on a null argument depends on the [conversion](http://docs.google.com/util/Formatter.html#syntax). **Returns:**This output stream **Throws:** IllegalFormatException - If a format string contains an illegal syntax, a format specifier that is incompatible with the given arguments, insufficient arguments given the format string, or other illegal conditions. For specification of all possible formatting errors, see the [Details](http://docs.google.com/util/Formatter.html#detail) section of the formatter class specification. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - If the format is null**Since:** 1.5

### printf

public [PrintStream](http://docs.google.com/java/io/PrintStream.html) **printf**([Locale](http://docs.google.com/java/util/Locale.html) l,  
 [String](http://docs.google.com/java/lang/String.html) format,  
 [Object](http://docs.google.com/java/lang/Object.html)... args)

A convenience method to write a formatted string to this output stream using the specified format string and arguments.

An invocation of this method of the form out.printf(l, format, args) behaves in exactly the same way as the invocation

out.format(l, format, args)

**Parameters:**l - The [locale](http://docs.google.com/java/util/Locale.html) to apply during formatting. If l is null then no localization is applied.format - A format string as described in [Format string syntax](http://docs.google.com/util/Formatter.html#syntax)args - Arguments referenced by the format specifiers in the format string. If there are more arguments than format specifiers, the extra arguments are ignored. The number of arguments is variable and may be zero. The maximum number of arguments is limited by the maximum dimension of a Java array as defined by the [Java Virtual Machine Specification](http://java.sun.com/docs/books/vmspec/). The behaviour on a null argument depends on the [conversion](http://docs.google.com/util/Formatter.html#syntax). **Returns:**This output stream **Throws:** IllegalFormatException - If a format string contains an illegal syntax, a format specifier that is incompatible with the given arguments, insufficient arguments given the format string, or other illegal conditions. For specification of all possible formatting errors, see the [Details](http://docs.google.com/util/Formatter.html#detail) section of the formatter class specification. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - If the format is null**Since:** 1.5

### format

public [PrintStream](http://docs.google.com/java/io/PrintStream.html) **format**([String](http://docs.google.com/java/lang/String.html) format,  
 [Object](http://docs.google.com/java/lang/Object.html)... args)

Writes a formatted string to this output stream using the specified format string and arguments.

The locale always used is the one returned by [Locale.getDefault()](http://docs.google.com/java/util/Locale.html#getDefault()), regardless of any previous invocations of other formatting methods on this object.

**Parameters:**format - A format string as described in [Format string syntax](http://docs.google.com/util/Formatter.html#syntax)args - Arguments referenced by the format specifiers in the format string. If there are more arguments than format specifiers, the extra arguments are ignored. The number of arguments is variable and may be zero. The maximum number of arguments is limited by the maximum dimension of a Java array as defined by the [Java Virtual Machine Specification](http://java.sun.com/docs/books/vmspec/). The behaviour on a null argument depends on the [conversion](http://docs.google.com/util/Formatter.html#syntax). **Returns:**This output stream **Throws:** IllegalFormatException - If a format string contains an illegal syntax, a format specifier that is incompatible with the given arguments, insufficient arguments given the format string, or other illegal conditions. For specification of all possible formatting errors, see the [Details](http://docs.google.com/util/Formatter.html#detail) section of the formatter class specification. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - If the format is null**Since:** 1.5

### format

public [PrintStream](http://docs.google.com/java/io/PrintStream.html) **format**([Locale](http://docs.google.com/java/util/Locale.html) l,  
 [String](http://docs.google.com/java/lang/String.html) format,  
 [Object](http://docs.google.com/java/lang/Object.html)... args)

Writes a formatted string to this output stream using the specified format string and arguments.

**Parameters:**l - The [locale](http://docs.google.com/java/util/Locale.html) to apply during formatting. If l is null then no localization is applied.format - A format string as described in [Format string syntax](http://docs.google.com/util/Formatter.html#syntax)args - Arguments referenced by the format specifiers in the format string. If there are more arguments than format specifiers, the extra arguments are ignored. The number of arguments is variable and may be zero. The maximum number of arguments is limited by the maximum dimension of a Java array as defined by the [Java Virtual Machine Specification](http://java.sun.com/docs/books/vmspec/). The behaviour on a null argument depends on the [conversion](http://docs.google.com/util/Formatter.html#syntax). **Returns:**This output stream **Throws:** IllegalFormatException - If a format string contains an illegal syntax, a format specifier that is incompatible with the given arguments, insufficient arguments given the format string, or other illegal conditions. For specification of all possible formatting errors, see the [Details](http://docs.google.com/util/Formatter.html#detail) section of the formatter class specification. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - If the format is null**Since:** 1.5

### append

public [PrintStream](http://docs.google.com/java/io/PrintStream.html) **append**([CharSequence](http://docs.google.com/java/lang/CharSequence.html) csq)

Appends the specified character sequence to this output stream.

An invocation of this method of the form out.append(csq) behaves in exactly the same way as the invocation

out.print(csq.toString())

Depending on the specification of toString for the character sequence csq, the entire sequence may not be appended. For instance, invoking then toString method of a character buffer will return a subsequence whose content depends upon the buffer's position and limit.

**Specified by:**[append](http://docs.google.com/java/lang/Appendable.html#append(java.lang.CharSequence)) in interface [Appendable](http://docs.google.com/java/lang/Appendable.html) **Parameters:**csq - The character sequence to append. If csq is null, then the four characters "null" are appended to this output stream. **Returns:**This output stream**Since:** 1.5

### append

public [PrintStream](http://docs.google.com/java/io/PrintStream.html) **append**([CharSequence](http://docs.google.com/java/lang/CharSequence.html) csq,  
 int start,  
 int end)

Appends a subsequence of the specified character sequence to this output stream.

An invocation of this method of the form out.append(csq, start, end) when csq is not null, behaves in exactly the same way as the invocation

out.print(csq.subSequence(start, end).toString())

**Specified by:**[append](http://docs.google.com/java/lang/Appendable.html#append(java.lang.CharSequence,%20int,%20int)) in interface [Appendable](http://docs.google.com/java/lang/Appendable.html) **Parameters:**csq - The character sequence from which a subsequence will be appended. If csq is null, then characters will be appended as if csq contained the four characters "null".start - The index of the first character in the subsequenceend - The index of the character following the last character in the subsequence **Returns:**This output stream **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - If start or end are negative, start is greater than end, or end is greater than csq.length()**Since:** 1.5

### append

public [PrintStream](http://docs.google.com/java/io/PrintStream.html) **append**(char c)

Appends the specified character to this output stream.

An invocation of this method of the form out.append(c) behaves in exactly the same way as the invocation

out.print(c)

**Specified by:**[append](http://docs.google.com/java/lang/Appendable.html#append(char)) in interface [Appendable](http://docs.google.com/java/lang/Appendable.html) **Parameters:**c - The 16-bit character to append **Returns:**This output stream**Since:** 1.5

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PrintStream.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/io/PipedWriter.html)   [**NEXT CLASS**](http://docs.google.com/java/io/PrintWriter.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/PrintStream.html)    [**NO FRAMES**](http://docs.google.com/PrintStream.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#2jxsxqh) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).