| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PrintWriter.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/PrintStream.html)   [**NEXT CLASS**](http://docs.google.com/java/io/PushbackInputStream.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/PrintWriter.html)    [**NO FRAMES**](http://docs.google.com/PrintWriter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#17dp8vu) | [METHOD](#3j2qqm3) |

## **java.io**

Class PrintWriter

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.io.Writer](http://docs.google.com/java/io/Writer.html)  
 **java.io.PrintWriter**

**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)

public class **PrintWriter**extends [Writer](http://docs.google.com/java/io/Writer.html)

Prints formatted representations of objects to a text-output stream. This class implements all of the print methods found in [PrintStream](http://docs.google.com/java/io/PrintStream.html). It does not contain methods for writing raw bytes, for which a program should use unencoded byte streams.

Unlike the [PrintStream](http://docs.google.com/java/io/PrintStream.html) class, if automatic flushing is enabled it will be done only when one of the println, printf, or format methods is invoked, rather than whenever a newline character happens to be output. These methods use the platform's own notion of line separator rather than the newline character.

Methods in this class never throw I/O exceptions, although some of its constructors may. The client may inquire as to whether any errors have occurred by invoking [checkError()](http://docs.google.com/java/io/PrintWriter.html#checkError()).

**Since:** JDK1.1

| **Field Summary** | |
| --- | --- |
| protected  [Writer](http://docs.google.com/java/io/Writer.html) | [**out**](http://docs.google.com/java/io/PrintWriter.html#out)            The underlying character-output stream of this PrintWriter. |

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

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

| **Method Summary** | |
| --- | --- |
| [PrintWriter](http://docs.google.com/java/io/PrintWriter.html) | [**append**](http://docs.google.com/java/io/PrintWriter.html#append(char))(char c)            Appends the specified character to this writer. |
| [PrintWriter](http://docs.google.com/java/io/PrintWriter.html) | [**append**](http://docs.google.com/java/io/PrintWriter.html#append(java.lang.CharSequence))([CharSequence](http://docs.google.com/java/lang/CharSequence.html) csq)            Appends the specified character sequence to this writer. |
| [PrintWriter](http://docs.google.com/java/io/PrintWriter.html) | [**append**](http://docs.google.com/java/io/PrintWriter.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 writer. |
| boolean | [**checkError**](http://docs.google.com/java/io/PrintWriter.html#checkError())()            Flushes the stream if it's not closed and checks its error state. |
| protected  void | [**clearError**](http://docs.google.com/java/io/PrintWriter.html#clearError())()            Clears the error state of this stream. |
| void | [**close**](http://docs.google.com/java/io/PrintWriter.html#close())()            Closes the stream and releases any system resources associated with it. |
| void | [**flush**](http://docs.google.com/java/io/PrintWriter.html#flush())()            Flushes the stream. |
| [PrintWriter](http://docs.google.com/java/io/PrintWriter.html) | [**format**](http://docs.google.com/java/io/PrintWriter.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 writer using the specified format string and arguments. |
| [PrintWriter](http://docs.google.com/java/io/PrintWriter.html) | [**format**](http://docs.google.com/java/io/PrintWriter.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 writer using the specified format string and arguments. |
| void | [**print**](http://docs.google.com/java/io/PrintWriter.html#print(boolean))(boolean b)            Prints a boolean value. |
| void | [**print**](http://docs.google.com/java/io/PrintWriter.html#print(char))(char c)            Prints a character. |
| void | [**print**](http://docs.google.com/java/io/PrintWriter.html#print(char%5B%5D))(char[] s)            Prints an array of characters. |
| void | [**print**](http://docs.google.com/java/io/PrintWriter.html#print(double))(double d)            Prints a double-precision floating-point number. |
| void | [**print**](http://docs.google.com/java/io/PrintWriter.html#print(float))(float f)            Prints a floating-point number. |
| void | [**print**](http://docs.google.com/java/io/PrintWriter.html#print(int))(int i)            Prints an integer. |
| void | [**print**](http://docs.google.com/java/io/PrintWriter.html#print(long))(long l)            Prints a long integer. |
| void | [**print**](http://docs.google.com/java/io/PrintWriter.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/PrintWriter.html#print(java.lang.String))([String](http://docs.google.com/java/lang/String.html) s)            Prints a string. |
| [PrintWriter](http://docs.google.com/java/io/PrintWriter.html) | [**printf**](http://docs.google.com/java/io/PrintWriter.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 writer using the specified format string and arguments. |
| [PrintWriter](http://docs.google.com/java/io/PrintWriter.html) | [**printf**](http://docs.google.com/java/io/PrintWriter.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 writer using the specified format string and arguments. |
| void | [**println**](http://docs.google.com/java/io/PrintWriter.html#println())()            Terminates the current line by writing the line separator string. |
| void | [**println**](http://docs.google.com/java/io/PrintWriter.html#println(boolean))(boolean x)            Prints a boolean value and then terminates the line. |
| void | [**println**](http://docs.google.com/java/io/PrintWriter.html#println(char))(char x)            Prints a character and then terminates the line. |
| void | [**println**](http://docs.google.com/java/io/PrintWriter.html#println(char%5B%5D))(char[] x)            Prints an array of characters and then terminates the line. |
| void | [**println**](http://docs.google.com/java/io/PrintWriter.html#println(double))(double x)            Prints a double-precision floating-point number and then terminates the line. |
| void | [**println**](http://docs.google.com/java/io/PrintWriter.html#println(float))(float x)            Prints a floating-point number and then terminates the line. |
| void | [**println**](http://docs.google.com/java/io/PrintWriter.html#println(int))(int x)            Prints an integer and then terminates the line. |
| void | [**println**](http://docs.google.com/java/io/PrintWriter.html#println(long))(long x)            Prints a long integer and then terminates the line. |
| void | [**println**](http://docs.google.com/java/io/PrintWriter.html#println(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) x)            Prints an Object and then terminates the line. |
| void | [**println**](http://docs.google.com/java/io/PrintWriter.html#println(java.lang.String))([String](http://docs.google.com/java/lang/String.html) x)            Prints a String and then terminates the line. |
| protected  void | [**setError**](http://docs.google.com/java/io/PrintWriter.html#setError())()            Indicates that an error has occurred. |
| void | [**write**](http://docs.google.com/java/io/PrintWriter.html#write(char%5B%5D))(char[] buf)            Writes an array of characters. |
| void | [**write**](http://docs.google.com/java/io/PrintWriter.html#write(char%5B%5D,%20int,%20int))(char[] buf, int off, int len)            Writes A Portion of an array of characters. |
| void | [**write**](http://docs.google.com/java/io/PrintWriter.html#write(int))(int c)            Writes a single character. |
| void | [**write**](http://docs.google.com/java/io/PrintWriter.html#write(java.lang.String))([String](http://docs.google.com/java/lang/String.html) s)            Writes a string. |
| void | [**write**](http://docs.google.com/java/io/PrintWriter.html#write(java.lang.String,%20int,%20int))([String](http://docs.google.com/java/lang/String.html) s, int off, int len)            Writes a portion of a string. |

| **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)) |

| **Field Detail** |
| --- |

### out

protected [Writer](http://docs.google.com/java/io/Writer.html) **out**

The underlying character-output stream of this PrintWriter.

**Since:** 1.2

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

### PrintWriter

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

Creates a new PrintWriter, without automatic line flushing.

**Parameters:**out - A character-output stream

### PrintWriter

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

Creates a new PrintWriter.

**Parameters:**out - A character-output streamautoFlush - A boolean; if true, the println, printf, or format methods will flush the output buffer

### PrintWriter

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

Creates a new PrintWriter, without automatic line flushing, from an existing OutputStream. This convenience constructor creates the necessary intermediate OutputStreamWriter, which will convert characters into bytes using the default character encoding.

**Parameters:**out - An output stream**See Also:**[OutputStreamWriter.OutputStreamWriter(java.io.OutputStream)](http://docs.google.com/java/io/OutputStreamWriter.html#OutputStreamWriter(java.io.OutputStream))

### PrintWriter

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

Creates a new PrintWriter from an existing OutputStream. This convenience constructor creates the necessary intermediate OutputStreamWriter, which will convert characters into bytes using the default character encoding.

**Parameters:**out - An output streamautoFlush - A boolean; if true, the println, printf, or format methods will flush the output buffer**See Also:**[OutputStreamWriter.OutputStreamWriter(java.io.OutputStream)](http://docs.google.com/java/io/OutputStreamWriter.html#OutputStreamWriter(java.io.OutputStream))

### PrintWriter

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

Creates a new PrintWriter, 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 writer. 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 string 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

### PrintWriter

public **PrintWriter**([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 PrintWriter, 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 writer. 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 string 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

### PrintWriter

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

Creates a new PrintWriter, 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 writer. 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

### PrintWriter

public **PrintWriter**([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 PrintWriter, 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 writer. 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(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.

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

### close

public void **close**()

Closes the stream and releases any system resources associated with it. Closing a previously closed stream has no effect.

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

### checkError

public boolean **checkError**()

Flushes the stream if it's not closed and checks its error state.

**Returns:**true if the print stream has encountered an error, either on the underlying output stream or during a format conversion.

### setError

protected void **setError**()

Indicates that an error has occurred.

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

### clearError

protected void **clearError**()

Clears the error state of this stream.

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

**Since:** 1.6

### write

public void **write**(int c)

Writes a single character.

**Overrides:**[write](http://docs.google.com/java/io/Writer.html#write(int)) in class [Writer](http://docs.google.com/java/io/Writer.html) **Parameters:**c - int specifying a character to be written.

### write

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

Writes A Portion of an array of characters.

**Specified by:**[write](http://docs.google.com/java/io/Writer.html#write(char%5B%5D,%20int,%20int)) in class [Writer](http://docs.google.com/java/io/Writer.html) **Parameters:**buf - Array of charactersoff - Offset from which to start writing characterslen - Number of characters to write

### write

public void **write**(char[] buf)

Writes an array of characters. This method cannot be inherited from the Writer class because it must suppress I/O exceptions.

**Overrides:**[write](http://docs.google.com/java/io/Writer.html#write(char%5B%5D)) in class [Writer](http://docs.google.com/java/io/Writer.html) **Parameters:**buf - Array of characters to be written

### write

public void **write**([String](http://docs.google.com/java/lang/String.html) s,  
 int off,  
 int len)

Writes a portion of a string.

**Overrides:**[write](http://docs.google.com/java/io/Writer.html#write(java.lang.String,%20int,%20int)) in class [Writer](http://docs.google.com/java/io/Writer.html) **Parameters:**s - A Stringoff - Offset from which to start writing characterslen - Number of characters to write

### write

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

Writes a string. This method cannot be inherited from the Writer class because it must suppress I/O exceptions.

**Overrides:**[write](http://docs.google.com/java/io/Writer.html#write(java.lang.String)) in class [Writer](http://docs.google.com/java/io/Writer.html) **Parameters:**s - String to be written

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

**Parameters:**x - the boolean value to be printed

### println

public void **println**(char x)

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

**Parameters:**x - the char value to be printed

### println

public void **println**(int x)

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

**Parameters:**x - the int value to be printed

### println

public void **println**(long x)

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

**Parameters:**x - the long value to be printed

### println

public void **println**(float x)

Prints a floating-point number and then terminates the line. This method behaves as though it invokes [print(float)](http://docs.google.com/java/io/PrintWriter.html#print(float)) and then [println()](http://docs.google.com/java/io/PrintWriter.html#println()).

**Parameters:**x - the float value to be printed

### println

public void **println**(double x)

Prints a double-precision floating-point number and then terminates the line. This method behaves as though it invokes [print(double)](http://docs.google.com/java/io/PrintWriter.html#print(double)) and then [println()](http://docs.google.com/java/io/PrintWriter.html#println()).

**Parameters:**x - the double value to be printed

### println

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

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

**Parameters:**x - the array of char values to be printed

### println

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

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

**Parameters:**x - the String value to be printed

### println

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

Prints an Object and then terminates 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/PrintWriter.html#print(java.lang.String)) and then [println()](http://docs.google.com/java/io/PrintWriter.html#println()).

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

### printf

public [PrintWriter](http://docs.google.com/java/io/PrintWriter.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 writer using the specified format string and arguments. If automatic flushing is enabled, calls to this method will flush the output buffer.

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 writer **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 [PrintWriter](http://docs.google.com/java/io/PrintWriter.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 writer using the specified format string and arguments. If automatic flushing is enabled, calls to this method will flush the output buffer.

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 writer **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 [PrintWriter](http://docs.google.com/java/io/PrintWriter.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 writer using the specified format string and arguments. If automatic flushing is enabled, calls to this method will flush the output buffer.

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 writer **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 [PrintWriter](http://docs.google.com/java/io/PrintWriter.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 writer using the specified format string and arguments. If automatic flushing is enabled, calls to this method will flush the output buffer.

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

Appends the specified character sequence to this writer.

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

out.write(csq.toString())

Depending on the specification of toString for the character sequence csq, the entire sequence may not be appended. For instance, invoking the 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)**Overrides:**[append](http://docs.google.com/java/io/Writer.html#append(java.lang.CharSequence)) in class [Writer](http://docs.google.com/java/io/Writer.html) **Parameters:**csq - The character sequence to append. If csq is null, then the four characters "null" are appended to this writer. **Returns:**This writer**Since:** 1.5

### append

public [PrintWriter](http://docs.google.com/java/io/PrintWriter.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 writer.

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.write(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)**Overrides:**[append](http://docs.google.com/java/io/Writer.html#append(java.lang.CharSequence,%20int,%20int)) in class [Writer](http://docs.google.com/java/io/Writer.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 writer **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 [PrintWriter](http://docs.google.com/java/io/PrintWriter.html) **append**(char c)

Appends the specified character to this writer.

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

out.write(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)**Overrides:**[append](http://docs.google.com/java/io/Writer.html#append(char)) in class [Writer](http://docs.google.com/java/io/Writer.html) **Parameters:**c - The 16-bit character to append **Returns:**This writer**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/PrintWriter.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/PrintStream.html)   [**NEXT CLASS**](http://docs.google.com/java/io/PushbackInputStream.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/PrintWriter.html)    [**NO FRAMES**](http://docs.google.com/PrintWriter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#17dp8vu) | [METHOD](#3j2qqm3) |

[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).