

If you don't know what a package is, read our [Java Packages Tutorial](#).

The `File` class has many useful methods for creating and getting information about files. For example:

Method	Type	Description
<code>canRead()</code>	Boolean	Tests whether the file is readable or not
<code>canWrite()</code>	Boolean	Tests whether the file is writable or not
<code>createNewFile()</code>	Boolean	Creates an empty file
<code>delete()</code>	Boolean	Deletes a file
<code>exists()</code>	Boolean	Tests whether the file exists
<code>getName()</code>	String	Returns the name of the file
<code>getAbsolutePath()</code>	String	Returns the absolute pathname of the file
<code>length()</code>	Long	Returns the size of the file in bytes
<code>list()</code>	String[]	Returns an array of the files in the directory
<code>mkdir()</code>	Boolean	Creates a directory

You will learn how to create, write, read and delete files in the next chapters:

[Create/Write Files »](#)

by completing
a Java
course today!



A circular green badge with a white border. Inside the circle, the text "w3schools" is written in a white, lowercase, sans-serif font along the top arc. In the center is a large white "W" followed by a superscript "3". Along the bottom arc, the word "CERTIFIED" is written in a white, uppercase, sans-serif font on the left, and "2022" is written on the right, separated by a small white dot.

Get started

CODE GAME




Play Game

Method	Type	Description
<code>canRead()</code>	Boolean	Tests whether the file is readable or not
<code>canWrite()</code>	Boolean	Tests whether the file is writable or not
<code>createNewFile()</code>	Boolean	Creates an empty file
<code>delete()</code>	Boolean	Deletes a file
<code>exists()</code>	Boolean	Tests whether the file exists
<code>getName()</code>	String	Returns the name of the file
<code>getAbsolutePath()</code>	String	Returns the absolute pathname of the file
<code>length()</code>	Long	Returns the size of the file in bytes
<code>list()</code>	String[]	Returns an array of the files in the directory
<code>mkdir()</code>	Boolean	Creates a directory

You will learn how to create, write, read and delete files in the next chapters:

- Create/Write Files »
 - Read Files »
 - Delete Files »



Get started

CODE GAME



Play Game

Java Create and Write To Files

[← Previous](#)

Next >

Create a File

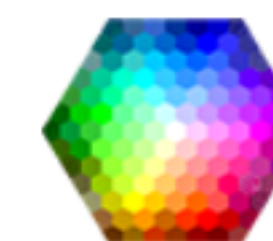
To create a file in Java, you can use the `createNewFile()` method. This method returns a boolean value: `true` if the file was successfully created, and `false` if the file already exists. Note that the method is enclosed in a `try...catch` block. This is necessary because it throws an `IOException` if an error occurs (if the file cannot be created for some reason):

Example

```
import java.io.File; // Import the File class
import java.io.IOException; // Import the IOException class to handle errors

public class CreateFile {
    public static void main(String[] args) {
        try {
            File myObj = new File("filename.txt");
            if (myObj.createNewFile()) {
                System.out.println("File created: " + myObj.getName());
            }
        }
    }
}
```

COLOR PICKER



Get certifie

CODE GAME



Write To a File

In the following example, we use the `FileWriter` class together with its `write()` method to write some text to the file we created in the example above. Note that when you are done writing to the file, you should close it with the `close()` method:

Example

```
import java.io.FileWriter; // Import the FileWriter class
import java.io.IOException; // Import the IOException class to handle errors

public class WriteToFile {
    public static void main(String[] args) {
        try {
            FileWriter myWriter = new FileWriter("filename.txt");
            myWriter.write("Files in Java might be tricky, but it is fun enough!");
            myWriter.close();
            System.out.println("Successfully wrote to the file.");
        } catch (IOException e) {
            System.out.println("An error occurred.");
            e.printStackTrace();
        }
    }
}
```

The output will be:

Successfully wrote to the file.


```
    } else {
        System.out.println("The file does not exist.");
    }
}
}
```

The output will be:

```
File name: filename.txt
Absolute path: C:\Users\MyName\filename.txt
Writeable: true
Readable: true
File size in bytes: 0
```

Run Example »

- Java File Handling
 - Java Files
 - Java Create/Write Files
 - Java Read Files
 - Java Delete Files
- Java How To
 - Add Two Numbers
- Java Reference
 - Java Keywords
 - Java String Methods

Note: There are many available classes in the Java API that can be used to read and write files in Java: `FileReader`, `BufferedReader`, `Files`, `Scanner`, `FileInputStream`, `FileWriter`, `BufferedWriter`, `FileOutputStream`, etc. Which one to use depends on the Java version you're working with and whether you need to read bytes or characters, and the size of the file/lines etc.

Tip: To delete a file, read our [Java Delete Files](#) chapter.

[← Previous](#)

Next >

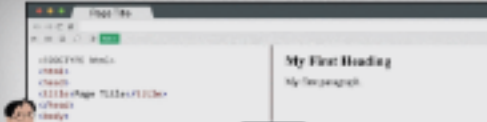
Get certified

To delete a file in Java, use the `delete()` method:

```
import java.io.File; // Import the File class

public class DeleteFile {
    public static void main(String[] args) {
        File myObj = new File("filename.txt");
        if (myObj.delete()) {
            System.out.println("Deleted the file: " + myObj.getName());
        } else {
            System.out.println("Failed to delete the file.");
        }
    }
}
```

```
Deleted the file: filename.txt
```



The screenshot shows a web browser window with a single tab titled "Page 10". The address bar displays "http://127.0.0.1:5501/". The page content is as follows:

- Navigation Menu:** A horizontal list of links: "Home", "About", "Contact", "Services", "Pages", "Blog", "Footer". The "Pages" link is highlighted with a green background.
- Main Content Area:**
 - Section Header:** "My First Heading" in bold black text.
 - Text:** "My first paragraph" in a standard black font.

A play button icon is overlaid on the right side of the browser window, indicating a video player interface.

COLOR PICKER



Get certified
by completing
a Java
course today!



Get started

CODE GAME

Play Game

Deleted the folder: Test

[Run Example »](#)