FILE HANDINLG

The File class of the java.io package is used to perform various operations on files and directories.

A directory is a collection of files and subdirectories. A directory inside a directory is known as subdirectory.

A file is a named location that can be used to store related information. For eg,**main.java** is a Java file

//Creating a Java File object

Syntax: File file = new File(String pathname);

we have created a file object named file. The object can be used to work with files and directories.

Example: File file = new File(“newfile.txt”);

**Java File Operation Methods**

|  |  |  |
| --- | --- | --- |
| Operation | Method | Package |
| To create file | createNewFile() | java.io.File |
| To read file | read() | java.io.FileReader |
| To write file | write() | java.io.FileWriter |
| To delete file | delete() | java.io.File |

### Java create files

To create a new file, we can use the createNewFile() method. It returns

* true if a new file is created.
* false if the file already exists in the specified location.

### Example: Create a new File

Here, we have used the file object to create the new file with the specified path.

// importing the File class

import java.io.File;

class Main {

public static void main(String[] args) {

// create a file object for the current location

File file = new File("newFile.txt");

try {

// trying to create a file based on the object

boolean value = file.createNewFile();

if (value) {

System.out.println("The new file is created.");

}

else {

System.out.println("The file already exists.");

}

}

catch(Exception e) {

e.getStackTrace();

}

}

}

### Java read files

To read data from the file, we can use subclasses of either [InputStream](https://www.programiz.com/java-programming/inputstream" \o "Java InputStream Class) or FileReader

/ importing the FileReader class

import java.io.FileReader;

class Main {

public static void main(String[] args) {

char[] array = new char[100];

try {

// Creates a reader using the FileReader

FileReader input = new FileReader("input.txt");

// Reads characters

input.read(array);

System.out.println("Data in the file:");

System.out.println(array);

// Closes the reader

input.close();

}

catch(Exception e) {

e.getStackTrace();

}

}

}

### Java write to files

To write data to the file, we can use subclasses of either [OutputStream](https://www.programiz.com/java-programming/outputstream" \o "Java OutputStream Class) or [Writer](https://www.programiz.com/java-programming/writer).

### Example: Write to file using FileWriter

// importing the FileWriter class

import java.io.FileWriter;

class Main {

public static void main(String args[]) {

String data = "This is the data in the output file";

try {

// Creates a Writer using FileWriter

FileWriter output = new FileWriter("output.txt");

// Writes string to the file

output.write(data);

System.out.println("Data is written to the file.");

// Closes the writer

output.close();

}

catch (Exception e) {

e.getStackTrace();

}

}

}

### Java delete files

We can use the delete() method of the File class to delete the specified file or directory. It returns

* true if the file is deleted.
* false if the file does not exist.

### Example: Delete a file

import java.io.File;

class Main {

public static void main(String[] args) {

// creates a file object

File file = new File("file.txt");

// deletes the file

boolean value = file.delete();

if(value) {

System.out.println("The File is deleted.");

}

else {

System.out.println("The File is not deleted.");

}

}

}