Working with Files and Directories in Java

# 1. How to Create a Directory in Java?

To create a directory, you use the File class and call the mkdir() method. If the directory was successfully created, mkdir() returns true.

Example:

import java.io.File;  
  
public class CreateDirectoryExample {  
 public static void main(String[] args) {  
 File directory = new File("MyFolder");  
 if (directory.mkdir()) {  
 System.out.println("Directory created successfully.");  
 } else {  
 System.out.println("Failed to create directory.");  
 }  
 }  
}

# 2. How to Create a File in Java?

To create a new file, use the File class together with the createNewFile() method. Since it might throw an IOException, you need to handle it using a try-catch block.

Example:

import java.io.File;  
import java.io.IOException;  
  
public class CreateFileExample {  
 public static void main(String[] args) {  
 File file = new File("myfile.txt");  
 try {  
 if (file.createNewFile()) {  
 System.out.println("File created successfully.");  
 } else {  
 System.out.println("File already exists.");  
 }  
 } catch (IOException e) {  
 System.out.println("An error occurred while creating the file.");  
 e.printStackTrace();  
 }  
 }  
}

# 3. How to Check if a File or Directory Exists?

Use the exists() method of the File class to check if a file or folder exists.

Example:

import java.io.File;  
  
public class CheckFileExistence {  
 public static void main(String[] args) {  
 File file = new File("myfile.txt");  
 if (file.exists()) {  
 System.out.println("The file exists.");  
 } else {  
 System.out.println("The file does not exist.");  
 }  
 }  
}

# 4. How to List Contents of a Directory?

Use the listFiles() method from the File class to get all files and folders inside a directory.

Example:

import java.io.File;  
  
public class ListDirectoryContents {  
 public static void main(String[] args) {  
 File currentDirectory = new File("."); // Current working directory  
 File[] files = currentDirectory.listFiles();  
  
 if (files != null) {  
 for (File file : files) {  
 if (file.isDirectory()) {  
 System.out.println("[DIR] " + file.getName());  
 } else {  
 System.out.println("[FILE] " + file.getName());  
 }  
 }  
 }  
 }  
}

# 5. Difference Between File and Path in Java

File is an older class used for file and directory handling. Path is part of the newer java.nio.file package and provides more powerful and flexible operations.

Example:

// Using File  
File file = new File("myfile.txt");  
  
// Using Path  
import java.nio.file.Path;  
import java.nio.file.Paths;  
  
Path path = Paths.get("myfile.txt");