**Java FileInputStream Class**

Java FileInputStream class obtains input bytes from a file. It is used for reading byte-oriented data (streams of raw bytes) such as image data, audio, video etc. You can also read character-stream data. But, for reading streams of characters, it is recommended to use FileReader class.

**Java FileInputStream class declaration**

Let's see the declaration for java.io.FileInputStream class:

1. **public** **class** FileInputStream **extends** InputStream

**Java FileInputStream class methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| int available() | It is used to return the estimated number of bytes that can be read from the input stream. |
| int read() | It is used to read the byte of data from the input stream. |
| int read(byte[] b) | It is used to read up to **b.length** bytes of data from the input stream. |
| int read(byte[] b, int off, int len) | It is used to read up to **len** bytes of data from the input stream. |
| long skip(long x) | It is used to skip over and discards x bytes of data from the input stream. |
| FileChannel getChannel() | It is used to return the unique FileChannel object associated with the file input stream. |
| FileDescriptor getFD() | It is used to return the FileDescriptor object. |
| protected void finalize() | It is used to ensure that the close method is call when there is no more reference to the file input stream. |
| void close() | It is used to closes the stream. |

**\*\*\* Sample file contains \*\*\***

***Java, C#, ASP.NET, SQL, Javascript***

**int read() & int read(byte[] b) & int read(byte[] b, int off, int len)**

|  |  |
| --- | --- |
| **package** com.company;  **import** java.io.FileInputStream; **import** java.io.IOException;  **public class** Main {   **public static void** main(String[] args) {  **try** {  FileInputStream fileInputStream = **new** FileInputStream(**"D:\\Cat.txt"**);  **int** i = fileInputStream.read();  System.***out***.println((**char**)i);  } **catch** (Exception ex){  System.***out***.println(**"Message : "** + ex);  }  } } | J |
| **package** com.company;  **import** java.io.FileInputStream; **import** java.io.IOException;  **public class** Main {   **public static void** main(String[] args) {  **try** {  FileInputStream fileInputStream = **new** FileInputStream(**"D:\\Cat.txt"**);  **int** i = 0;  **while** ((i = fileInputStream.read()) != -1){  System.***out***.print((**char**)i);  }  } **catch** (Exception ex){  System.***out***.println(**"Message : "** + ex);  }  } } | Java, C#, ASP.NET, SQL, Javascript |

|  |  |
| --- | --- |
| **package** com.company;  **import** java.io.FileInputStream; **import** java.io.IOException;  **public class** Main {   **public static void** main(String[] args) {  **try** {  FileInputStream fileInputStream = **new** FileInputStream(**"D:\\Cat.txt"**);  **int** i = 0;  **byte**[] buffer = **new byte**[30];  i = fileInputStream.read(buffer, 5, 20);  **for** (**byte** x : buffer){  **char** c = (**char**)x;  System.***out***.print(c);  }  } **catch** (Exception ex){  System.***out***.println(**"Message : "** + ex);  }  } } | -----Java, C#, ASP.NET, S |

**long skip(long x)**

|  |  |
| --- | --- |
| **package** com.company;  **import** java.io.FileInputStream; **import** java.io.IOException;  **public class** Main {   **public static void** main(String[] args) {  **try** {  FileInputStream fileInputStream = **new** FileInputStream(**"D:\\Cat.txt"**);  fileInputStream.skip(5);  **int** i = 0;  **while** ((i = fileInputStream.read()) != -1){  System.***out***.print((**char**)i);  }  } **catch** (Exception ex){  System.***out***.println(**"Message : "** + ex);  }  } } | **C#, ASP.NET, SQL, Javascript** |

**int available()**

|  |  |
| --- | --- |
| **package** com.company;  **import** java.io.FileInputStream; **import** java.io.IOException;  **public class** Main {   **public static void** main(String[] args) {  **try** {  FileInputStream fileInputStream = **new** FileInputStream(**"D:\\Cat.txt"**);  **int** s = fileInputStream.available(); *//GIVES US THE LENGTH OF THE STRING* System.***out***.println(**"Total no of char : "** + s);   fileInputStream.skip(5);   **int** s1 = fileInputStream.available(); *//skip METHOD MAKES THE DIFFERENCE* System.***out***.println(**"Total no of char AFTER SKIP METHOD: "** + s1);   **int** i = 0;  **while** ((i = fileInputStream.read()) != -1){  System.***out***.print((**char**)i);  }  } **catch** (Exception ex){  System.***out***.println(**"Message : "** + ex);  }  } } | Total no of char : 34  Total no of char AFTER SKIP METHOD: 29  C#, ASP.NET, SQL, Javascript |