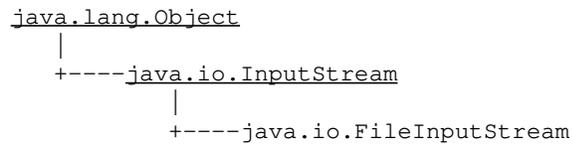


# Class `java.io.FileInputStream`



public class **FileInputStream**  
extends [InputStream](#)

File input stream, can be constructed from a file descriptor or a file name.

**See Also:**

[FileOutputStream](#), [File](#)

**Version:**

1.23, 08/11/95

**Author:**

Arthur van Hoff

---

## Constructor Index

- o **[FileInputStream](#)**(String)  
Creates an input file with the specified system dependent file name.
- o **[FileInputStream](#)**(File)  
Creates an input file from the specified File object.
- o **[FileInputStream](#)**(int)  
Creates an input file with the specified system dependent file descriptor.

## Method Index

- o **[available](#)**()  
Returns the number of bytes that can be read without blocking.
- o **[close](#)**()  
Closes the input stream.
- o **[finalize](#)**()  
Closes the stream when garbage is collected.
- o **[getFD](#)**()

Returns the file descriptor associated with this stream.

o **read()**

Reads a byte of data.

o **read(byte[])**

Reads data into an array of bytes.

o **read(byte[], int, int)**

Reads data into an array of bytes.

o **skip(long)**

Skips n bytes of input.

## Constructors

### o **FileInputStream**

```
public FileInputStream(String name) throws FileNotFoundException
```

Creates an input file with the specified system dependent file name.

**Parameters:**

name – the system dependent file name

**Throws:** IOException

If the file is not found.

### o **FileInputStream**

```
public FileInputStream(File file) throws FileNotFoundException
```

Creates an input file from the specified File object.

**Parameters:**

file – the file to be opened for reading

**Throws:** IOException

If the file is not found.

### o **FileInputStream**

```
public FileInputStream(int fd) throws IOException
```

Creates an input file with the specified system dependent file descriptor.

**Parameters:**

fd – the system dependent file descriptor

**Throws:** IOException

If an I/O error has occurred.

## Methods

### o **read**

```
public int read() throws IOException
```

Reads a byte of data. This method will block if no input is available.

**Returns:**

the byte read, or -1 if the end of the stream is reached.

**Throws:** IOException

If an I/O error has occurred.

**Overrides:**

read in class InputStream

**o read**

```
public int read(byte b[]) throws IOException
```

Reads data into an array of bytes. This method blocks until some input is available.

**Parameters:**

b – the buffer into which the data is read

**Returns:**

the actual number of bytes read, -1 is returned when the end of the stream is reached.

**Throws:** IOException

If an I/O error has occurred.

**Overrides:**

read in class InputStream

**o read**

```
public int read(byte b[],  
                int off,  
                int len) throws IOException
```

Reads data into an array of bytes. This method blocks until some input is available.

**Parameters:**

b – the buffer into which the data is read

off – the start offset of the data

len – the maximum number of bytes read

**Returns:**

the actual number of bytes read, -1 is returned when the end of the stream is reached.

**Throws:** IOException

If an I/O error has occurred.

**Overrides:**

read in class InputStream

**o skip**

```
public long skip(long n) throws IOException
```

Skips n bytes of input.

**Parameters:**

n – the number of bytes to be skipped

**Returns:**

the actual number of bytes skipped.

**Throws:** [IOException](#)

If an I/O error has occurred.

**Overrides:**

[skip](#) in class [InputStream](#)

**o available**

```
public int available() throws IOException
```

Returns the number of bytes that can be read without blocking.

**Returns:**

the number of available bytes, which is initially equal to the file size.

**Overrides:**

[available](#) in class [InputStream](#)

**o close**

```
public void close() throws IOException
```

Closes the input stream. This method must be called to release any resources associated with the stream.

**Throws:** [IOException](#)

If an I/O error has occurred.

**Overrides:**

[close](#) in class [InputStream](#)

**o getFD**

```
public final int getFD()
```

Returns the file descriptor associated with this stream.

**Returns:**

the file descriptor.

**o finalize**

```
protected void finalize() throws IOException
```

Closes the stream when garbage is collected.