

Java 11 – Features and Comparison

Every 6 months, Oracle releases a new version of Java. The latest version, Java 11, is already making a lot of buzz in the computer science world even before it was released by declaring that from now on Java is gonna be paid for commercial use.

Following articles will show important features, enhancements and deprecated features about JDK 11.

Important Changes and Information

- The deployment stack required for running applets and web applications has been removed from JDK which was deprecated in JDK 9.
- Entire section of supported browsers has been removed from list of supported configurations due to unavailability of deployment stack.
- Auto-update has been removed from JRE installations in Windows and MacOS.
- JavaFX and Java Mission Control is now available as a separate download.
- Java language translation for French, German, Italian, Korean, Portuguese (Brazilian), Spanish, and Swedish is no longer provided.
- In this version, JRE or Server JRE is no longer offered. Only JDK is offered.
- Updated packaging format for Windows has changed from tar.gz to .zip
- Updated package format for macOS has changed from .app to .dmg

New Features and Enhancements

1. New String methods:

- **isBlank():** This is a boolean method. It just returns true when a string is empty and vice-versa.

```
class GFG {  
    public static void main(String args[])  
    {  
        String str1 = "";  
        System.out.println(str1.isBlank());  
  
        String str2 = "GeeksForGeeks";  
        System.out.println(str2.isBlank());  
    }  
}
```

Output:

Custom Cut Wood Delivered

We specialize in custom cutting for any size project – no project too big or too small!

True
False

- **lines():** This method is to return a collection of strings which are divided by line terminators.

```
class GFG {  
    public static void main(String args[])  
    {  
        String str = "Geeks\nFor\nGeeks";  
        System.out.println(str  
                                .lines()  
                                .collect(  
                                    Collectors.toList()));  
    }  
}
```

Output:

Geeks
For
Geeks

- **repeat(n):** Result is the concatenated string of original string repeated the number of times in the argument.

```
class GFG {  
    public static void main(String args[])  
    {  
        String str = "GeeksForGeeks";  
        System.out.println(str.repeat(4));  
    }  
}
```

Output:

GeeksForGeeksGeeksForGeeksGeeksForGeeksGeeksForGeeks

- **stripLeading():** It is used to remove the white-space which is in-front of the string

```
class GFG {  
    public static void main(String args[])  
    {  
        String str = "  GeeksForGeeks";  
        System.out.println(str.stripLeading());  
    }  
}
```

Output:

GeeksForGeeks

- **stripTrailing():** It is used to remove the white-space which is in back of the string

```
class GFG {  
    public static void main(String args[])  
    {  
        String str = "GeeksForGeeks ";  
        System.out.println(str.stripTrailing());  
    }  
}
```

Output:

GeeksForGeeks

- **strip():** It is used to remove the white-spaces which are in-front and back of the string

```
class GFG {  
    public static void main(String args[])  
    {  
        String str = "  GeeksForGeeks ";  
        System.out.println(str.strip());  
    }  
}
```

Output:

GeeksForGeeks

2. New File Methods

- **writeString():** This is to write some content in a file.

```
jshell>Files.writeString(Path.of(example.txt), "GeeksForGeeks!");
```

- **readString():** This is used to read the contents of a file.

```
jshell>Files.readString(Path.of(example.txt));
```

Output: "GeeksForGeeks!"

→

- **isSameFile():** This method is used to know whether two paths locate the same file or not.

```
jshell>Files.isSameFile(Path.of("example1.txt"), Path.of("example1.txt"))
```

Output: true

```
jshell>Files.isSameFile(Path.of("example2.txt"), Path.of("example1.txt"))
```

Output: false

3. Pattern recognizing methods:

- **asMatchPredicate():** This method is similar to Java 8 method **asPredicate()**. Introduced in JDK 11, this method will create a predicate if pattern matches with input string.

```
jshell>var str = Pattern.compile("aba").asMatchPredicate();
```

```
jshell>str.test(aabb);
```

Output: false

```
jshell>str.test(aba);
```

Output: true

4. Epsilon Garbage Collector:

This handles memory allocation but does not have actual memory reclamation mechanism.

Once the available Java heap is exhausted, JVM will shut down.

Its goals are:-

- Performance testing
- Memory pressure testing
- last drop latency improvements

5. **Removed the Java EE and CORBA modules:** These modules were deprecated in Java 9 with declaration to remove those in further JDK versions.

6. **Removal of thread functions: stop(Throwable obj) and destroy()** objects have been removed from the JDK 11 because they only throw **UnsupportedOperationException** and **NoSuchMethodError** respectively. Other than that, they were of no use.

7. **TimeUnit Conversion:** This method is used to convert the given time to a unit like DAY, MONTH, YEAR and for time too.

```
jshell>TimeUnit c = TimeUnit.DAYS;
```

Output: DAYS

```
jshell>c.convert(Duration.ofHours(24));
```

Output: 1

```
jshell>c.convert(Duration.ofHours(50));
```

Output: 2

8. **Optional.isEmpty()** This method returns true if the value of any object is null and else returns false.

```
jshell>Optional str = Optional.empty();
```

```
jshell>str.isEmpty();
```

Output: true

```
jshell>Optional str = Optional.of("TonyStark");
```

```
jshell>str.isEmpty();
```

Output: false

9. **Local-Variable Syntax for Lambda Parameters:** JDK 11 allows 'var' to be used in lambda expressions. This was introduced to be consistent with local 'var' syntax of Java 10.

//Variable used in lambda expression

```
public class VarInLambdaExample {  
    public static void main(String[] args) {  
        IntStream.of(1, 2, 3, 5, 6, 7)  
            .filter((var i) -> i % 2 == 0)  
            .forEach(System.out::println);  
    }  
}
```

Output:

2
6

//Variable without using lambda expression

```
public class WithoutVarInLambdaExample {  
    public static void main(String[] args) {  
        IntStream.of(1, 2, 3, 5, 6, 7)  
            .filter(i -> i % 2 == 0)  
            .forEach(System.out::println);  
    }  
}
```

Output:

2
6

Removed Features and Options

1. Removal of com.sun.awt.AWTUtilities Class
2. Removal of Lucida Fonts from Oracle JDK
3. Removal of appletviewer Launcher
4. Oracle JDK's javax.imageio JPEG Plugin No Longer Supports Images with alpha
5. Removal of sun.misc.Unsafe.defineClass
6. Removal of Thread.destroy() and Thread.stop(Throwable) Methods

7. Removal of sun.nio.ch.disableSystemWideOverlappingFileLockCheck Property
8. Removal of sun.locale.formatasdefault Property
9. Removal of JVM-MANAGEMENT-MIB.mib
10. Removal of SNMP Agent
11. Removal of Java Deployment Technologies
12. Removal of JMC from the Oracle JDK
13. Removal of JavaFX from the Oracle JDK
14. JEP 320 Remove the Java EE and CORBA Modules

Deprecated Features and Options

1. ThreadPoolExecutor Should Not Specify a Dependency on Finalization
2. JEP 335 Deprecate the Nashorn JavaScript Engine
3. Deprecate -XX+AggressiveOpts
4. Obsolete Support for Commercial Features
5. Deprecate Stream-Based GSSContext Methods
6. JEP 336 Deprecate the Pack200 Tools and API



Custom Cut Wood Delivered

We specialize in custom cutting for any size project – no project too big or too small!

Recommended Posts:

- [AngularJS vs Angular2 | Features and Comparison](#)
- [Java 9 features with examples](#)
- [What are C++ features missing in Java?](#)
- [Comparison of Inheritance in C++ and Java](#)
- [Comparison of Java with other programming languages](#)
- [Comparison of Exception Handling in C++ and Java](#)
- [Comparison of static keyword in C++ and Java](#)
- [Image Processing in Java | Set 14 \(Comparison of two images \)](#)
- [Comparison of Autoboxed Integer objects in Java](#)
- [Comparison of yield\(\), join\(\) and sleep\(\) in Java](#)
- [Output of Java Programs | Set 53 \(String Comparison\)](#)

Comparison of boolean data type in C++ and Java

Interesting Facts and Features about AngularJS

String comparison using == vs strcmp() in PHP

Comparison between static and instance method in PHP



universedecoder

Check out this Author's [contributed articles](#).

If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please Improve this article if you find anything incorrect by clicking on the "Improve Article" button below.



Article Tags : [Java](#) [Technical Scripter](#) [java-file-handling](#) [Java-Strings](#) [Technical Scripter 2018](#)

Practice Tags : [Java-Strings](#) [Java](#)



3

1

☐ To-do ☐ Done

Based on 1 vote(s)

[Feedback/ Suggest Improvement](#)[Add Notes](#)[Improve Article](#)

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

[Load Comments](#)

A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

[About Us](#)
[Careers](#)
[Privacy Policy](#)
[Contact Us](#)

PRACTICE

[Courses](#)
[Company-wise](#)
[Topic-wise](#)
[How to begin?](#)

LEARN

[Algorithms](#)
[Data Structures](#)
[Languages](#)
[CS Subjects](#)
[Video Tutorials](#)

CONTRIBUTE

[Write an Article](#)
[Write Interview Experience](#)
[Internships](#)
[Videos](#)

@geeksforgeeks, Some rights reserved