

# JDK 11 Features with Examples

## 1. Running Java Files Without Compilation

Java 11 allows running Java files directly without manually compiling them.

```
java Hello.java
```

## 2. var in Lambda Expressions

Java 11 allows using 'var' in lambda expressions for explicit type inference.

```
(var name) -> System.out.println(name.toUpperCase());
```

## 3. New String Methods

Java 11 introduces several new String methods: `isBlank()`, `strip()`, `repeat()`, and `lines()`.

```
" ".isBlank(); // true
" Hello ".strip(); // "Hello"
"Java ".repeat(3); // "Java Java Java"
```

## 4. Files API Enhancements

`Files.writeString()` and `Files.readString()` simplify file read/write operations.

```
Files.writeString(Path.of("file.txt"), "Hello, Java 11!");
String content = Files.readString(Path.of("file.txt"));
```

## 5. Immutable Collections

`List.copyOf()` and `Set.copyOf()` create immutable collections.

```
List<String> copyList = List.copyOf(originalList);
```

## 6. Optional API Enhancements

New methods: `isEmpty()` and `orElseThrow()`.

```
Optional<String> name = Optional.of("Java 11");
System.out.println(name.isEmpty()); // false
```

## 7. HTTP Client

Java 11 introduced a modern HTTP Client in `java.net.http`.

```
HttpClient client = HttpClient.newHttpClient();
HttpRequest request =
    HttpRequest.newBuilder().uri(URI.create("https://example.com")).GET().build();
```

## 8. ZGC (Z Garbage Collector)

ZGC is a low-latency garbage collector for high-performance applications.

```
java -XX:+UnlockExperimentalVMOptions -XX:+UseZGC MyApp.java
```

## 9. Flight Recorder & Mission Control

Built-in monitoring tool for performance analysis.

```
java -XX:+FlightRecorder -XX:StartFlightRecording=duration=60s,filename=recording.jfr  
MyApp
```