Versions of java

Java has evolved steadily—bringing modern features, better performance, and stronger tooling with every major version.

- Java 8: Lambdas, Streams, New Date API
- Java 11: HTTP Client, Performance, First modern LTS
- Java 17: Sealed Classes, Modern syntax
- Java 21: Virtual Threads, Structured Concurrency
- Java 24: GC improvements, Post-quantum security, Stream Gatherers

Java 8 (2014) – Game-changer

1. Lambda Expression which is an anaonymous funtion.
2.Functional Interface: Interface with exactly one abstract method.
3. Stream API: to process the collections in a fucntional style.
4.New Date and Time API
5.Default and static methods
6.Optional class: to avoid null values and null pointer exceptions
7.Collectors: to collect results into collections.

Java 11 (2018) – First LTS post

1. Compile free launch: run the code without compilation
2. New string methods: isBlank(),lines(),repeat(),stringLeading(),stripTrailing(),
3. var in a lambda expression and it can be used to apply modifiers to local variables
4. Introduces toArray() method to convert a collection into an array.
Arrayvalues=listValues.toArray(String[]::new).
5. some new methods in files class
   a.readString(Path path, Charset cs)
   b.writeString(Path path, Charset cs)
6.Http Client is a standerd which is recomended to use instead of otherHTTP Client API's like Apache Http Client API.
7. Optional enhancement: isEmpty() method

Java 17 (2021) – Stable & modern

1.Sealed classes: better control over class hierarchies
2.Pattern Matching for instanceof
3. Records as a permanent feature
4.Strong Encapsulation by default.
5. New Garbage collectors:ZGC nad G1 enhancements
6. Contect-specific Deserialization filters
7. Foreign function & MemoryAPI
8. Vector API

Java 21 (2023) – Future-ready

1. Virtual threads: simplify high concurrency apps
2. Record patterns
3. Sequenced collections- consistent data ordering
4. String templates:cleaner string handling
5. Scoped values:to manage threads
6. Structured concurrency: better thread management
7. Unnamed classes and Instance Main methods

Java 24 (2025) – Smart & secure

1. Stream Gatherers for flexible data processing
2. Pattern Matching for primitives (Preview)
3.Compact Object Headers (smaller memory footprint)
4.Simpler instance main() methods
5.Scoped Values – safer alternative to ThreadLocal