

<u>Report on the Features and Applications</u> of Python and Java.

Chethina Randidu - 06/19/2025

1 Introduction

Python and Java are two of the most widely used programming languages in the software development industry. Both languages have distinct features, design philosophies, and use cases, making them suitable for different types of applications and development needs. In this report, I have analyzed all the key features and use cases of the two languages.

2 Python

Python is a **high-level**, **interpreted programming** *language* created by *Guido van Rossum* and first released in **1991**. It is designed to be simple, readable, and easy to learn, making it an excellent choice for both beginners and experts. Python supports multiple programming paradigms, including **procedural**, **object-oriented**, **and functional programming**. Its extensive standard library and third-party packages make it highly versatile for various applications, from web development to data science.

3 Java

Java is a **high-level**, **compiled programming** *language* developed by *James Gosling* at *Sun Microsystems* (now owned by *Oracle*) and released in **1995**. It is designed to be platform-independent, allowing Java code to run on any platform that supports the *Java Virtual Machine* (JVM). Java is strictly object-oriented and is known for its **robustness, security, and performance**. It is widely used for building **enterprise-level applications, Android apps, and large-scale systems.**

4 Syntax Comparison

Syntax is like set of rules that define how code is written in a programming language. Python and Java have vast difference when it comes to these rules, let's discuss them below.

4.1 Code blocks

• **Python**: Uses indentation to define blocks, enforcing consistent and readable code structure.

```
1. if x > 0;
2. print("Positive")
3. else:
4. print("Non-positive")
```

• Java: Using curly braces { } to define code blocks.

```
1. If (x > 0) {
2. System.out.printIn ("Positive");
3.} else {
4. System.out.printIn ("Non-positive");
5.}
```

4.2 Variable Declaration

• **Python:** Variables are declared without specifying their types, as Python is dynamically typed.

1. x = 5 2. y = "Hello"

• **Java:** Variables must be declared with their type, as Java is statically typed.

```
1. Int x = 5;
2. String y = "Hello";
```

4.3 Semicolons

- **Python:** Does not require semicolons at the end of statements.
- Java: Requires semicolons at the end of each statement.

4.4 Classes and Objects

• **Python:** Classes are defined using the *class* keyword, and objects are created by calling the class as a function.

```
1. class Person:
2. def __init__(self, name):
3. self.name = name
4.
5. p = Person("Alice")
```

• **Java:** Classes are defined using the *class* keyword, and objects are created using the *new* keyword.

```
1. public class Person {
2. String name;
3. public Person(String name) {
4. this.name = name;
5. Person p = new Person("Alice");
```

Key Difference: Python's syntax is more concise and readable due to its use of indentation and lack of semicolons, while Java's syntax is more verbose but provides explicit structure with curly braces and type declarations.

5 Comparison of Data Types

Data types define the kind of data that a variable can hold. Python and Java handle data types differently due to their typing systems.

Feature	Python	Java
Typing	Dynamic	Static
Basic Types	Int, float, str, bool,	byte, short, int, long,
	complex	float, double, char
Mutability	Lists, dicts mutable;	Arrays, objects mutable;
	strings, tuples, immutable	strings
Complex Numbers	Built-in support	No built-in support
Enums	Emulated via classes	Built-in support

5.1 Basic Data Types

o Python: Supports integers, floats, strings, booleans, complex numbers, etc.

• **Java:** Supports primitive types (e.g., byte, short, int, long, float, double, char, boolean) and reference types (e.g., String, arrays, classes).

5.2 Type Declaration

- **Python:** Variables do not need to be declared with a type; the type is inferred at runtime.
- **Java:** Variables must be declared with a specific type before use.

5.3 Mutability

- **Python:** Some data types are mutable (e.g., lists, dictionaries), while others are im- mutable (e.g., strings, tuples).
- **Java:** Strings are immutable, while arrays and most objects are mutable.

5.4 Complex Numbers

- **Python:** Has built-in support for complex numbers.
- Java: Does not have built-in support; requires libraries or custom classes.

5.5 Enums

- **Python:** Does not have built-in enums; can be emulated using classes.
- **Java:** Has built-in support for enums.

Key Difference: Python's dynamic typing allows for more flexibility and shorter code, while Java's static typing provides better type safety and early error detection.

6 Comparison of Memory Management

Memory management refers to how a programming language handles the allocation and deallocation of memory for programs. Both Python and Java use automatic memory management through garbage collection, but their approaches differ.

6.1 Garbage Collection

• **Python:** Uses a combination of reference counting and garbage collection. Reference counting tracks the number of references to an object, and when the count drops to zero, the memory is freed. For circular references, Python uses a

garbage collector.

• **Java:** Relies solely on garbage collection. The JVM periodically runs the garbage collector to identify and free memory occupied by objects that are no longer referenced.

6.2 Manual Memory Management

- **Python:** Does not allow manual memory management; all memory management is automatic.
- **Java:** While garbage collection is automatic, Java provides some ways to influence memory management, such as using the *finalize* method or calling *System.gc()*, though these are generally not recommended.

6.3 Performance

- **Python:** Memory management can be less efficient due to the overhead of reference counting, but it is sufficient for most applications.
- **Java:** Java's garbage collection can be tuned for performance, making it efficient for large-scale applications.

Key Difference: Python's memory management is simpler and more automatic, while Java's allows for more control and tuning, making it better suited for performance-critical applications.

7 Applications and Uses

Both Python and Java are versatile languages used in a wide range of applications, catering to different development needs.

7.1 Applications of Python

- **Data Science and Machine Learning:** Libraries like *NumPy*, *Pandas, TensorFlow*, and *PyTorch* make Python ideal for data analysis and machine learning.
- **Web Development:** Frameworks like *Django* and *Flask* are popular for building web applications.
- **Automation and Scripting:** Python's simplicity suits automating repetitive tasks.

- **Scientific Computing:** Libraries like *SciPy* and *Matplotlib* support scientific applications.
- **Artificial Intelligence and NLP:** Libraries like *NLTK* and *spaCy* facilitate AI development.
- **Game Development:** *Pygame* is used for 2D games.
- **Desktop GUI Applications:** *Tkinter* and *PyQt* enable desktop app development.
- 7.2 Applications of Java
 - Enterprise Applications: Java's robustness suits large-scale enterprise systems.
 - **Android App Development:** Java is a primary language for Android apps.
 - **Web Applications:** Frameworks like *Spring* and *JavaServer Faces* are popular.
 - **Big Data Technologies:** Java is used in Hadoop and Apache Spark.
 - **Cloud Applications:** Java's portability supports cloud-based applications.
 - **IoT Applications:** Java handles distributed systems in IoT.
 - **Gaming Applications:** Java supports cross-platform games.

Key Difference: Python is preferred for rapid development, data science, and scripting, while Java excels in enterprise applications, Android development, and high-performance systems.

8 My View

Python and Java are both powerful programming languages with their strengths. Python's simplicity, readability, and extensive libraries make it ideal for rapid development, data science, and web applications. Its dynamic typing and automatic memory management reduce development time but may introduce performance overhead. Java's robustness, performance, and platform independence make it suitable for enterprise applications, Android development, and large-scale systems. It's static typing and tunable garbage collection provide better type safety and performance, but require more verbose code. Python is recommended for beginners due to its ease of learning, while Java suits those interested in enterprise or Android development. Proficiency in both languages can be cool.

References,

- 1. <u>https://raygun.com/blog/java-vs-python/</u>
- <u>https://www.geeksforgeeks.org/difference-between-python-and-java/</u>
- 3. https://www.snaplogic.com/glossary/python-vs-java
- 4. https://www.datacamp.com/blog/python-vs-java
- 5. https://brightdata.com/blog/web-data/java-vs-python
- 6. <u>https://exadel.com/news/java-vs-python-which-one-should-you-choose/</u>
- 7. <u>https://realpython.com/python-memory-management/</u>
- 8. <u>https://kinsta.com/blog/python-vs-java/</u>
- 9. https://www.imaginarycloud.com/blog/python-vs-java
- 10. https://www.coursera.org/articles/python-vs-java
- 11. <u>https://medium.com/@arushkrisp/comparing-python-and-java-garbage-collection-memory-management-in-two-popular-languages-8306d6ea7f2d</u>