Character Sets in Java

Alphabets: A-Z, a-z
Digits: 0-9
Special Characters: ! @ # $ % ^ & * ( ) etc
Java uses UTF-16 encoding for char data type, meaning each character is stored in 16-bit units.

Delimiters

A **delimiter** is a character or sequence of characters that separates tokens in a string or a data stream. In Java, common delimiters include:

- **Whitespace characters** (space, tab, newline)
- **Punctuation marks** (,, ;, .)
- **Operators** (+, -, /, *, =)
- **Special characters** ({}, [], ())

Delimiters: Symbols that are used in Java programming to separate tokens in a program are called **delimiters**. These include punctuation marks, whitespace characters, and operators that define boundaries between different elements in Java code.

! % & * ( ) / ? ; etc

Unicode

Unicode is a **16-bit** character encoding scheme used in Java to represent characters from multiple languages. It assigns a unique **hexadecimal code** to each character.

- **Size**: 16-bit
- **Hexadecimal Range**: 0x0000 to 0xFFFF
- **Total Characters**: **160,755**

Unicode supports multiple encoding formats such as **UTF-8, UTF-16, and UTF-32**, with **UTF-16** being the default in Java.

Earlier character encoding schemes, such as **AEC (Alphabet Encoding for Computers)** and **ISO-8859**, were used but are now considered outdated due to their limited character support and lack of multilingual compatibility. Unicode replaced these older schemes to provide a universal character representation system.

ASCII (American Standard Code for Information Interchange)

ASCII is a **7-bit** encoding scheme that supports **128 characters** (0-127). It includes:

- Uppercase and lowercase letters (A-Z, a-z)
- Digits (0-9)
- Special symbols (@, #, &, etc.)
- Control characters (\n, \t, \r, etc.)

Extended ASCII

Extended ASCII is an **8-bit** encoding (supports **256 characters**) that includes the original **ASCII (0-127)** plus **additional symbols, foreign characters, and graphical elements** (128-255). (Total range: 0 to 255)

\n	**Newline (Line Break)**
\t	**Tab (Horizontal Tab)**
\r	**Carriage Return**
\b	**Backspace**
\f	**Form Feed (Page Break)**
\'	**Single Quote (')**
\"	**Double Quote (")**
****	**Backslash (\)**
\0	Null Character (ASCII 0)

Examples (with outputs):

```
        System.out.println("Hello\nWorld");
                                Hello
                                 World

        System.out.println("Java\tProgramming");

                                Java    Programming

         System.out.println("Hello\rYo");
                                Yollo

        System.out.println("ABCD\bE");
                                ABCE

        System.out.println("he said, \'Hello\' to me.");
                                he said, 'Hello' to me.

        System.out.println("She said, \"Java is OOP!\"");
                                She said, "Java is OOP!"

        System.out.println("\\Documents");
                                \Documents

        System.out.println("Java\0Programming");
                                JavaProgramming

    }
}
```

In Java, **tokens** are the smallest units of a program. Java has the following types of tokens:

1. **Keywords**
2. **Identifiers**
3. **Literals**
4. **Operators**
5. **Separators:**
    - **comma ,**
    - **semi-colon ;**
    - **parenthesis ( )**
    - **Curly braces { }**
    - **Square Brackets [ ]**
6. **Punctuators**
    - **Ternary ?**
    - **Terminator ;**
    - **Member operator .**

Java Library Classes

Java provides a rich set of pre-defined **library classes** in the form of packages, which help in performing various tasks such as data handling, input/output operations, networking, and more.

---

1. java.lang (Core Classes) - Automatically Imported

- Contains fundamental classes required for Java programs.
- **Common Classes:**
    - Object – Base class of all Java classes
    - String – Immutable sequence of characters
    - Math – Provides mathematical functions like sqrt(), pow(), abs()
    - System – Contains System.out.println(), input/output methods
    - Integer, Double, Character – Wrapper classes for primitive types

---

2. java.util (Utility Classes)

- Provides data structures, collections, and utility functions.
- **Common Classes:**
    - ArrayList, LinkedList – Dynamic arrays
    - HashMap, TreeMap – Key-value storage
    - HashSet, TreeSet – Unique element collections
    - Collections – Utility class for sorting, searching
    - Random – Generates random numbers
    - Scanner – Reads input from the user

Use of `import` Keyword and `*` in Java

1. import Keyword in Java

The import keyword in Java is used to **bring external classes or entire packages** into a program.

import java.util.Scanner; - Imports Scanner class into program

import java.util.*; - Imports all classes from java.util package

** Java **automatically imports java.lang** (e.g., System, String, Math), so no need to import it manually.

3. java.io (Input/Output Classes)

- Handles file reading/writing and streams.
- **Common Classes:**
  - File – Represents files and directories
  - BufferedReader, FileReader – Reads data from files
  - BufferedWriter, FileWriter – Writes data to files
  - InputStream, OutputStream – Handles byte stream input/output

---

4. java.net (Networking Classes)

- Supports network communication using TCP/IP.
- **Common Classes:**
  - Socket – Connects to network servers
  - ServerSocket – Listens for client connections
  - URL – Represents a web URL
  - HttpURLConnection – Handles HTTP requests

Standalone Applications and Applets in Java

1. Standalone Applications

A **standalone application** is a **self-contained Java program** that runs independently on a user's computer **without requiring a web browser** or external resources.

Characteristics of Standalone Applications:

- Runs directly on an **operating system** (Native Environment)
- Can be **console-based** or **GUI-based**.
- Uses Java's **JVM (Java Virtual Machine)** for execution.

2. Applets *(Deprecated in Java 9, Removed in Java 11)*

A **Java Applet** is a **small Java program** that runs inside a web browser using the **Java Plugin**. Applets were mainly used for interactive web applications but are now **obsolete** due to security concerns and lack of browser support.

Characteristics of Applets:

- Runs inside a **web browser**.
- Requires a **Java-enabled browser** and appletviewer for execution.
- Cannot access local system resources for security reasons.
- Written by extending the Applet class from java.applet package.