Character Set

A character set is a defined collection of symbols, letters, numbers, punctuation marks, and other characters.

Each character in the set is assigned a unique numerical code, called a code point, which allows computers to store, transmit, and interpret text.

Two of the most common character sets are ASCII and Unicode.

ASCII stands for The American Standard Code for Information Interchange. It's the oldest and most widely used character set.

Unicode is a newer character set, designed to support all of the world's writing systems.

Character Encodings

Character encoding is the process of assigning numbers to various characters, called glyphs.

A glyph can be an alphabetical character in any language, punctuation, or emojis, for example.

There are different ways to represent glyphs, with a numeric value.

ASCII Encodings

	Size	Includes Latin Alphabet	Notes
US-ASCII	7 bits	No	Smaller range of characters
ISO-8859-1	8 bits	Yes	More Widely Supported than US-ASCII

Unicode Encodings

	Size	Benefits
UTF-8	Variable (1 to 4 bytes)	**Most popular encoding on the internet.** Includes ISO-8859-1, and more. Can represent characters from all writing systems
UTF-16	2 bytes	Widely Supported
UTF-32	4 bytes	More efficient and straightforward to process, but uses more storage space. Rarely used.

Which should you use?

In general, UTF-8 is the better choice for most applications.

It's more efficient, more widely supported, and can represent a wider range of characters.

However, if you're only working with ASCII characters, ISO-8859-1 may be a better choice for efficiency reasons.

Java has the most common encodings specified on an enum, called StandardCharsets.

https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/nio/charset/StandardCharsets.html

Support on Files for reading data from a file

All of these methods read the entire contents of a file into memory.

These methods support files up to about 2 gigabytes.

Method Signature	**Description**	**Closes file?**
byte[] readAllBytes(Path path) throws IOException	Reads entire contents of **any** file into a byte array.	Yes
String readString(Path path) throws IOException	Reads entire contents of a **text** file into a string	Yes
List<String> readAllLines(Path path) throws IOException	Reads entire contents of a text file, into a list of string.	Yes
Stream<String> lines(Path path) throws IOException	Reads entire contents of a text file	On Terminal Operation