Strings in Go

- Go has two additional integer types called byte and rune that **are aliases** for uint8 and int32 data types. **In Go, the byte and rune data types are used to distinguish characters from integer values.**
- Golang doesn’t have a char data type. **It uses byte and rune to represent character values.**
- Characters or rune literals are expressed in Go by enclosing them in single quotes, **as in 'x' or '\n'** . Rune literals such as ‘a’ , ‘b’, ‘c’, ‘x’ or ‘\n’ are represented using Unicode Code Points. **A code point is a numeric value that represents a rune literal.**
- The character encoding scheme ASCII which is a Unicode subset, comprises 128 code points.
- A string is a series of bytes values. **A string is a slice of bytes** and any byte slice can be encoded in a string value.
- The Go terminology for code points is runes . A rune represent a single unicode character. **Rune 0x61 in hexadecimal represents the rune literal ‘a’.**