String Operators

There is two string operators: + and [].

To concatenate, i.e., append, two strings, use the **+** operator.

- `"abc" + "def"` returns a new string, `"abcdef"`

To get the Nth character of a string, use *string***[***n***]**. This returns a string with the character. Unlike other languages, JavaScript does not have a separate character data type.

- Indexing is zero-based, i.e., the first element is element 0.
- If the index is too big, the empty string is returned.
  - `"abcde"[0]` returns `"a"`
  - `"abcde"[4]` returns `"e"`
  - `"abcde"[5]` returns `""`

String Methods

Strings come with many useful methods attached.

To get the length of a string, use *string***.length**.

To see if a search string is inside another bigger string, use *bigString***.indexOf(***searchString***)**. This scans the big string from left to right for the search string. If found, it returns the index of the first character in the big string where the search string was found. If not found, it returns -1.

> Search is case-sensitive. `"Now is the time".indexOf("the")` returns -1 because `"The"` does not match `"the"`.

To see if a string starts with a particular substring, use *string***.startsWith(***subString***)**. This returns true if the string starts with the substring, otherwise it returns false.

> There is also *string***.endsWith(***subString***)**.

Often, when working with user input, you want to "clean up" the input. Several string methods are handy for this.

To change a string to all lower case, use *string***.toLowerCase()**. This returns a new string. It does not modify the original string.

> There is also *string***.toUpperCase()**.