FREE eBook

LEARNING

string

Free unaffiliated eBook created from
Stack Overflow contributors.

#string

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: string

It is an unofficial and free string ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official string.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with string

Remarks

This section provides an overview of what string is, and why a developer might want to use it.

It should also mention any large subjects within string, and link out to the related topics. Since the Documentation for string is new, you may need to create initial versions of those related topics.

Versions

Version	Remark	Release Date
ASCII String	7 bit	1963-06-17
UTF-8	8 bit default, variable length through surrogates	1992-09-30
UCS-2	16 bit, like UTF-16 without surrogates, used my Microsoft Windows NT	1993-07-23
UTF-16	16 bit default, variable length through surrogates	1996-07-31

Examples

Strings

A string is a sequence of character literals. To date, strings are supported by all modern programming languages,[1] but there is no consensus between language designers on how strings should be classified. As far as programming language design is concerned there are two primary concerns to take into consideration.

1. Should a string be treated as a primitive- or a composite value?
2. What string operations should be provided by the language itself?[2]

By making a string a primitive value the string operations provided by the language are all built-in; and cannot be defined in the language itself. This allows certain optimizations at a compiler level — specifically with respect to memory layout and re-use of strings using a so called string pool. The trade-off lies in not being able to use drop in replacements for the string operations, and such functions would have to be called using normal function calls while the syntax for using built-in operations is usually distinctly different. The consequence being that if and when more effective algorithms are discovered one cannot simply change which string library one is using. A contrived example being having substring search as part of the core feature set before Boyer–Moore string

search algorithm was developed.

Conversely, by defining a string to be a composite value such as an array of characters all the usual array operations become automatically applicable to strings. However, this results in all strings within the language as being fixed-length.[3]

Ultimately, another approach is representing strings as a list of characters which - as with arrays - allows the representation to inherit all list operations.

[1] Save the more esoteric ones such as Piet
[2] Languages usually offer string comparisons, primarily equality, concatenating strings with other strings, substituting single characters and substrings and lexographic sorting of strings. [3] Usually

Read Getting started with string online: https://riptutorial.com/string/topic/4567/getting-started-with-string

Credits

S. No	Chapters	Contributors
1	Getting started with string	Community, Filip Allberg, Thomas Weller