

String Functions

מסמך זה הורד מהאתר <http://underwar.livedns.co.il>.
אין להפיץ מסמך זה במדיה כלשהי, ללא אישור מפורש מאת המחבר.
מחבר המסמך איננו אחראי לכל נזק, ישיר או עקיף, שיגרם עקב השימוש במידע
המופיע במסמך, וכן לנכונות התוכן של הנושאים המופיעים במסמך. עם זאת,
המחבר עשה את מירב המאמצים כדי לספק את המידע המדויק והמלא ביותר.

כל הזכויות שמורות לניר אדר

Nir Adar

Email: underwar@hotmail.com

Home Page: <http://underwar.livedns.co.il>

אנא שלחו תיקונים והערות אל המחבר.

String functions

פונקציות הפועלות על מחרוזות הינן כלי חשוב בכל שפת תכנות. מסמך זה בא לרכז פעולות נפוצות שניתן לבצע על מחרוזות, והסברים כיצד ניתן לבצע אותן במגוון שפות שונות.

יישור טקסט לשמאל

בהינתן המחרוזת `myString`, ואורך מבוקש של הפלט `width`, נרצה להחזיר מחרוזת חדשה שמכילה את המחרוזת `myString`, ואחריה רווחים במספר כזה שיביאה לאורך המבוקש `width`. לדוגמא, בהינתן המחרוזת "Hello", ובהינתן שהאורך המבוקש הוא 10, תוחזר המחרוזת " Hello" – כלומר Hello וחמישה רווחים אחריו.

C#	PadRight method <code>myString.PadRight (width)</code>
VB 6	Use custom function: LeftJustify <code>function LeftJustify(myString as string, width as integer) as string</code> <code> dim rightPadding as string</code> <code> dim i as Integer</code> <code> for i = 1 to width - len(myString)</code> <code> rightPadding = rightPadding + chr(32)</code> <code> next</code> <code> LeftJustify = left(myString, width) + rightPadding</code> <code>end function</code>
VB .NET	PadRight method <code>myString.PadRight (width)</code> LSet function <code>LSet (myString, width)</code>

יישור טקסט לימין

בהינתן המחרוזת `myString`, ואורך מבוקש של הפלט `width`, נרצה להחזיר מחרוזת חדשה שמכילה את המחרוזת `myString`, ולפניה רווחים במספר כזה שיביאה לאורך המבוקש `width`. לדוגמא, בהינתן המחרוזת "Hello", ובהינתן שהאורך המבוקש הוא 10, תוחזר המחרוזת " Hello" – כלומר חמישה רווחים, שלאחריהם המילה Hello.

C#	PadLeft method <code>myString.PadLeft (width)</code>
VB 6	Use custom function: RightJustify function RightJustify(myString as string, width as integer) as string <pre> dim leftPadding as string dim i as Integer for i = 1 to width - len(myString) leftPadding = leftPadding + chr(32) next RightJustify = leftPadding + left(myString, width) </pre> end function
VB .NET	PadRight method <code>myString.PadRight (width)</code> RSet function <code>RSet (myString, width)</code>

יצירת מחרוזת אחת מתוך מערך מחרוזות

בהינתן מערך המחרוזות `myArray`, וכן תו מפריד, נרצה ליצור מחרוזת אחת שתכיל את כל המחרוזות של המערך ברצף, כשביניהן מפריד התו המפריד שצוין.

C#	Join method <code>String.Join(separator, myArray)</code>
VB 6	Join function <code>Join(myArray[, separator])</code>
VB .NET	Join method <code>String.Join(separator, myArray)</code> Join function <code>Join(myArray[, separator])</code>

יצירת מערך מחרוזות מתוך מחרוזת אחת

בהינתן המחרוזת myString ותו מפריד אחד או יותר, נרצה ליצור מערך של מחרוזות, שכל תא בו הוא תת מחרוזת של myString, שנחסם ב-myString על ידי התווים המפרידים.

C#	Split method <code>myString.Split ([separatorArray])</code>
VB6	Split function <code>Split(myString [, separator [, count [, compare]])</code>
VB .NET	Split method <code>myString.Split ([separatorArray])</code> Split function <code>Split(myString [, separator [, count [, compare]])</code>

המרת מחרוזת לאותיות קטנות

בהינתן המחרוזת myString, נרצה להחזיר מחרוזת שתכיל את אותו תוכן כמו myString, אולם אותיות גדולות (Capital Letters) במחרוזת myString יהפכו לאותיות קטנות במחרוזת התוצאה.

C#	ToLower <code>myString.ToLower ()</code>
VB 6	LCase / LCase\$ <code>LCase (myString)</code> <code>LCase\$ (myString)</code> LCase returns a variant (string) containing the contents of <i>myString</i> , with all uppercase characters converted to uppercase. LCase\$ does the same thing as LCase, except that it returns a string value instead of a string value contained within a variant. StrConv <code>StrConv (myString, vbLowerCase)</code> StrConv is a general-purpose string conversion function that can be used to convert to lowercase, among other things. <i>vbLowerCase</i> is a pre-defined constant whose value is 2.
VB .NET	ToLower <code>myString.ToLower ()</code> LCase / LCase\$ <code>LCase (myString)</code> <code>LCase\$ (myString)</code> LCase returns a variant (string) containing the contents of <i>myString</i> , with all uppercase characters converted to uppercase. LCase\$ does the same thing as LCase, except that it returns a string value instead of a string value contained within a variant. StrConv <code>StrConv (myString, VbStrConv.LowerCase)</code> StrConv is a general-purpose string conversion function that can be used to convert to uppercase, among other things. <i>VbStrConv.LowerCase</i> is a pre-defined constant.

המרת האות הראשונה בכל מילה במחרוזת לאות גדולה.

בהינתן המחרוזת myString, נרצה להחזיר מחרוזת שתכיל את אותו תוכן כמו myString, אולם בכל מילה, האות הראשונה במילה תוחלף ב-Capital Letter המתאימה.

VB 6	<p>StrConv function <code>StrConv(myString, vbProperCase)</code> Returns a string in which all letter characters at the start of each word are converted to uppercase. Valid word separators (and their character codes) are: NULL (0) TAB (9) LINEFEED (10) VERTICAL TAB (11) FORM FEED (12) RETURN (13) SPACE (32)</p>
VB .NET	<p>StrConv function <code>StrConv(myString, VbStrConv.ProperCase)</code> Returns a string in which all letter characters at the start of each word are converted to uppercase. Valid word separators (and their character codes) are: NULL (0) TAB (9) LINEFEED (10) VERTICAL TAB (11) FORM FEED (12) RETURN (13) SPACE (32)</p>

המרת מחרוזת לאותיות גדולות

בהינתן המחרוזת `myString`, נרצה להחזיר מחרוזת שתכיל את אותו תוכן כמו `myString`, אולם אותיות קטנות (Lowercase) במחרוזת `myString` יהפכו לאותיות גדולות במחרוזת התוצאה.

C#	ToUpper method <code>myString.ToUpper()</code>
VB 6	UCase / UCase\$ functions <code>UCase(myString)</code> <code>UCase\$(myString)</code> UCase returns a variant (string) containing the contents of <code>myString</code> , with all lowercase characters converted to uppercase. UCase\$ does the same thing as UCase, except that it returns a string value instead of a string value contained within a variant. StrConv function <code>StrConv(myString, vbUpperCase)</code> StrConv is a general-purpose string conversion function that can be used to convert to uppercase, among other things. <code>vbUpperCase</code> is a pre-defined constant whose value is 1. But really, why use this function when you've got UCase or UCase\$, which are more readable and take only one argument?
VB .NET	ToUpper method <code>myString.ToUpper()</code> UCase / UCase\$ functions <code>UCase(myString)</code> <code>UCase\$(myString)</code> StrConv function <code>StrConv(myString, VbStrConv.UpperCase)</code> StrConv is a general-purpose string conversion function that can be used to convert to uppercase, among other things. <code>VbStrConv.UpperCase</code> is a pre-defined constant.

השוואת מחרוזות

בהינתן שתי מחרוזות, `string1` ו-`string2`, נרצה להחזיר ערך שלם שיציין יחס בין המחרוזות, כמצוין בטבלה:

relationship between <code>string1</code> and <code>string2</code>	value returned by function
<code>string1 < string2</code>	negative
<code>string1 = string2</code>	0
<code>string1 > string2</code>	positive

ההשוואה בין מחרוזות היא לפי סדר לקסיקוגרפי.

C#	<p>Compare method String.Compare(<i>string1</i>, <i>string2</i> [, <i>ignoreCase</i> [, <i>culture</i>]]) <i>Compare</i> is case-sensitive by default. Case-sensitivity can be set through the optional boolean parameter <i>ignoreCase</i>; set it to false for a case-insensitive comparison or true for a case-sensitive one. The optional parameter <i>culture</i> (a CultureInfo object) is used when making string comparisons using non-English languages.</p>
VB	<p>StrComp function StrComp(<i>string1</i>, <i>string2</i> [, <i>mode</i>]) Case-sensitivity can be set through the integer parameter <i>mode</i>; set it to 0 (or the built-in constant vbBinaryCompare) for a case-sensitive comparison or 1 (or the built-in constant vbTextCompare) for a case-insensitive one.</p>
VB .NET	<p>Compare method String.Compare(<i>string1</i>, <i>string2</i> [, <i>ignoreCase</i> [, <i>culture</i>]]) <i>Compare</i> is case-sensitive. For a case-insensitive comparison, set the optional parameter <i>ignoreCase</i> (a boolean) to false. The optional parameter <i>culture</i> (a CultureInfo object) is used when making string comparisons using non-English languages.</p> <p>StrComp function StrComp(<i>string1</i>, <i>string2</i> [, <i>mode</i>]) Case-sensitivity can be set through the parameter <i>mode</i>; set it to CompareMethod.Binary for a case-sensitive comparison or CompareMethod.Text for a case-insensitive one.</p>

שרשור מהרזות

בהינתן שתי מהרזות, string1 ו-string2, נרצה להחזיר מהרזות חדשה, שתכיל את שתי המחרוזות אחת אחרי השנייה.

C#	<p>+ operator string1 + string2</p> <p>Concat function String.Concat(<i>string1</i>, <i>string2</i> [, <i>string3</i> [, <i>string4</i>]])</p> <p>+= operator string1 += string2 Appends the contents of <i>string2</i> to <i>string1</i>.</p>
----	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

STL's <i>string</i> class (C++)	+ operator <code>string1 + string2</code> += operator <code>string1 += string2</code> Appends the contents of <i>string2</i> to <i>string1</i> .
VB 6	& and + operators <code>string1 & string2</code> <code>string1 + string2</code> Although both & and + are string concatenation operators, & is the preferred one since there's no ambiguity, especially in the case of attempting to concatenate two string values contained in variants.
VB .NET	+ operator <code>string1 + string2</code> Concat function <code>String.Concat(string1, string2[, string3[, string4]])</code>

מחיקת תת מחרוזת מתוך מחרוזת

בהינתן מחרוזת `myString`, נקודת התחלה `start` ואורך `length`, נרצה להחזיר מחרוזת חדשה שכוללת את `myString` כאשר המחרוזת באורך `length` שמתחילה במיקום `start` הוסרה ממנה.

C#	Remove method <code>myString.Remove(start, length)</code>
STL's <i>string</i> class (C++)	<code>myString.erase(start, length)</code> Removes <i>length</i> characters from <i>myString</i> , starting at position <i>start</i> .
VB .NET	Remove method <code>myString.Remove(start, length)</code>

מציאת מופע ראשון

בהינתן מחרוזת `myString` ומחרוזת חיפוש `searchString`, נרצה להחזיר את מיקום התו הראשון של המופע הראשון של `searchString` בתוך `myString`, או ערך מיוחד, שהוא בדרך כלל -1, אם לא קיים מופע כזה.

C#	IndexOf method <code>myString.IndexOf(searchSubstring)</code>
----	-------------------------------------------------------------------------

STL's <i>string</i> class (C++)	<p>find method <code>myString.find(searchSubstring[, start])</code> Searches <i>myString</i> for the first occurrence of <i>searchSubstring</i> (which can either be a string or char) starting at <i>start</i> (if <i>start</i> is omitted, the search starts at the beginning of the string). If <i>searchSubstring</i> is found, it returns the position of the first character in <i>myString</i> where the match occurs. If not, it returns the value <code>string::npos</code>.</p> <p>rfind method <code>myString.rfind(searchSubstring[, start])</code> Like <code>find()</code>, except the search starts at the end of <i>myString</i> and proceeds from right to left.</p>
VB 6	<p>InStr function <code>InStr([start as number], phrase as String, searchstring as String [, compare as integer])</code> InStr returns a variant of subtype long specifying the character position of the first occurrence of <i>searchstring</i> in <i>string</i>. The first character position of a string is 1, not 0. InStr returns 0 if <i>string</i> is zero-length, <i>searchstring</i> is not found or <i>start</i> is past any occurrence of <i>searchstring</i>. InStr returns Null if either <i>string</i> or <i>searchstring</i> are Null. InStr returns <i>start</i> if <i>searchstring</i> is zero-length. <i>start</i> is an optional argument that specifies the character position where searching for substring in <i>string</i> should begin. The default is 1, referring to the first character in <i>string</i>. <i>compare</i> is an optional argument that specifies the type of string comparison used. The default value of 0 (or the built-in constant <code>vbBinaryCompare</code>) means that the comparison is binary – that is, case-sensitive. A value of 1 (or the built-in constant <code>vbTextCompare</code>) means that the comparison is textual and hence case-insensitive. If <i>compare</i> is omitted, the Option Compare setting determines the comparison type.</p> <p>InStrRev function <code>InStrRev(phrase as String, searchstring as String[, start as Integer[, compare as Integer]])</code> InStrRev is similar to InStr, except that its search is in the reverse direction (hence the name). InStrRev's search starts with the last character and proceeds towards the first. There are two other differences: The order of InStrRev's optional parameters differs from that of InStr. InStrRev returns a long while InStr returns a variant of subtype long.</p>
VB .NET	<p>IndexOf method <code>myString.IndexOf(searchSubstring)</code></p> <p>InStr function <code>InStr([start as number], phrase as String, searchstring as String [, compare as integer])</code></p> <p>InStrRev function <code>InStrRev(phrase as String, searchstring as String[, start as Integer[, compare as Integer]])</code> InStrRev is similar to InStr, except that its search is in the reverse direction (hence the name). InStrRev's search starts with the last character and proceeds towards the first. There are two other differences: The order of InStrRev's optional parameters differs from that of InStr. InStrRev returns a long while InStr returns a variant of subtype long.</p>

הוספת מחרוזות לתוך מחרוזות שניה

בהינתן מחרוזת `myString`, נקודת התחלה `start` ומחרוזת `insertString`, נרצה להחזיר מחרוזת חדשה הכוללת את `myString`, כאשר החל מהמיקום `start` מוכלת בה המחרוזת `insertString`.

C#	Insert method <code>myString.Insert(start, insertString)</code>
STL's <i>string</i> class (C++)	insert method <code>myString.insert(start, insertString)</code>
VB 6	
VB .NET	Insert method <code>myString.Insert(start, insertString)</code>

אורך מחרוזות

בהינתן מחרוזת `myString`, נרצה להחזיר את אורכה באותיות.

C#	Length property <code>myString.Length</code>
STL's <i>string</i> class (C++)	size method <code>myString.size()</code> length method <code>myString.length()</code> Both methods return a value of type <code>string::size_type</code> containing the number of characters in <code>myString</code> .
VB 6	Len function <code>Len(myString)</code> Returns a long containing the number of characters in <code>myString</code> . LenB function <code>LenB(myString)</code> Returns a long containing the number of bytes in <code>myString</code> .
VB .NET	Length property <code>myString.Length</code> Len function <code>Len(myString)</code> Returns an integer containing the number of characters in <code>myString</code> .

יצירת מחרוזת הבנויה מתווים חוזרים

נרצה ליצור מחרוזת המכילה רצף של תווים זהים באורך נתון `length`.

C#	<p>Use string constructor <code>string(char theChar, int length)</code> Creates <i>length</i> chars string containing <i>theChar</i>.</p>
VB 6	<p>String / String\$ functions <code>String(value as number, character as String)</code> <code>String\$(value as number, character as String)</code> String returns a variant (string) made up of <i>number</i> occurrences of <i>character</i> in a row. String\$ does the same thing as String, except that it returns a string value instead of a string value contained within a variant.</p> <p>Space / Space\$ functions <code>Space(value as number)</code> <code>Space\$(value as number)</code> Space returns a variant (string) made up of <i>number</i> occurrences of space characters in a row. Space\$ does the same thing as Space, except that it returns a string value instead of a string value contained within a variant.</p>
VB .NET	<p>Use custom function: RepeatedCharacters VB's string class doesn't have a method for creating a string of repeated characters, nor is there a built-in string function for creating a string made of repeated characters other than spaces. Use the function below.</p> <pre>Public Function RepeatedCharacters(myCharacter as Char, repetitions as Integer) as String Dim result as string result = "" RepeatedCharacters = result.PadLeft(repetitions, myCharacter); End Function</pre>

הפיכת מהזורת

בהינתן המחרוזת `myString`, נרצה להחזיר מהרוזת שתכיל את `myString` בסדר הפוך.

C#	ReverseString method C#'s string class doesn't have a method for reversing strings. Use the function below (note that it's been declared as static -- you'll probably want to put it in a string utility class). <pre>public static string ReverseString(string theString) { System.Text.StringBuilder theBuffer = new System.Text.StringBuilder(); int i; for (i = (theString.Length - 1); i >= 0; i--) { theBuffer.Append(theString[i]); } return theBuffer.ToString(); }</pre>
VB 6	StrReverse function <code>StrReverse(myString)</code> StrReverse reverses the order of characters in a string.
VB .NET	StrReverse function <code>StrReverse(myString)</code> StrReverse reverses the order of characters in a string.

מציאה והחלפה

בהינתן המחרוזת `myString`, מחרוזת חיפוש `searchString` ומחרוזת החלפה `replacementString`, נרצה להחזיר מהרוזת שתכיל את `myString` כך שכל המופעים של `searchString` שבה הוחלפו ב-`replacementString`.

C#	Replace method <code>myString.Replace(searchString, replacementString)</code>
VB 6	Replace function <code>Replace(myString, searchString, replacementString)</code> If <i>oldsubstring</i> is the empty string (""), Replace returns an unchanged copy of <i>myString</i> . If <i>newsubstring</i> is the empty string, Replace returns a copy of <i>string</i> with one or more occurrences of <i>oldsubstring</i> removed.
VB .NET	Replace method <code>myString.Replace(searchString, replacementString)</code> Replace function <code>Replace(myString, searchString, replacementString)</code>

הוצאת תת מחרוזת

בהינתן המחרוזת `myString`, מיקום התחלה `start` ואורך `length`, נחזיר מחרוזת חדשה שהיא חלק מ-`myString`, המתחילה ב-`start` ושארובה `length`.

C#	Substring method <code>myString.Substring(start[, length])</code>
STL's <i>string</i> class (C++)	substr method <code>myString.substr(start, length)</code> Returns a string created by copying <i>length</i> characters from <i>myString</i> starting at position <i>start</i> . Both <i>start</i> and <i>length</i> are integers. Note that position 0 denotes the first character in the string. If <i>start</i> + <i>length</i> is greater than <i>myString.size()</i> , the resulting string contains all the characters starting from position <i>start</i> to the end of the <i>myString</i> . The result is undefined if <i>start</i> is greater than <i>myString.size()</i> .
VB	Mid / Mid\$ functions <code>Mid(myString, start [,length])</code> <code>Mid\$(myString, start [,length])</code> Returns a variant (string) containing <i>length</i> characters from the middle of <i>string</i> , starting with character <i>start</i> . The first character position of a string is 1, not 0. If <i>length</i> is not specified, then <code>Mid</code> returns the rightmost portion of the string, starting at character <i>start</i> (it essentially becomes the <code>Right</code> function). Does the same thing as <code>Mid</code> , except that it returns a string value instead of a string value contained within a variant. Most texts on VB recommend the use of <code>Mid\$</code> over <code>Mid</code> , since it returns a string instead of a variant (theoretically yielding better performance and using less memory).
VB .NET	Substring method <code>myString.Substring(start[, length])</code> Mid function <code>Mid(myString, start [,length])</code>

הוצאת תת מחרוזת משמאל

בהינתן המחרוזת `myString` ואורך `length`, נחזיר מחרוזת חדשה שהיא חלק מ-`myString`, המתחילה בתו הראשון, ושארובה `length`.

C#	Use the Substring() method: <code>myString.Substring(0, length)</code>
STL's <i>string</i> class (C++)	Use the substr() method: <code>myString.substr(0, length)</code> Returns a string containing the <i>length</i> leftmost characters of <i>myString</i> .
VB	Left(myString, length) Left\$(myString, length) Left returns a variant (string) containing the <i>length</i> leftmost characters of <i>myString</i> . Left\$ does the same thing as <code>Left</code> , except that it returns a string value instead of a string value contained within a variant. Most texts on VB recommend the use of <code>Left\$</code> over <code>Left</code> , since it returns a string instead of a variant (theoretically yielding better performance and using less memory).

הוצאת תת מחרוזת מימין

בהינתן המחרוזת `myString` ואורך `length`, נחזיר מחרוזת חדשה שהיא חלק מ-`myString`, הכוללת את `length` התווים הימניים של המחרוזת המקורית.

C#	Use the Substring() method: <code>myString.Substring(myString.Length-length, length)</code>
STL's <i>string</i> class (C++)	Use the substr() and size() methods: <code>myString.substr(mystring.size() - length, length)</code> Returns a string containing the <i>length</i> rightmost characters of <i>myString</i> .
VB	Right(myString, length) Right\$(myString, length) Right returns a variant (string) containing the <i>length</i> rightmost characters of <i>myString</i> . Right\$ does the same thing as Right, except that it returns a string value instead of a string value contained within a variant. Most texts on VB recommend the use of Right\$ over Right, since it returns a string instead of a variant (theoretically yielding better performance and using less memory).

מחיקת רווחים מובילים

בהינתן המחרוזת `myString`, נרצה להחזיר מחרוזת זזה, בה מוסרים "whitespaces" מתחילת המחרוזת.

C#	TrimStart method <code>myString.TrimStart()</code>
VB 6	LTrim / LTrim\$ function <code>LTrim(myString)</code> <code>LTrim\$(myString)</code> LTrim returns a variant (string) containing a copy of <i>myString</i> without any leading spaces. LTrim\$ does the same thing as LTrim, except that it returns a string value instead of a string value contained within a variant.
VB .NET	TrimStart method <code>myString.TrimStart()</code> LTrim function <code>LTrim(myString)</code> LTrim returns a variant string containing a copy of <i>myString</i> without any leading spaces.

מחיקת רווחים לבנים

בהינתן המחרוזת `myString`, נרצה להחזיר מחרוזת זהה, בה מוסרים "whitespaces" מתחילת המחרוזת ומסופה.

C#	Trim method <code>myString.Trim()</code>
VB 6	Trim / Trim\$ methods <code>Trim(myString)</code> <code>Trim\$(myString)</code> Trim returns a variant (string) containing a copy of <code>myString</code> without any leading or trailing spaces. Trim\$ does the same thing as Trim, except that it returns a string value instead of a string value contained within a variant.
VB .NET	Trim method <code>myString.Trim()</code> Trim function <code>Trim(myString)</code>

מחיקת רווחים לבנים מסוף המחרוזת

בהינתן המחרוזת `myString`, נרצה להחזיר מחרוזת זהה, בה מוסרים "whitespaces" מסוף המחרוזת.

C#	TrimEnd method <code>myString.TrimEnd()</code>
VB 6	RTrim / RTrim\$ functions <code>RTrim(myString)</code> <code>RTrim\$(myString)</code> RTrim returns a variant (string) containing a copy of <code>myString</code> without any trailing spaces. RTrim\$ does the same thing as RTrim, except that it returns a string value instead of a string value contained within a variant.
VB .NET	TrimEnd method <code>myString.TrimEnd()</code> RTrim function <code>RTrim(myString)</code> RTrim returns a string containing a copy of <code>myString</code> without any trailing spaces.

EOF