

PASCAL - STRINGS

http://www.tutorialspoint.com/pascal/pascal_strings.htm

Copyright © tutorialspoint.com

The string in Pascal is actually a sequence of characters with an optional size specification. The characters could be numeric, letters, blank, special characters or a combination of all. Extended Pascal provides numerous types of string objects depending upon the system and implementation. We will discuss more common types of strings used in programs.

You can define a string in many ways –

- **Character arrays** – This is a character string which is a sequence of zero or more byte-sized characters enclosed in single quotes.
- **String variables** – The variable of String type, as defined in Turbo Pascal.
- **Short strings** – The variable of String type with size specification.
- **Null terminated strings** – The variable of **pchar** type.
- **AnsiStrings** – Ansistrings are strings that have no length limit.

Pascal provides only one string operator, string concatenation operator +.

Examples

The following program prints first four kinds of strings. We will use AnsiStrings in the next example.

```
program exString;
var
  greetings: string;
  name: packed array [1..10] of char;
  organisation: string[10];
  message: pchar;

begin
  greetings := 'Hello ';
  message := 'Good Day!';

  writeln('Please Enter your Name');
  readln(name);

  writeln('Please Enter the name of your Organisation');
  readln(organisation);

  writeln(greetings, name, ' from ', organisation);
  writeln(message);
end.
```

When the above code is compiled and executed, it produces the following result –

```
Please Enter your Name
John Smith
Please Enter the name of your Organisation
Infotech
Hello John Smith from Infotech
```

Following example makes use of few more functions, let's see –

```
program exString;
uses sysutils;
var
  str1, str2, str3 : ansistring;
  str4: string;
  len: integer;
```

```

begin
  str1 := 'Hello ';
  str2 := 'There!';

  (* copy str1 into str3 *)
  str3 := str1;
  writeln('appendstr( str3, str1) : ', str3 );

  (* concatenates str1 and str2 *)
  appendstr( str1, str2);
  writeln( 'appendstr( str1, str2) ' , str1 );
  str4 := str1 + str2;
  writeln('Now str4 is: ', str4);

  (* total length of str4 after concatenation *)
  len := byte(str4[0]);
  writeln('Length of the final string str4: ', len);
end.

```

When the above code is compiled and executed, it produces the following result –

```

appendstr( str3, str1) : Hello
appendstr( str1, str2) : Hello There!
Now str4 is: Hello There! There!
Length of the final string str4: 18

```

Pascal String Functions and Procedures

Pascal supports a wide range of functions and procedures that manipulate strings. These subprograms vary implement-wise. Here, we are listing various string manipulating subprograms provided by Free Pascal –

Sr.No.	Function & Purpose
1	function AnsiCompareStr <i>constS1; constS2: Integer;</i> Compares two strings
2	function AnsiCompareText <i>constS1; constS2: Integer;</i> Compares two strings, case insensitive
3	function AnsiExtractQuotedStr <i>varSrc: PChar; Quote: Char;</i> Removes quotes from string
4	function AnsiLastChar <i>constS: PChar;</i> Gets last character of string
5	function AnsiLowerCase <i>constS: ;</i> Converts string to all-lowercase
6	function AnsiQuotedStr <i>constS; Quote: Char;</i>

Quotes a string

7

function AnsiStrComp*S1: PChar; S2: PChar:Integer;*

Compares strings case-sensitive

8

function AnsiStrIComp*S1: PChar; S2: PChar:Integer;*

Compares strings case-insensitive

9

function AnsiStrLComp*S1: PChar; S2: PChar; MaxLen: Cardinal:Integer;*

Compares L characters of strings case sensitive

10

function AnsiStrLIComp*S1: PChar; S2: PChar; MaxLen: Cardinal:Integer;*

Compares L characters of strings case insensitive

11

function AnsiStrLastCharStr: *PChar:PChar;*

Gets last character of string

12

function AnsiStrLowerStr: *PChar:PChar;*

Converts string to all-lowercase

13

function AnsiStrUpperStr: *PChar:PChar;*

Converts string to all-uppercase

14

function AnsiUpperCase*constS: ;*

Converts string to all-uppercase

15

procedure AppendStr*varDest: ; constS: ;*

Appends 2 strings

16

procedure AssignStr*varP: PString; constS: ;*

Assigns value of strings on heap

17

function CompareStr*constS1: ; constS2: :Integer; overload;*

Compares two strings case sensitive

18

function CompareText*constS1: ; constS2: :Integer;*

Compares two strings case insensitive

19 **procedure DisposeStrS: PString; overload;**

Removes string from heap

20 **procedure DisposeStrS: PShortString; overload;**

Removes string from heap

21 **function IsValidIdentconstIdent: Boolean;**

Is string a valid pascal identifier

22 **function LastDelimiterconstDelimiters; ; constS: Integer;**

Last occurrence of character in a string

23 **function LeftStrconstS; ; Count: Integer;;**

Gets first N characters of a string

24 **function LoadStrIdent: Integer;;**

Loads string from resources

25 **function LowerCaseconstS: ; overload;**

Converts string to all-lowercase

26 **function LowerCaseconstV: variant;; overload;**

Converts string to all-lowercase

27 **function NewStrconstS: PString; overload;**

Allocates new string on heap

28 **function RightStrconstS; ; Count: Integer;;**

Gets last N characters of a string

29 **function StrAllocSize: Cardinal;PChar;**

Allocates memory for string

30 **function StrBufSizeStr: PChar;SizeUInt;**

Reserves memory for a string

31

procedure StrDisposeStr: PChar;

Removes string from heap

32

function StrPasStr: PChar;;

Converts PChar to pascal string

33

function StrPCopyDest: PChar; Source: :PChar;

Copies pascal string

34

function StrPLCopyDest: PChar; Source: ; MaxLen: SizeUInt:PChar;

Copies N bytes of pascal string

35

function UpperCaseconst: ::;

Converts string to all-uppercase

Processing math: 100%