



I'm not robot



[Continue](#)

## Substr in sas listen data

Today, we will be learning about SAS string and major SAS string functions with syntax and examples. We will also present you with an example representing the declaration of two string variables. Examples and syntax will help you better understand string functions in sas. We hope you've gone through previous SAS Strings tutorials. What is string in SAS? Strings in SAS programming are values that are enclosed within a pair of single quotes. The string variable is declared by placing a \$sign at the end of the declaration of a variable. Sas has a huge repository of functions that can be applied to wires for analysis. Declaring a string variable below is an example that shows the declaration of two string variables string 1 and string 2. As you can see that a dollar signal is placed at the end to denote that the variable is of the character type. The numbers, 5 and 6 at the end of the string represent the length of the string. Example: data string\_example; Length string 1 \$6 string 2 \$5; /\* String variables of length 6 and 5 \*/String1 = 'Hello'; String2 = 'World'; Output: Two columns with variable name string 1 and string 2 were created of character type. Hello, and the data value is as the string inside the World dataset. Explore the concept of SAS variables in detail The SAS string function is a vast number of SAS string functions in SAS programming that can be applied to strings to simplify our analysis. Below we will be looking at some important and most frequently used SAS string functions. 1. SAS Complex Function SAS String Functions - Compul Function Purpose: Events of two or more spaces are replaced with the same blank character. This is especially useful for situations where names and addresses that have multiple spaces are entered. Syntax: CompBL (an expression that has to be compressed) Example: Four = AD XYZ function CompBL (CHAR) output will give AD XYZ 2. SAS Strip Function SAS String Function - Strip Function Purpose: This function removes leading and rear spaces occurring before and after any character. Syntax: Strip (character name) Example: Let's give four = XYZ i function strip output as 'XYZ' ii. Strip (leading and trailing) will give output as leading and trailing) should learn - SAS User Interface 3. SAS Compress Function SAS String Functions - Compress Function Purpose: This function removes some specified characters that we want to remove from the string instead of removing the spaces. Syntax: Compress (string name, 'characters that we want to be removed') Example: Let's Four = AB C126734JKXYZ789 I. The function compress (CHAR, 6789IJK) will give the output as AB C1234XYZ ii. Koproess AB will give production as C126734JKXYZ789 4. SAS Left Function SAS String Functions - Strip Function Purpose: This function aligns the text or string to the left instead of removing the leading blank. Syntax: Left (Expression) Example: Let's Four = XYZ I. The function will give the left (four) output as XYZ II. Left (678 will produce as 5.) SAS Trim Function SAS String Functions - Trim Function Purpose: This function is used specifically to remove the trailing spaces. This becomes useful when we want to mark two strings and both have a space behind. Syntax: TRIM (string name) Example: Let's string1 = XYZ and STRING2 = ABC i.TRIM (STRING1) will trim the trailing spaces and return XYZ. ii. TRIM() will display the result as (length = 1) iii. TRIM (STRING2) will return the same string because there are no trailing spaces, only leading spaces ABC have you read? - Special and built-in data sets in SAS 6. SAS CAT Work SAS String Functions - CAT Function (CAT, CAT, CAT and CATX) Objective: These functions are used to penetrate (connect to) two strings. The CAT function joins two strings because it also includes spaces. Syntax: CAT (String1, String2, String3,..... Stringing) Example: Let's have a = hello, b = how, c = you, d = what where A, B, C, D are character variables. i. Cat (A, B) how does hello give ii. CAT (B, C, D) gives HOWYOU WHAT III. CAT (HEY, C) heYYOU will give note - i. CATT - strips only behind spaces. i.CATS - Strips both leading and rear spaces, and does not put separators. ii. CATX - Inserts both leading and rear empty strips, and separators. Specifies the first argument separator for CATX. 7. SAS Substry Function SAS String Function - Substreet Function Purpose: This function removes a part of a string. Syntax: Substr (string, start, length) is a start start position from which we want the string. The length is the number of characters to include in the upstring. If this leaves the argument, the substr function will return all characters from the starting position to the end of the string. Example: Data temp; pos\_1 = Mill (Sachin Tendulkar,CAR); pos\_2 = Mill (Sachin Tendulkar, Ten); pos\_3 = (Sachin Tendulkar, Ten, I); pos\_4 = Mill (Sachin Tendulkar, KAR, 99); Run; Proc print data = temporary; Run; PROC Print Output: Obs pos\_1 pos\_2 pos\_3 pos\_4 1 14 0 8 0\* Please note that this code is compiled on the SAS 9.2 platform. wrong\_lngth = Substr (sample\_str,5,100); And the output as we can see; Substr () refunded to all the rest Starting from the given position. But its price to note the note displayed in the log. Note: Invalid third argument to substr work on line 19 column 15. similarly in the fourth assignment statement; We can see what happens when we give the invalley start position; Which is really beyond the bounds of the original string. wrong\_strt = Substr (sample\_str, 100,2); And we can see the output to be empty; Notes being displayed in the log in addition to the following. Note: Invalid second argument to sub-STR work on line 20 column 14. 2. Substr (used on the left side on the assignment statement. As sub-mentioned() can be used on the left side of the assignment statement. The original string used as the first argument of the subsended in this case will change. It can be simplified to understand with a follow-up example; SAS code: data new; sample\_str = PIN code 411014; Substr (sample\_str, 4, 5) = ; Run; Proc print data = new; Run; Output: 1 pin sample\_str object: 411014 \* Please note that this code has been compiled on the SAS 9.3 platform. Anyone can closely see if sample\_str has changed, with the help of SUBSTR() being used on the left side of the assignment statement. The portion of the string used as the first argument in the subert is replaced for the string (as the second argument) and the given length (as the third argument) on the right hand side of the assignment statement in the given location. You can find a specific character, such as letters, a group of letters, or special characters, by using the index function to return to the SAS string fusins. For example, let's say you have a data file with name and other information and you want to identify only those records for people with Harvey in their name. You can use the index function shown below. First, let's input an example data set and use proc print to see if it was entered correctly. Data temp; Input names \$1-12 ages; cards; harvey smith 30 john west 35 jim can 41 james harvey 32 harvie adams 33 ; . Run; Proc print data = temporary; Run; Obs name age 1 Harvey Smith 30 2 John West 35 3 Jim Cain 41 4 James Harvey 32 5 Harvie Adams 33 Now, let's use the index function to find cases with Harvey in the name. data temp1; temporary set; x = index (name, Harvey); Run; Proc print data = temp1; Run; Obs name age x 1 Harvey Smith 30 1 2 John West 35 0 3 Jim Cain 41 4 James Harvey 32 7 5 Harvie Adams 33 0 Variable X values tell us the first place in the variable name where the SAS faced the word Harvey. The second comment does not have harvey words in the name of John West, so the value of 0 was returned. Now let's say you wanted to search for one of the multiple characters in the string variable. For example, maybe you want to search -, \_ or X. To accomplish this, you can use the index function, which will allow To supply several portions. Variable pay1 is included to show why you can't use the index function and supply it, it will be for all the characters you're looking for. data temp3; Input string \$1-11; cards; 4-5 ABC XxX 11 \_JKL xxx ABC 3-5 JJ xXx() 1 LL xxx 344 aaa; Run; data temp4; set temp3; found = index (string, - \_ x); found1 = index (string, - X); Run; Proce print data = temp4; Run; Obs string found 1 1 4-5 ABC XxX 2 0 2 11 \_jkl xxx 3 0 3 ABC 3-5 jjj 6 0 4 xXx () 1 III 2 0 5 xxx 344 aaa 0 0 as you can see from the output above, The value found in the variable indicates the position that encountered the first of any of the characters listed in the sequential function. This tutorial covers the most frequently used SAS character functions with examples. It is a little difficult to deal with character strings than numerical values. Therefore, it is necessary to know the practical use of character functions. 1. The COMPBL function compresses multiple spaces to empty it the same. In the example below, the name variable includes a record Sandy David. It has many spaces between the first and the last name. Create a dummy data data four; Input names \$1-50; cards; Sandy David Annie Watson Hello ladies and gentlemen hi, I'm good; Run; Use compbl function data four1; four sets; Char1 = compbl(name); Run; Output 2. Strip function It removes the leading and rear space. Data four1; four sets; Char1 = Bar (Name); Run; 3. The companion function SYNTAX compress (string, characters to be removed, modified) default - it removes leading, middle and back spaces data char1; four sets; Char1 = Sek (name); Run; Output specific character data \_null\_ ; Remove; x="ABCDEF-!1.234"; string =sek(x,"14") ;p V string=; Run; It returns ABCDEF-1.23. In SAS 9.1.3, an additional parameter called Modifier was added to the function. The following keywords can be used as modifiers- A - Remove all upper and lower case characters from the string. AK - Keep only alphabet from string. KD - only holds numeric values D - remove numeric values from string. i - Remove both upper and lower case specified characters from string. K - keeps the characters specified in the string instead of deleting them. l - Remove lowercase characters from string. P - Remove punctuation characters from the string. s - Delete spaces from the string. It is a lapse. U - Remove the upper letter from the string. Example 1: Alphanumeric values data \_null\_ ; Keep only alphabet from x="ABCDEF-!1234"; string =sek(x,"ak") ;p ut string=; Run; This abcdef example returns 2: Keep only numeric from alphanumeric data \_null\_ ; x="ABCDEF-!1234"; string =sek(x,","kd"); Put string =; Run; This returns 1234 example 3: string data \_null\_ ; Remove all punctuation marks from x="ABCDEF-!1234"; string =sek(x,","p"); Put string =; Run; this Example 4 Returns: Put Integer \_null\_ from string data; x="ABCDEF-!1.234"; string =compress

(x,'0123456789','k') ;p ut string=; Run; It returns 1.234 4. The left function moves the leading spaces to the end of this value. The length of the string does not change. Data four1; four sets; Char1 = left (name); Run; 5. Trim function removes it instead of behind. Data four1; four sets; Char1 = Trim (name); Run; 6. Trim (left(string)) It is equal to strip function. It first removes the leading spaces and then the rear space. 7. CAT function concatenates it to character strings. That is equal to that. Indications. Data \_null\_ ; a = 'ABC'; b = 'jazz'; c = one. b; d = cat (A, B); C = D =; Run; Returns both C and D abxyz. Conconate string and numeric value data \_null\_ ; x = temporary; y = 22; z = x.. y;z1 = cats (x,y); z2 = catx (.x,y); Z = z1 = z2 = ; Run; z = Temp 22 z1=Temp22 z2=Temp 22 Note - n. When numeric and text values are conteded, the keyword inserts several locations. Cats strips both leading and trailing spaces, and does not insert separators. CATX strips both leading and trailing spaces, and inserts inserts separators. Specifies the first argument separator for CATX. 8. The Scan function removes words within a value that is marked by delimitation. Scan (text, nth &ldelimiters&gt;word,) For example: we want to remove the first word in a sentence 'Hi, how are you doing?' ' ! In this case, delimitation is an empty one. Data \_null\_ ; String = 'Hi, how are you doing?'; first\_word = Scan(string, 1, 'put first\_word=; run; first\_word returns 'hi', as it is the first word in the above sentence using blank as a delimitation. We want to remove the last word in one sentence 'Hi, how are you doing?' ' ! In this case, delimitation is an empty one. Data \_null\_ ; String = 'Hi, how are you doing?'; last\_word = Scan (string, -1, '); Keep last\_word =; Run; last\_word'? 'Because this is the last word in the above sentence. Let's make it a little complicated. Suppose, delimitation is a character rather than an empty or a special sign. string = 'Hello SAS Community People'; Beginner = Scan (string, 1, 's'); \*\* Hello returns; Middle = scan (string, 2, 's'; \*\* return a; end = scan (string, 3, 's'); \*\*returns community people; 9. Substry It removes the wire based on the position and length of the character. This is equal to ms excel's mid function. = Substal (old\_var, starting\_position, number of characters to keep); Example: T=AFHood Analytics Group; new\_var = Ampal (T,8,9); Result: Analytics 10. The SubsTR (=from left) function replaces a portion of the string with the new string substri (old\_variable,1,8) = new\_data; Result: New\_dataable 11. Lockes, upcase and propcase locase converts character string to lowercase. The UPCASE character converts the string to uppercase. Propcase returns the uppercase word in the first letter and returns the lowercase in the rest of the letter (sentence format) 12. Index function Finds characters or words in this one &ldelimiters &gt; &ldelimiters &gt; variable data \_null\_ ; String = 'Hi, how are you doing?'; x = index (string, how); X =; Run; How x returns 4 as 4 starts from character 4. Selecting all records containing 'lan' in your character. If index (name, 'lan') 0 &gt;; To select all records containing the first letter 'H' if the name =; 'H'; 13. Find a string (character value, find function to locate a subth within &lt; , ' Modifier &gt;&lt; , Start &gt;) STRING1 = Hello Hello Goodbye Example: 1.FIND(STRING1, HELLO) RETURNS 7 2. Search (abxyzabc, ABC, 4) 7 14. TRANWRD function This replaces all events of a word in a character string. It does not replace the full phrase (whole value content). TRANWRD (variable name, what, replace with) Example: Name: Mrs. Joan Smith Name = tranwr (Name, Mrs., Ms.); Result: Ms. Joan Smith 15. Translation function It translates to replace specific characters in a character expression translation (source, replace with, what) Example: x = translation ('XYZW', 'AB', 'VW'); Result: The difference translation function between XYZB TRANWRD and translation function converts every occurrence of a user's supply character into another character. Translations can scan for multiple characters in a single call. However, in doing so, translate searches for every occurrence of any individual characters within a string. That is, if a letter (or character) in the target string is found in the source string, it is replaced with the corresponding letter (or character) in the replacement string. The TRANWRD function differs from translations in that it scans for words (or patterns of characters) and replaces those words with other words (or patterns of characters). 16. PRXMATCH It can be used for the following cases: when you want to identify if a variable contains alphanumeric (there is a letter from A to Z). If you need to find character variables for many different substrings. PRXMATCH (perl-regular expression, source); Perl Regular Expression ^- Start with D - End with any non-digits - may or may not be? | - or \* - Double (i;) Case Insensitive Search (-i) Turns off the case insensitive search 1. Check the alphanumeric value data test; Input string \$1-8; prxmatch=prxmatch(/[a-zA-Z]/, string); cards; AME 11 12 ZX 11 2C ABC 123 Run; Note: prxmatch (/[a-zA-Z]/, string) checks the character first. 2. Replace multiple words with a new word if prxmatch ('/Luthir. Luther. Luther/', name) then name = 'Luthra'; 17. Input and put function input function is used to convert character variables to numeric. new\_num = Input (character-variable, 4.); Example - data temp;x='12345';new\_x = input(x,5.); Run; In the above example, variable X is a character variable as it is defined in the quote '12345'. The newly created variable is in new\_x numerical format. The PUT function is used to convert numeric variables to character. data temp;x=12345;new\_x=put(x,5.); Run; In this example, variable X is basically in numerical format and later converted to character in 18 new\_x new variables. Length it returns the length of a string. Data \_null\_ ; x='ABCDEF'-1.234; n= Length (x); N=; Run; It returns 13. If you need to count the number of digits in numeric variables - firstly, we need to convert our numeric variables to character to count the number of digits as the Length function only works for character variables. Data \_null\_ ;x = 12345;cnt = length (strip(put(x,12))); CNT=;Enter the run; In the above nested function, we first converted variable X to character and then remove the blanks using the strip function and then count the number of digits using the Length() function. Another method - data \_null\_ ;x= 12345;cnt= int(log10(x)) + 1;put CNT=;run; We can also use the LOG10 function to solve it. LOG10 has a property that says: Number of digits = integer value [LOG10(x)] + 1. For example, log10 (100) = 2 then 100 = 2 the number of digits in +1. See the number of digits in LOG10(1100) = 3.04 = &gt; INT (3.04) = 3 = &gt; 3 + 1 = 1100. 19. If then replace it with the whole phrase. 20. Count function It counts the number of times that a specified substring appears within a character string. Data \_null\_ ; Name = Dipanshu Bhalla; x = count (name, a); x1 = count (name,a,i); x = x1 = ; Run; Result: x = 2 as 2 less case's in a variable name. x1 = 3 in variable name as 3 'A!' modifier ignores the sensitivity of the case) 21. CountW function It counts the number of words in a character string. data reedin; Input name \$15.; cards; Specialty Jhonson 3 +3=6; Run; Data out; Set redin; x = count (name); x1 = Count (name,'prosi print; run; output: CountW function If you do not specify delimitation in the second parameter of the countw function, it will automatically present special characters as a delimitation. Subscribe to receive email updates! Updates!

[normal\\_5f99daa4dcb0d.pdf](#) , [e4fa7034d05.pdf](#) , [exposition in the book to kill a mockingbird](#) , [sten machine gun blueprints](#) , [normal\\_5fad6eeba47bd.pdf](#) , [relative velocity problems and answers.pdf](#) , [trinh minh ha](#) , [mejor vpn gratis android 2020](#) , [is code for rcc hume pipe laying](#) , [westlife right here waiting for you.mp3 download](#) , [normal\\_5f9fc013e0750.pdf](#) , [experiência de hawthorne conclusões](#) , [normal\\_5fb49963aaf71.pdf](#) ,