

The 5 CATs in the Hat – Sleek Concatenation String Functions

Kirk Paul Lafler, sasNerd

Abstract

SAS functions are an essential component of the SAS Base software. Representing a variety of built-in and callable routines, functions serve as the "work horses" in the SAS software providing users with "ready-to-use" tools designed to ease the burden of writing and testing often lengthy and complex code for a variety of programming tasks. The advantage of using SAS functions is evident by their relative ease of use, and their ability to provide a more efficient, robust, and scalable approach to simplifying a process or programming task. SAS functions span several functional categories, including character, numeric, character string matching, data concatenation, truncation, data transformation, search, date and time, arithmetic and trigonometric, hyperbolic, state and zip code, macro, random number, statistical and probability, financial, SAS file I/O, external files, external routines, sort, to name a few. This SAS programming tip highlights the old, alternate, and new methods of concatenating strings and/or variables together.

Introduction

SAS functions span several functional categories, including character, numeric, character string matching, data concatenation, truncation, data transformation, search, date and time, arithmetic and trigonometric, hyperbolic, state and zip code, macro, random number, statistical and probability, financial, SAS file I/O, external files, external routines, sort, to name a few. This SAS programming tip highlights the old, alternate, and new methods of concatenating strings and/or variables together. The code, results and analysis appear below.

SAS Code:

```
data _null_ ;
  length NUM 3. A B C D E $ 8 BLANK $ 1 ;
  A = 'The' ;
  NUM = 5 ;
  B = ' Cats' ;
  C = 'in' ;
  D = ' the' ;
  E = 'Hat' ;
  BLANK = ' ' ;

* Old way of concatenating with TRIM and LEFT functions and concatenation operator ;
OLD = trim(left(A)) || BLANK || trim(left(NUM)) || BLANK || trim(left(B)) ||
      BLANK || trim(left(C)) || BLANK || trim(left(D)) || BLANK || trim(left(E)) ;

* Using the STRIP function and concatenation operator ;
STRIP = strip(A) || BLANK || strip(NUM) || BLANK || strip(B) || BLANK ||
       strip(C) || BLANK || strip(D) || BLANK || strip(E) ;

* Using the CAT functions to concatenate character and numeric values together ;
❶ CAT = cat (A, NUM, B, C, D, E) ;
❷ CATQ = catq(BLANK, A, NUM, B, C, D, E) ;
❸ CATS = cats(A, NUM, B, C, D, E) ;
❹ CATT = catt(A, NUM, B, C, D, E) ;
❺ CATX = catx(BLANK, A, NUM, B, C, D, E) ;
      put OLD= / STRIP= / CAT= / CATQ= / CATS= / CATT= / CATX= / ;
run ;
```

Results:

The 5 CATs in the Hat – Sleek Concatenation String Functions, continued

OLD = The 5 Cats in the Hat

STRIP = The 5 Cats in the Hat

CAT = The 5 Cats in the Hat

CATQ = "The " 5 " Cats " "in " " the " "Hat "

CATS = The5CatsintheHat

CATT = The5 Catsin theHat

CATX = The 5 Cats in the Hat

Analysis:

In the SAS code, above, a single numeric variable, NUM, and six character variables: A, B, C, D, E, and BLANK are defined with their respective values as: NUM=5, A='The', B=' Cats', C='in', D=' the', E='Hat' and BLANK=' '. The oldest way of concatenating two or more strings or variables together is then specified, using the TRIM and LEFT functions with the concatenation operator "||" in an assignment statement. An alternate approach using a TRIM function with the concatenation operator "||" is then specified in an assignment statement to join two or more strings or variables together. Finally, the newer and more robust concatenation approach is then specified using the CAT family of functions: CAT, CATQ, CATS, CATT, and CATX.

- ❶ **CAT**, the simplest of concatenation functions, joins two or more strings and/or variables together, end-to-end producing the same results as with the concatenation (double bar) operator.
- ❷ **CATQ** is similar to the default features of the CATX function, but the CATQ function adds quotation marks to any concatenated string or variable.
- ❸ **CATS** removes all leading and trailing blanks and concatenates two or more strings and/or variables together.
- ❹ **CATT** removes trailing blanks and concatenates two or more strings and/or variables together.
- ❺ **CATX**, perhaps the most robust CAT function, removes leading and trailing blanks and concatenates two or more strings and/or variables together with a delimiter between each.

Conclusion

SAS functions are an essential component of the SAS Base software. Representing a variety of built-in and callable routines, functions serve as the "work horses" in the SAS software providing users with "ready-to-use" tools designed to ease the burden of writing and testing often lengthy and complex code for a variety of programming tasks. In this paper I shared programming techniques related to concatenating strings. The oldest way of concatenating two or more strings or variables together is shown using the TRIM and LEFT functions with the concatenation operator "||" in an assignment statement. An alternate approach uses the TRIM function with the concatenation operator "||" in an assignment statement to join two or more strings or variables together. Finally, the newer and more robust concatenation approach is then specified using the CAT family of functions: CAT, CATQ, CATS, CATT, and CATX.

Acknowledgments

The author thanks the WUSS 2023 Conference Committee, particularly the Beginner's Essential Techniques Section Chair and Coordinator, Denis Nyongesa, for accepting my abstract and paper; the WUSS 2023 Academic Chair, Lida Gharibvand, and the Operations Chair, Julie Kilburn, for organizing and supporting a great "live" conference event; SAS Institute Inc. for providing SAS users with wonderful software; and SAS users everywhere for being the nicest people anywhere!

Trademark Citations

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.

About the Author

Kirk Paul Lafler is an educator, developer, programmer, consultant, and data analyst; currently working as a lecturer and adjunct professor at San Diego State University and the University of California San Diego Extension; and teaching SAS, SQL, Python, Excel, and cloud-based technology courses to users around the world. Kirk has decades of programming experience and specializes in SAS software, SQL, RDBMS technologies (Oracle, SQL-Server, Teradata, DB2), Python, and other languages and productivity tools. Kirk is the author of the popular PROC SQL: Beyond the Basics Using SAS, Third Edition (SAS Press. 2019) and is actively involved with SAS, SQL, Python, R, ML, and cloud-computing user groups, conferences, and blogs as an Invited speaker, educator, keynote, and leader; and is the recipient of 27 “Best” contributed paper, hands-on workshop (HOW), and poster awards.

Comments and suggestions can be sent to:

Kirk Paul Lafler, sasNerd
SAS® / SQL / RDBMS / Python / Cloud-based Consultant, Developer, Programmer,
Data Analyst, Educator and Author
sasNerd
E-mail: KirkLafler@cs.com
LinkedIn: <https://www.linkedin.com/in/KirkPaulLafler/>
Twitter: @sasNerd