# How can I remove leading zeros in SAS, and what are some examples of how to do so?

Authored by

**stats writer**

June 26, 2024

Removing leading zeros in SAS is a process used to eliminate any zeros that appear at the beginning of a character variable. This is often necessary when working with data that includes numerical values stored as character strings, as leading zeros can affect data analysis and comparisons. To remove leading zeros in SAS, the TRIM function can be used, along with other character functions such as COMPRESS and SUBSTR. An example of removing leading zeros in SAS would be using the TRIM function to remove leading zeros from a character variable containing a person's age, such as "012" being converted to "12". Other examples include removing leading zeros from zip codes or social security numbers. By utilizing SAS's character functions, leading zeros can be easily removed to ensure accurate and efficient data analysis.

## Remove Leading Zeros in SAS (With Examples)

## The easiest way to remove leading zeros in a character variable in SAS is to use the INPUT function to convert the variable to a numeric variable, which automatically removes leading zeros.

## This function uses the following basic syntax:

```
data new_data;
set original_data;
no_zeros = input(some_column, comma9.);
run;
```

## The following example shows how to use this syntax in practice.

### Example: Remove Leading Zeros in SAS

**Suppose we have the following dataset in SAS that shows the total sales made by various retail stores:**

```
/*create dataset*/
data original_data;
input store $ sales $;
datalines;
A 055
B 145
C 199
D 0000443
E 0093
F 00004302
G 38
H 0055
;
run;

/*view dataset*/
proc printdata=original_data;
```

| Obs | store | sales |
|---|---|---|
| 1 | A | 055 |
| 2 | B | 145 |
| 3 | C | 199 |
| 4 | D | 0000443 |
| 5 | E | 0093 |
| 6 | F | 00004302 |
| 7 | G | 38 |
| 8 | H | 0055 |

**We can use the following code to remove all leading zeros from values in the sales column:**

```
/*remove leading zeros in sales column*/
data new_data;
set original_data;
no_zeros = input(sales, comma9.);
run;

/*view results*/
proc printdata=new_data;
```

| Obs | store | sales | no_zeros |
|---|---|---|---|
| 1 | A | 055 | 55 |
| 2 | B | 145 | 145 |
| 3 | C | 199 | 199 |
| 4 | D | 0000443 | 443 |
| 5 | E | 0093 | 93 |
| 6 | F | 00004302 | 4302 |
| 7 | G | 38 | 38 |
| 8 | H | 0055 | 55 |

**Notice that all leading zeros have been removed from the values in the no_zeros column.**

**Note that the new no_zeros column is a numeric column.**

**If you would instead like to keep it as a character column, you can wrap the PUT function around the INPUT function as follows:**

```
/*remove leading zeros in sales column*/
data new_data;
set original_data;
no_zeros = put(input(sales, comma9.), 8.);
run;

/*view results*/
```

```
proc printdata=new_data;
```

| Obs | store | sales | no_zeros |
|---|---|---|---|
| 1 | A | 055 | 55 |
| 2 | B | 145 | 145 |
| 3 | C | 199 | 199 |
| 4 | D | 0000443 | 443 |
| 5 | E | 0093 | 93 |
| 6 | F | 00004302 | 4302 |
| 7 | G | 38 | 38 |
| 8 | H | 0055 | 55 |

**If we use use proc contents to view the data type of each variable in the dataset, we'll see that no_zeros is a character variable:**

```
/*view data type of each variable in new dataset*/
proc contentsdata=new_data;
```

| Alphabetic List of Variables and Attributes | | | |
|---|---|---|---|
| # | Variable | Type | Len |
| 3 | no_zeros | Char | 8 |
| 2 | sales | Char | 8 |
| 1 | store | Char | 8 |

**The following tutorials explain how to perform other**

## common tasks in SAS: