

# Learning to Add Titles to SAS Output: A Comprehensive Guide

Authored by  
**Mohammed loot**

October 31, 2025

## RECOMMENDED CITATION

Mohammed loot (2025). *Learning to Add Titles to SAS Output: A Comprehensive Guide*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=7292>

In the realm of data analysis and reporting, presenting your findings clearly and professionally is paramount. When working with [SAS](#), the ability to add descriptive titles to your output tables and charts is essential for enhancing readability and comprehension. The [TITLE statement](#) in [SAS](#) provides a straightforward yet powerful mechanism to achieve this, allowing you to quickly and accurately label your results and provide crucial context for any statistical output.

This comprehensive guide is designed to transform your reporting skills by exploring the depth and flexibility of the [TITLE statement](#). We will move beyond simply adding text, covering everything from crafting a single, simple heading to implementing multiple, elaborately styled lines of description. We will meticulously detail the programming [syntax](#), explore advanced customization options, and provide practical, runnable code examples to illustrate precisely how you can leverage titles to elevate the professionalism and clarity of your [SAS](#) reports.

## Understanding the Purpose of the SAS TITLE Statement

The core function of the [TITLE statement](#) is to specify one or more lines of descriptive text that are displayed prominently as titles at the very top of your [SAS](#) output. These headings are automatically generated and placed above any procedures that follow the statement in your program code. This functionality is immensely useful for distinguishing between different analytical segments, providing a clear heading for various report sections, or simply adding an immediate, informative label to a specific table or graph.

A critical characteristic of the ```TITLE``` statement is its scope and persistence: once defined, the titles remain active and are applied to all subsequent output generated by procedures until they are explicitly modified or cleared. This behavior ensures remarkable consistency, allowing the same report heading to be maintained across multiple analytical steps without redundant coding. Understanding this default persistence is essential for effective report management, preventing outdated or irrelevant titles from appearing on new output.

Furthermore, the [TITLE statement](#) provides significant styling flexibility. Users are not limited to defining only the text content; they can also apply a wide array of stylistic enhancements, including adjustments to color, font family, size, and text justification. This robust capability transforms basic text labels into integrated, customizable elements that are vital for creating professional and aesthetically pleasing data reports.

## Implementing a Single-Line Title (Basic Syntax)

The simplest and most common application involves adding a single title line to your [SAS](#) output. This is accomplished using the fundamental ```TITLE``` statement followed immediately by the desired text, which must be enclosed in single or double quotation marks. This method is perfectly suited for quick labeling or when your output only necessitates one clear, descriptive line at the

header.

The basic [syntax](#) is straightforward and highly accessible, even for beginner SAS programmers. Once executed, this single title will precede all subsequent SAS procedures that produce output, such as [PROC PRINT](#), ``PROC FREQ``, or ``PROC MEANS``, maintaining its visibility until a new ``TITLE`` statement is encountered later in the program or the current title is explicitly cleared.

The following code block demonstrates the essential structure for defining and applying a single title:

```
/*define title*/  
title "This is a title";  
  
/*view dataset with title*/  
proc print data=my_data;
```

In this example, the text "This is a title" will appear as the primary heading for the output generated by the [PROC PRINT](#) step. This simple, declarative approach ensures immediate and concise labeling, making your SAS output instantly more informative and easier to navigate for the audience.

## Creating Multi-Layered Headings (Advanced TITLE Usage)

For reports demanding greater detail, hierarchical structure, or multi-layered contextual information, [SAS](#) enables the specification of multiple titles through the use of numbered [TITLE statements](#), specifically ``TITLE1``, ``TITLE2``, ``TITLE3``, and so forth. Each numerical statement creates a distinct line of title text, which is displayed sequentially: ``TITLE1`` occupies the top position, followed immediately by ``TITLE2``, and so on down the page.

This tiered approach is particularly advantageous when you need to separate information, such as presenting a major report title, a specific subtitle detailing the analysis parameters, and a third line providing a date or source reference. Crucially, each numbered title line can be individually customized with unique styling [attributes](#), offering granular control over the visual hierarchy and overall appearance of your report headings.

Managing the visibility of these multiple titles is straightforward. To clear all currently active titles at once, you simply use the [TITLE statement](#) without any text or options (e.g., ``TITLE;``). Alternatively, if you only need to remove a specific line without affecting the others, you can use the numbered clear statement, such as ``TITLE2;``, which clears only the second title line.

Here is the [syntax](#) demonstrating how to define and style multiple, distinct title lines:

```
/*define titles*/  
title1 color="purple" height=25pt bold italic underline=1 "This is a Title";  
title2 font="Helvetica" justify=left height=18pt "Second Title";  
title3 color="green" justify=right height=14pt "Third Title";  
  
/*view dataset with title*/  
proc print data=my_data;
```

This code explicitly demonstrates how to apply unique styles to each title line. For instance, `TITLE1` is dramatically styled with purple color, boldness, italics, and underlining, establishing it as the main header. Meanwhile, `TITLE2` uses a specified font (Helvetica) and is left-justified. This detailed level of customization enables the creation of highly branded and expertly organized report outputs.

## Customizing Appearance with Style Attributes

The true expressive power of the [TITLE statement](#) is derived from its ability to accept various style [attributes](#) that fundamentally modify the appearance and layout of your titles. These [attributes](#) can be applied to the general `TITLE` statement or specifically to individual numbered titles (`TITLE1`, `TITLE2`, etc.), affording extensive control over the visual presentation. Below is a detailed breakdown of the most commonly utilized styling options:

**color:** This [attribute](#) controls the font color for the title text. You have the flexibility to use standard SAS color names (e.g., "red", "blue") or modern RGB hexadecimal color values for precise branding.

**height:** Used to define the font size of the title text, this is typically specified in points (pt) or sometimes inches (in). Adjusting the height, for instance using `height=25pt`, allows you to control the hierarchical prominence and visual weight of your titles relative to the output body.

**font:** This allows you to explicitly select a specific font family, such as "Times New Roman", "Arial", or "Helvetica". Specifying the font ensures that your titles maintain consistency with the overall aesthetic or official branding guidelines of your document.

**justify:** Controls the horizontal alignment of the title text on the page. Available options are `left`, `right`, or `center` (which is the default behavior). Proper justification helps position the title appropriately for optimal report layout.

**style:** This [attribute](#) applies various standard font styles. Common choices include `bold`, `italic`, or `underline`. These styles can also be combined (e.g., `bold italic`) to create visually striking titles. For underlining, you may specify `underline=1` for a single line.

By skillfully combining these powerful [attributes](#), you gain the ability to create highly customized and visually impactful titles. These titles not only convey essential information but also significantly

contribute to the professional presentation and overall clarity of your SAS analytical output. Experimentation is encouraged to achieve the perfect visual impact for your specific reporting needs.

## Practical Example 1: Adding a Simple Title to Output

To solidify the understanding of the basic application, let us walk through a practical example demonstrating how to add a single, simple title to a [dataset](#) output in [SAS](#). This scenario illustrates the foundational use of the `TITLE` statement, where a clear, descriptive heading is applied to a standard tabular report generated by a procedure.

Initially, we establish a sample [dataset](#) using the [DATA step](#). This dataset, named `original\_data`, contains basic hypothetical information simulating sports statistics--specifically team names, points scored, and rebounds achieved. After the data is successfully created and compiled, we invoke the [PROC PRINT](#) procedure to display the contents of the table, ensuring it is preceded by our single line of title text.

It is crucial to observe the precise placement of the `title` statement immediately before the analytical procedure. This placement guarantees that the specified text, "This is a title," appears prominently above the printed table, instantly providing context and a clear label for the viewer accessing the report output.

```
/*create dataset*/  
data original_data;  
input team $ points rebounds;  
datalines;  
A 25 10  
A 18 4  
B 27 9  
B 33 13  
;  
run;  
  
/*view dataset*/  
title "This is a title";  
proc print data=original_data;
```

This is a title			
Obs	team	points	rebounds
1	A	25	10
2	A	18	4
3	B	27	9
4	B	33	13

As confirmed by the visual output, the phrase "This is a title" is correctly rendered directly above the table generated by [PROC PRINT](#). This simple yet highly effective methodology ensures that every piece of your SAS output is appropriately labeled and immediately understandable to anyone reviewing the data.

## Practical Example 2: Styling and Layering Multiple Titles

When producing sophisticated reports, utilizing multiple, distinct titles with customized styling is crucial for conveying complex information and reinforcing report branding. This second example meticulously demonstrates how to define and apply unique style [attributes](#) across several title lines, resulting in a rich, informative, and visually appealing heading for your [SAS](#) reports.

We begin again by generating the familiar ``original_data`` [dataset](#). The core demonstration lies in the implementation of the ``TITLE1``, ``TITLE2``, and ``TITLE3`` statements, where each is assigned a specific combination of [attributes](#) controlling color, font, height, justification, and style effects. This approach naturally establishes a clear hierarchical presentation: ``TITLE1`` serves as the dominant main title, followed by ``TITLE2`` and ``TITLE3`` offering secondary and tertiary details, respectively.

Note the detailed styling: ``TITLE1`` is made prominent using purple text, increased height, and the combined effects of bold, italic, and underline. In contrast, ``TITLE2`` utilizes a specific font (Helvetica) and is set to be left-justified, providing a visual break. Finally, ``TITLE3`` is styled in green and is right-justified. This remarkable level of customization fully showcases the versatility and professional reporting capabilities inherent in the [TITLE statement](#).

```
/*create dataset*/  
data original_data;  
input team $ points rebounds;  
datalines;  
A 25 10  
A 18 4  
B 27 9
```

**B 33 13**

;

**run;**

```
/*view dataset*/
```

```
title1 color="purple" height=25pt bold italic underlin=1 "First Title";
```

```
title2 font="Helvetica" justify=left height=18pt "Second Title";
```

```
title3 color="green" justify=right height=14pt "Third Title";
```

```
proc print data=original_data;
```

The image shows the output of a SAS PROC PRINT statement. At the top, there are three titles: 'First Title' in purple, bold, italic, and underlined; 'Second Title' in blue; and 'Third Title' in green. Below the titles is a data table with four rows and four columns: Obs, team, points, and rebounds.

Obs	team	points	rebounds
1	A	25	10
2	A	18	4
3	B	27	9
4	B	33	13

The resulting output clearly displays the multi-layered and uniquely styled titles above the data table. This advanced application of the [TITLE statement](#) yields highly organized and visually appealing reports, transforming complex data presentation into a professional asset.

## Best Practices for Professional Report Titles

Effectively utilizing the [SAS TITLE statement](#) requires thoughtful application that extends beyond simply adding text; it is about strategic implementation to maximize the clarity and professionalism of your analytical reports. Adhering to certain best practices will ensure your titles enhance, rather than detract from, your data presentation.

**Clarity and Conciseness:** Always prioritize titles that are clear, concise, and accurately reflective of the output's content. Lengthy or ambiguous titles can confuse readers and diminish the report's effectiveness.

**Maintain Consistency:** Establish and uphold a consistent style, format, and hierarchy for titles throughout all related reports or documentation. Consistency builds reader familiarity and

reinforces the professional integrity of your work.

**Strategic Use of Hierarchy:** Leverage the numbered title statements (`TITLE1`, `TITLE2`, etc.) to create a logical flow of information. Reserve `TITLE1` for the main subject, use `TITLE2` for specific analytical subsets or parameters, and employ subsequent titles for supplementary details like data sources or execution dates.

**Judicious Customization:** While SAS offers abundant styling options, use them prudently. Overuse of colors, multiple fonts, and excessive bolding can lead to a cluttered appearance. Choose styles that genuinely enhance readability and complement the overall design of your report.

**Remember to Clear Titles:** Due to the persistent nature of the title statements, it is vital to clear them when they are no longer relevant to upcoming procedures. A simple `TITLE;` statement will universally clear all active titles, preventing them from bleeding onto unrelated output. Alternatively, using `TITLEN;` (e.g., `TITLE3;`) will clear a specific single numbered title.

Mastering the robust [TITLE statement](#) is an essential skill for any [SAS](#) user focused on producing high-quality, easily readable reports. By integrating these guidelines and experimenting with the various customization options, you can significantly improve the presentation and impact of your analytical results.

For those looking to expand their SAS knowledge further, the following tutorials explain how to perform other common tasks and unlock more capabilities within the software:

[SAS Tutorial: Creating Custom Formats](#)

[SAS Tutorial: Merging Datasets](#)

[SAS Tutorial: Using PROC FREQ for Frequency Tables](#)