



# getline()



Remember to free the returned string to avoid memory leaks.

The `getline()` function is used to read a line from a terminal with advanced editing capabilities, such as line editing, history, and tab completion. It's defined in the "getline/getline.h" system header.

It is part of the GNU Readline library, which is commonly used in interactive command-line applications.

```
#include <getline/getline.h>

// "prompt" is a string displayed as a prompt to the user
//      if NULL, no prompt is displayed
char *getline(const char *prompt);

// the library must be linked during compilation
cc program.c -o program -l readline
```

Returns a dynamically allocated string containing the input line. If the input line contains a newline character, includes it.

Returns NULL in case of error or EOF.

- Reads input from standard input stream
  - A file descriptor can be specified, to read from it instead
- Supports line editing features (arrow keys, backspace, history navigation...)
- Can be configured to expand tab-based completions
- Can store input in a history buffer, with help of "add\_history()"

## Example Usage

```
#include <stdio.h>
#include <stdlib.h>
#include <readline/readline.h>
#include <readline/history.h>

int main(void)
{
    char *input;

    while ((input = readline("> ")) != NULL)
    {
        if (*input)
        {
            printf("You entered: %s\n", input);
            add_history(input);
        }
        free(input);
    }
    return (0);
}
```