

# Function fgets () in C

Char \* fgets (char \* str, int n, FILE \* stream) in Standard C library reads a line from the given Stream and stores it in the string pointed by str. It stops when any of the following conditions are encountered: (n-1) the character has been read, the newline character (new line) is read or encountered end-of-file.

**Char \* fgets (char \* str, int n, FILE \* stream)** in Standard C library reads a line from the given Stream and stores it in the string pointed by str. It stops when any of the following conditions are encountered: (n-1) the character has been read, the newline character (new line) is read or encountered end-of-file.

## Declare function fgets () in C

Below is the declaration for fgets () in C:

```
char * fgets ( char * str , int n , FILE * stream )
```

### Parameters

**str** - This is the pointer to an array of char where the string to be read is stored.

**n** - This is the maximum number of characters to be read (including the last null character). Usually the length of the passed array is str used.

**stream** - This is the pointer to a FILE object that identifies the Stream, where characters are read from.

### Returns the value

If successful, the function returns the same parameter str. If EOF is encountered and no characters have been read, the contents of str remain unchanged and a null pointer is returned.

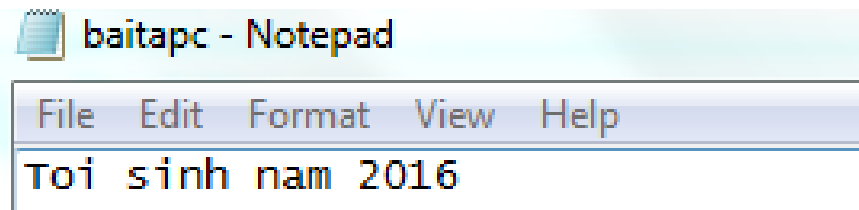
If an error occurs, a null pointer is returned.

### For example

The following program C illustrates the usage of the function fgets () in C:

```
#include int main () { FILE * fp ; char str [ 60 ]; /* mo file de doc */ fp
```

Suppose we have baitapc.txt with the following content. This file will be used as input for C program for example:



Compiling and running the above C program will result:



### According to Tutorialspoint

Previous lesson: Function fgetc () in C

Next lesson: Function fputc () in C

You finished reading the article "**Function fgetc () in C**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.