**Strings**

We define **strings** under the **data segment**.

.DATA  
someString DB "This is a string.", 0DH, 0AH, '$'

ASSEMBLY

Notice that we have also included a **line feed** and a **character return**, with the parts separated by **commas**. Also notice that we placed a $ character at the end. This symbol denotes the end of the string. Essentially, we have concatenated four strings.

To **display** the string, we need to use **function**  along with INT 21H. In the DX register, we place the string using a special instruction, LEA.

MOV AH, 9  
LEA DX, someString  
INT 21H

ASSEMBLY

LEA stands for **load effective address**. This loads the **offset address** of the first element of the string. INT 21H works with function to print the string character by character, starting at the offset address defined in the DX register, until it finds the $ character. This is also why we used DB when declaring the string instead of DW, since we need to go byte by byte instead of word by word (which is two bytes at a time).

The complete code will look like this:

ORG 0100H  
  
.DATA  
someString DB "This is a string.", 0DH, 0AH, '$'  
  
.CODE  
MAIN PROC  
 MOV AH, 9  
 LEA DX, someString  
 INT 21H  
MAIN ENDP  
END MAIN  
RET

ASSEMBLY