Current Skill String.

String

A string is a sequence of characters. In other words, a string is an array of character data type. It is the most used data structures ever.

Declaring a string is as simple as declaring a one dimensional array. Below is the basic syntax for declaring a string.

```
str_name : STRING[size];
```

In the above syntax str_name is any name given to the string variable and size is used define the length of the string, i.e the number of characters strings will store.

Initializing a String

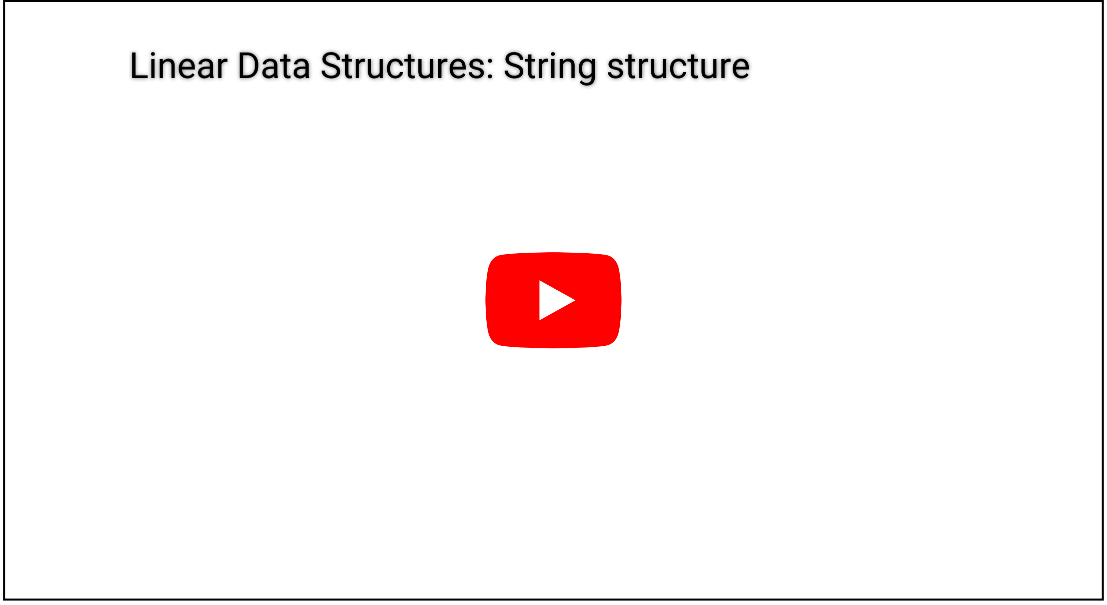
A string can be initialized in different ways. We will explain this with the help of an example. Below is an example to declare a string with name as str and initialize it with “GoMyCode”.

```
str : STRING[] := "GoMyCode";
```

```
str : STRING[50] := "GoMyCode";
```

```
str : STRING[] := {'G','o','M','y','C','o','d','e'};
```

There are many function that we use directly with String such as, Concat, ToLower, ToString, ToInteger, ToFloat.


Linear Data Structures: String structure

String

Now, we are going to practice some of what we've learned about string.

We will make three examples of string manipulation.

The first one is about comparing two string.

We can browse a string like we browse an array, we have simply to call the string identifier, with an index inside brackets.

Let’s see the code below :

```
ALGORITHM compare_two_strings
VAR
   str1, str2, : STRING[50];
   i : INTEGER;
BEGIN
   Write("Give the first string to compare");
   Read(str1);
   Write("Give the second string to compare");
   Read(str2);

   IF (str1.length <> str2.length) THEN
   // if the length of the two string is different we can make sure that they are no
       Write("The Strings are not equals");
   ELSE
       FOR i FROM 0 TO str1.length-1 STEP 1 DO
```

```
            IF (str1[i]<>str2[i]) THEN
                BREAK;// we break if in the same position the caracters of two string:
            END_IF
        END_FOR
        IF (i = str1.length) THEN
            Write("The Strings are equals");
        ELSE
            Write("The Strings are not equals");
        END_IF
    END_IF

END
```

Second, let's suppose that we have the same string but the first one is upper case, the algorithm will return that the two string are not equal.

To make sure that we escape this corner case, we need to make sure that we convert the two strings into an uppercase or lowercase.

Let’s see the code below :

```
ALGORITHM compare_ignore_two_strings
VAR
    str1, str2, : STRING[50];
    i : INTEGER;
BEGIN
    Write("Give the first string to compare");
    Read(str1);
    Write("Give the second string to compare");
    Read(str2);

    IF (str1.length <> str2.length) THEN
        Write("The Strings are not equals");
    ELSE
        str1 := ToUpper(str1); // this function will convert the character into upper
        str2 := ToUpper(str2);
        FOR i FROM 0 TO str1.length-1 STEP 1 DO
            IF (str1[i]<>str2[2]) THEN
                BREAK;
            END_IF
        END_FOR
        IF (i = str1.length-1) THEN
            Write("The Strings are equals");
        ELSE
            Write("The Strings are not equals");
        END_IF
    END_IF

END
```

In the third and last algorithm, we are going to remove the blanks from the beginning of a given string.

Let's take a look at this algorithm:

```
ALGORITHM delete_blank_begin
VAR
    str : STRING[] := "   GoMyCode";
    i : INTEGER := 0;
    j : INTEGER := 0;
BEGIN
    WHILE ( str[0]=' ') DO
        j := 0;
        WHILE (j < str.length) DO
            str[j] := str[j+1]; // translation from right to left
            j := j+1; // update index
        END_WHILE
    END_WHILE
END
```

‹ Previous next ›