

**ARB06**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> ARB06		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 31, 2024	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>ARB06</b>	<b>1</b>
1.1	ARexx For Beginners - Article 6 - Centring Text . . . . .	1

# Chapter 1

## ARB06

### 1.1 ARexx For Beginners - Article 6 - Centring Text

AREXX FOR BEGINNERS

ARTICLE 6 - CENTRING TEXT OUTPUT

BY FRANK BUNTON

COPYRIGHT © FRANK P. BUNTON 1995-1998

ARexx supports a centring function `CENTER()` or `CENTRE()`. Note that both the American and British spellings are accepted!

We start to talk more fully about Functions in Article 16 but this one goes hand in hand with the SAY instructions so I will talk about it now.

Its format is:-

```
CENTRE(string,length[,pad])
```

"String" is the string to be centred.

"Length" is the final length of the string that will be output.

"Pad" is the character that will be used at the start and end of the output string to make up the requested length. The square brackets [] indicate that this is an optional item. If it is left out the default character, a space, is used.

Only one character can be used as a pad. If more than one appears in the pad string then only the first is accepted.

To illustrate this, let's have some examples. I will use the "." in lieu of the default pad character (space) as it would be hard to see what happens without some indication! So where you see a "." just pretend it is a space. The output is after the --->

```
SAY CENTRE('Test String',20) ---> .....Test String.....
```

```

SAY CENTRE('Test String',20,'*') ---> ****Test String*****
SAY CENTRE(' Test String ',18,'*') ---> ** Test String **
SAY CENTRE(' Test String ',20,'*!') ---> *** Test String ****
SAY CENTRE(' Test String ',5,'*') ---> st St

```

The second last example shows that only the first character in the "pad" is used.

The last example illustrates that, if the length specified is less than the length of the string, then the string is cut from both ends so that only the centre most characters (5 in this case) are used.

Try out this program:-

```

/* Example6-1.rexx */
DO Len = 1 to 20
  SAY CENTRE(' Test String ',Len,'*')
END

```

"Len" is a symbol that can hold numbers of any value, in this case all integer numbers between 1 and 20. Symbols are discussed in Article 7.

I will talk about DO & END in Article 10 but, briefly, what it does in this format is to allow the same line (that between DO and END) to be executed 20 times. For each of the twenty times the symbol "Len" has a new value which increments from 1 to 20 in steps of 1. The string:-

```
' Test String '
```

has 13 characters (including the spaces) so that while "Len" is 13 or less, no \*s are added. However, when "Len" starts to exceed 13, then \*s are added. The output should be:-

```

S
S
St
t St
t Str
st Str
st Stri
est Stri
est Strin
Test Strin
Test String
Test String
Test String
Test String *
* Test String *
* Test String **
** Test String **
** Test String ***
*** Test String ***
*** Test String ****

```

There are lots of other things that we can do with strings like stripping off leading and trailing spaces, or all spaces, or all of certain character occurrences, or selecting certain parts of a string but we will leave

all this to another time.

=== End of Text ===

---