

A SIMPLE STRING EDITOR

Edit Functions

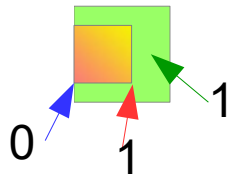
Insert/Append, Delete and Backspace

Navigation Functions

Left & Right Arrows

The String to be edited will be initialized to be one character long with that character being the SPACE character, `ascii(32)`

The cursor is placeholder that is initially defined as a character space (`ascii 32`) and is always highlighted.



CursorBegin



CursorEnd



strCurrentTextLength



strRightLength



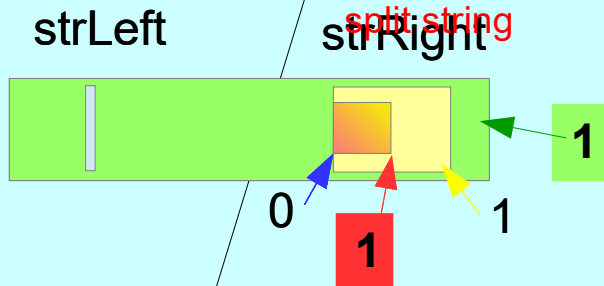
Cursor Character Placeholder

The last character is the cursor placeholder character and can NEVER be deleted

OnNewCharacter("A")

```
//Split the current string  
strLeft = strCurrentText.substring(0, CursorBegin);  
//include the selection on the right  
strRight = strCurrentText.substring(CursorBegin, strCurrentText.length());
```

On any EDIT function (APPEND, INSERT, DELETE) the current string is initially split such that the Cursor and its underlying character is ALWAYS the first character of the Right Portion of the split string

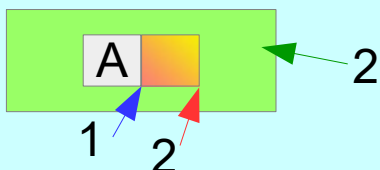


EDIT Function

CONDITION implies APPEND

`CursorEnd = strCurrentTextLength`

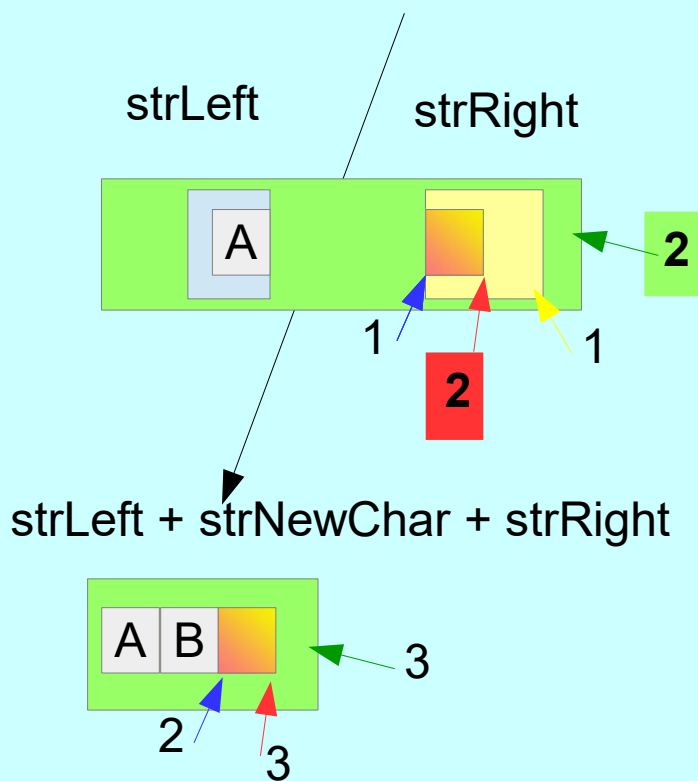
strLeft + strNewChar + strRight



ACTION

`strLeft + strNewChar + strRight`
`CursorPosition++`

OnNewCharacter("B")



EDIT Function

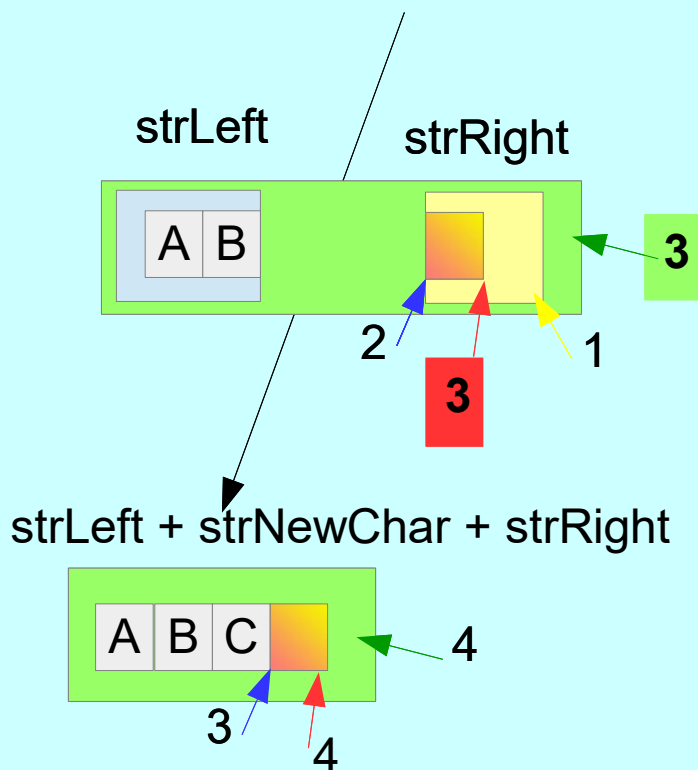
CONDITION implies APPEND

`CursorEnd = strCurrentTextLength`

ACTION

`strLeft + strNewChar + strRight`
`CursorPosition++`

OnNewCharacter("C")



EDIT Function

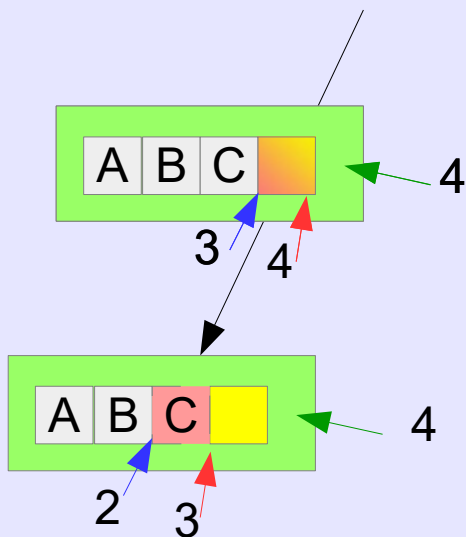
CONDITION implies APPEND

`CursorEnd = strCurrentTextLength`

ACTION

`strLeft + strNewChar + strRight`
`CursorPosition++`

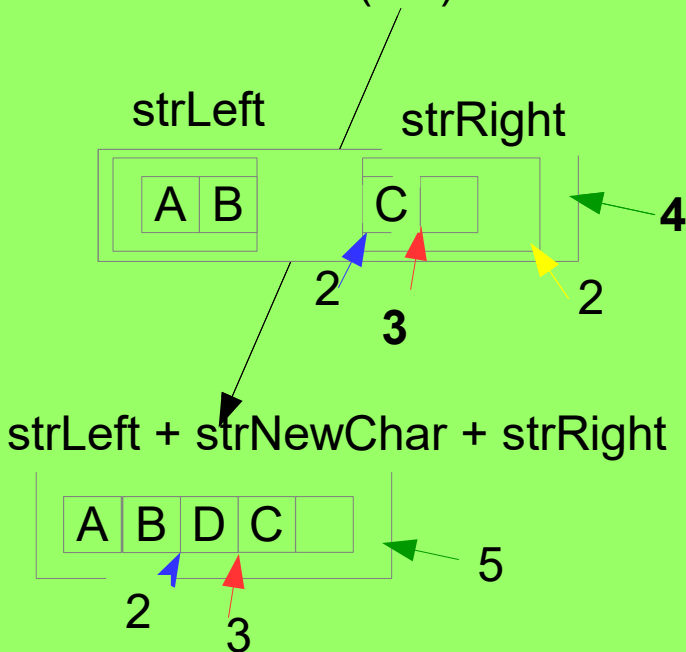
OnNewCharacter(LeftArrow)



Navigation Function
Left Arrow

ACTION
CursorPosition--

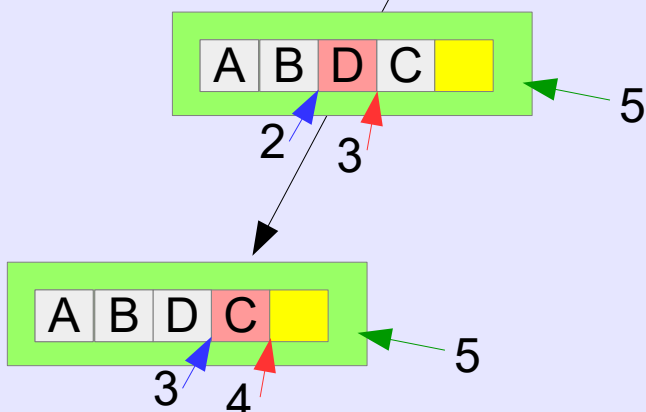
OnNewCharacter("D")



EDIT Function
CONDITION implies INSERT
CursorEnd < strCurrentTextLength

ACTION
strLeft + strNewChar + strRight
*NOTE CURSOR COORDINATES DO NOT CHANGE

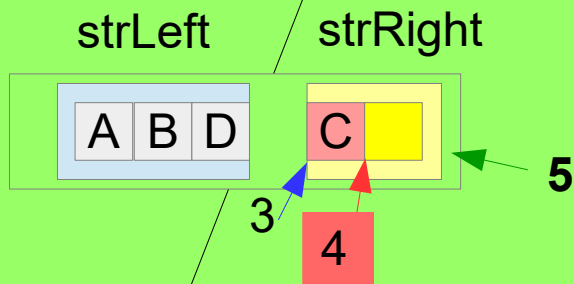
OnNewCharacter(RightArrow)



Navigation Function
Right Arrow

ACTION
CursorPosition++

OnNewCharacter("E")

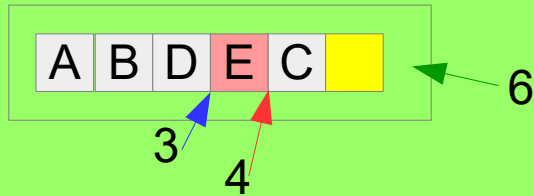


EDIT Function

CONDITION implies INSERT

`CursorEnd < strCurrentTextLength`

strLeft + strNewChar + strRight

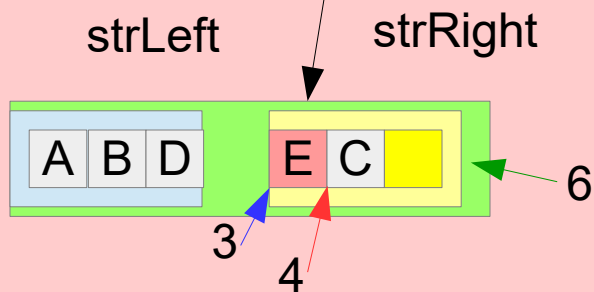


ACTION

`strLeft + strNewChar + strRight`

*NOTE CURSOR COORDINATES DO NOT CHANGE

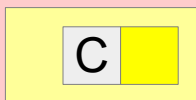
OnNewCharacter(Delete)



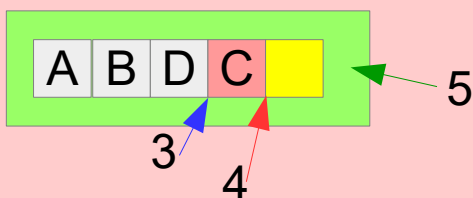
Delete Function

The last character is the **cursor place holder** character and can NEVER be deleted

`strModifiedRight = strRight.substring(1, strRight.length())`



strLeft + strModifiedRight



ACTION

`strModifiedRight = strRight.substring(1, strRight.length())`

`strNewString = strLeft + strModifiedRight`

*NOTE CURSOR COORDINATES DO NOT CHANGE

NOTES

The cursor is a placeholder that is initially defined as a character SPACE (ascii 32) and is ALWAYS highlighted. This initial SPACE character can NEVER be deleted AND is ALWAYS highlighted IF the string being edited is empty. Any character in the string being edited can be highlighted, thus identifying the cursors current placeholder.

On any EDIT function (APPEND, INSERT, DELETE) the current string is initially split such that the Cursor is ALWAYS the first character of the Right Portion of the split string

The Cursor Coordinates are ALWAYS and ONLY modified by the LEFT ARROW and RIGHT ARROW Navigation characters AND the APPEND Function

The new string resulting from the APPEND or INSERT Function is ALWAYS the concatenation of the Left portion of the split string + the New Character + the Right portion of the split string

The DELETE Function always operates on the character under the cursor which is ALWAYS the first character of Right portion of the split string

The BACKSPACE is simply a LEFT ARROW + DELETE

The CursorBegin coordinate is never referenced; however the coordinate is required as parameters for the ***EditText.setSelection(begin, end)*** Function