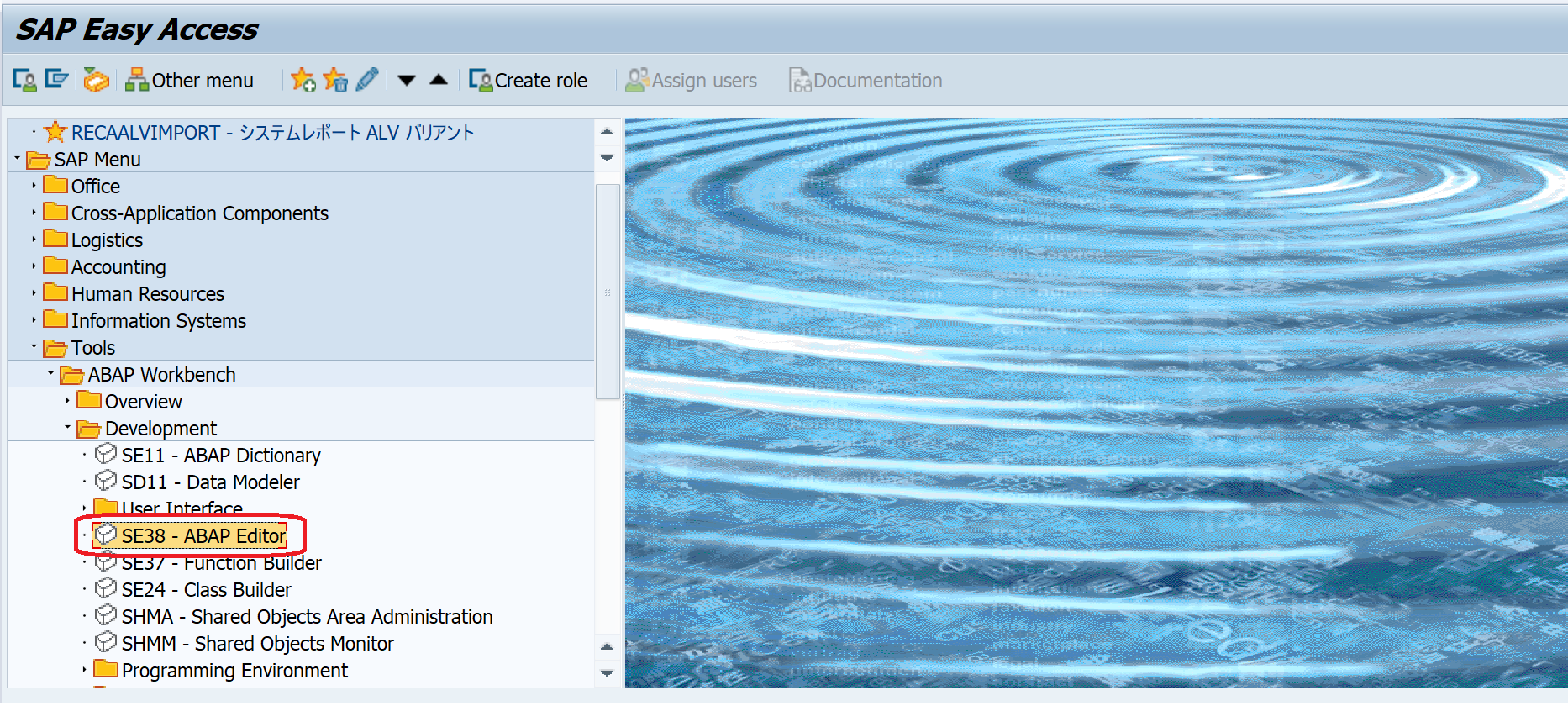
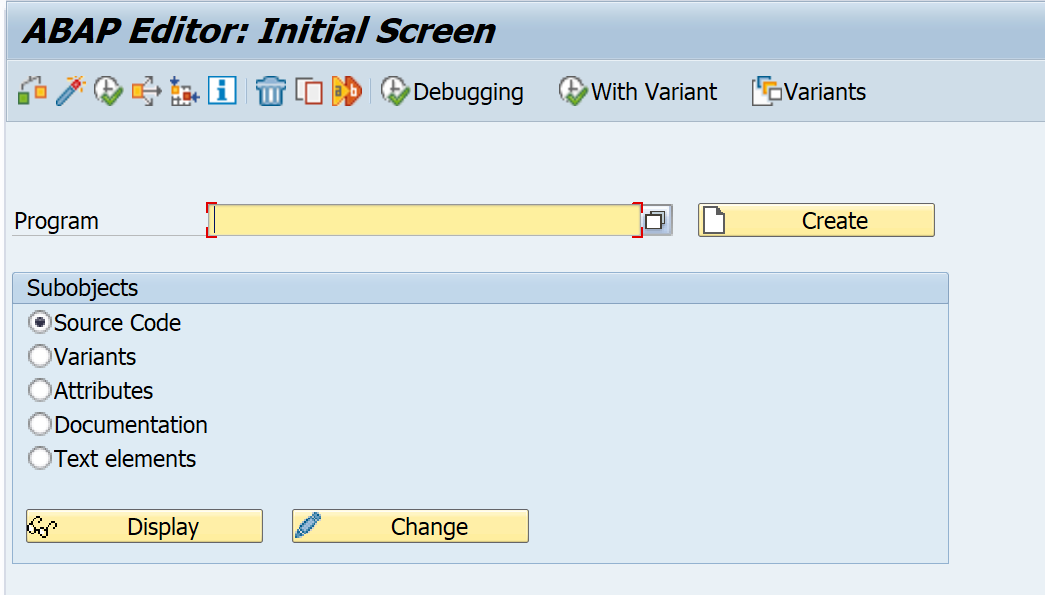
# ABAP Statements

## Create the program

* Go to Tools -> ABAP Workbench -> Development SE38 – ABAP Editor



* Or we can search SE38 straight in the search box and we will see this window appear



## Intro to WRITE statement

* Let’s begin with WRITE statement with some indentation

**REPORT** zalvgrid\_v0.

**DATA**: text **TYPE** string **VALUE** '0123456789ABCDEF',

col **TYPE** i **VALUE** **25**,

len **TYPE** i **VALUE** **5**.

**WRITE** text. "write full text

**WRITE** /**5**(**10**) text. "write indent 5 with len of 10

**WRITE** **AT** col(len) text. "same same but if with pre-defined variables

Typically, if you want to output something out to the screen, speaking of classical report, we need to write the screen so WRITE statement will be the best option among thousands

## Data (Variable) declaring

Let consider this variable declaring in Python

x = **5**

y = "John"

Now, let put it in the ABAP context, we will have this below

**DATA**: x **TYPE** i **VALUE** **5**,

y **TYPE** STRING **VALUE** "John".

## Constant declaring

CONSTANT con1 **TYPE** p **DECIMALS** **1** **VALUE** '6.6'.

## String concatenate

**DATA**: t1 **TYPE** c **LENGTH** **10** **VALUE** 'We',

t2 **TYPE** c **LENGTH** **10** **VALUE** 'have',

t3 **TYPE** c **LENGTH** **10** **VALUE** 'all',

t4 **TYPE** c **LENGTH** **10** **VALUE** 'the',

t5 **TYPE** c **LENGTH** **10** **VALUE** 'time',

t6 **TYPE** c **LENGTH** **10** **VALUE** 'in',

t7 **TYPE** c **LENGTH** **10** **VALUE** 'the',

t8 **TYPE** c **LENGTH** **10** **VALUE** 'world',

space.

**CONCATENATE** t1 t2 t3 t4 t5 t6 t7 t8

**INTO** **DATA**(result).

**CONCATENATE** t1 t2 t3 t4 t5 t6 t7 t8

**INTO** result **SEPARATED BY** space.

## Condense

**DATA**: **BEGIN OF** sentence,

word1 **TYPE** c **LENGTH** **30** **VALUE** 'She',

word2 **TYPE** c **LENGTH** **30** **VALUE** 'feeds',

word3 **TYPE** c **LENGTH** **30** **VALUE** 'you',

word4 **TYPE** c **LENGTH** **30** **VALUE** 'tea',

word5 **TYPE** c **LENGTH** **30** **VALUE** 'and',

word6 **TYPE** c **LENGTH** **30** **VALUE** 'oranges',

**END OF** sentence,

text **TYPE** string.

text = sentence.

write text “first write

**CONDENSE** text.

write text “second write

At the first write we will see the result with some additional spaces such as “She (…) feeds (…) … “but at the second write the text is condensed so all spaces will be shrinked to 1 space only.

Then condense means: “Shorten the spaces to one space only.” Let consider the short program below

**DATA** sentence **TYPE** c **length** **30** **VALUE** 'she love'.

**WRITE** sentence.

**CONDENSE** sentence.

**WRITE** sentence.

The sentence before condense will be ‘she love’ then after condense, it will become ‘she love’

This keyword can be use to standardize the output string, make the string clearer

## Find length of a string

Use keyword STRLEN() to find length of string

**DATA** search\_word **TYPE** string **VALUE** 'hello'.

**WRITE** / strlen( search\_word ).

The returned result will be 5. But let consider this short code below here

**DATA** search\_word **TYPE** c **LENGTH** **300** **VALUE** 'hello'.

**WRITE** / strlen( search\_word ).

You will see the length of 300 in the data definition but the result in the strlen still remain 5, keep that in mind.

## Replace

REPLACE [{FIRST OCCURRENCE}|{ALL OCCURRENCES} **OF**] pattern

**IN** [section\_of] dobj **WITH** new

[IN {CHARACTER|BYTE} MODE]

[replace\_options].

Let consider the code below

**DATA**: v\_text **TYPE** string.

v\_text = 'AAAAFINDCBBBFINDB'.

**write** v\_text.

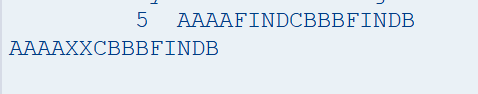
REPLACE 'FIND' **IN** v\_text **WITH** 'XX'.

**IF** sy-subrc = **0**.

**WRITE** / v\_text.

**ENDIF**.

The result will be:

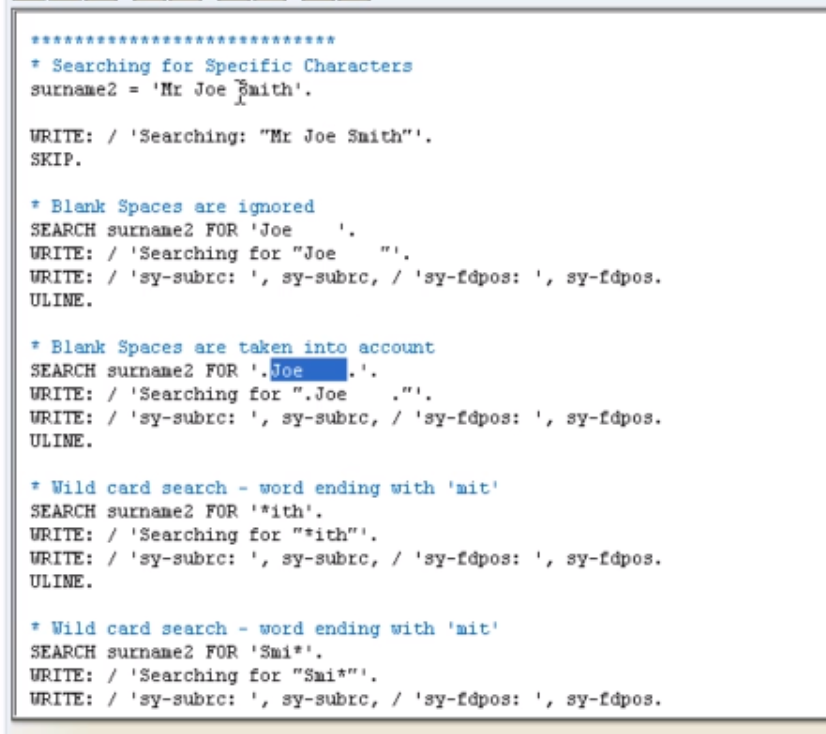


The **first** ‘FIND’ will be replaced with ‘XX’. To replace all occurrence in the text, we use

REPLACE ALL OCCURRENCES **OF** 'FIND' **IN** v\_text **WITH** 'XX'.

This time, all the ‘FIND’ will be replaced with ‘XX’

## Search



Consider the code below

**DATA** surename **TYPE** string **VALUE** 'Mr Joe Smith'.

**SEARCH** surename **FOR** 'Smith'.

**IF** sy-subrc = **0**.

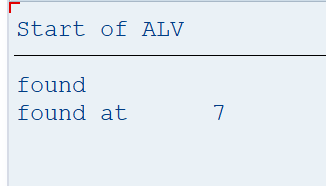
**WRITE** / 'found'.

**WRITE**: / 'found at ', sy-fdpos.

**ELSE**.

**WRITE** / 'not found'.

**ENDIF**.



* The pattern will be SEARCH obj FOR keyword.
* The start key of object will begin with the count of 0 to n

## Shift

This one is interesting. This statement shifts the content of a variable dobj. In [places](SAPEVENT:ABAPSHIFT_PLACES), you can specify the number of places to be shifted and in [direction](SAPEVENT:ABAPSHIFT_DIRECTION) the direction of the shift. In [deleting](SAPEVENT:ABAPSHIFT_DELETING), you can specify which characters to delete from the data object by the shift. If you use no addition at all, then the content is shifted to the left by one place.

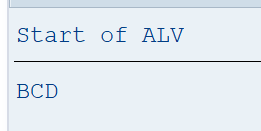
Let consider the first variant of Shift as default, just Shift keyword is defined

**DATA** alphabet(**10**) **VALUE** 'ABCD'.

**SHIFT** alphabet.

**WRITE** alphabet.

The content is shifted to the left by one place



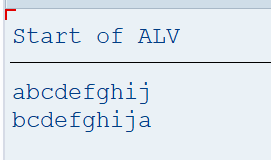
Second variant will be circular shifting

**DATA** alphabet(**10**) **VALUE** 'abcdefghij'.

**write** alphabet.

**SHIFT** alphabet CIRCULAR.

**WRITE** / alphabet.



## Split

**DATA**: string **TYPE** string **VALUE** '123 \*\* abc \*\* 456',

sep **TYPE** c **LENGTH** **3** **VALUE** '\*\*',

a1 **TYPE** string,

a2 **TYPE** string,

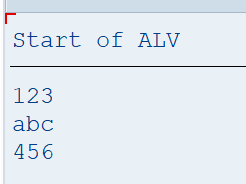
a3 **TYPE** string.

**SPLIT** string **AT** sep **INTO** a1 a2 a3.

**WRITE** / a1.

**WRITE** / a2.

**WRITE** / a3.



Split into internal table

**DATA** : LV\_STRING **TYPE** STRING **value** 'SPLIT ME AT SPACE'.

**TYPES**: **BEGIN OF** TY\_STRING,

STR(**25**) **TYPE** C,

**END OF** TY\_STRING.

**DATA** IT\_STRING **TYPE** **TABLE OF** TY\_STRING.

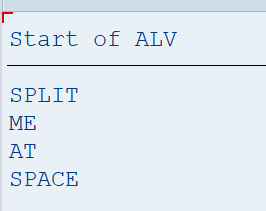
**DATA** WA\_STRING **TYPE** TY\_STRING .

**SPLIT** LV\_STRING **AT** ' ' **INTO** **TABLE** IT\_STRING .

**LOOP AT** IT\_STRING **INTO** WA\_STRING.

**WRITE** :/ WA\_STRING-STR.

**ENDLOOP**.



## References

* Internal table: <https://www.guru99.com/all-about-sap-internal-tables.html#:~:text=Work%20areas%20are%20single%20rows,one%20line%20at%20a%20time>.
* [SHIFT string in ABAP](https://www.testingbrain.com/sap/abap-tutorial/syntax/shift-abap-keyword.html)
* [SPLIT string in ABAP](https://solutionsapproblems.com/example-split-in-sap-abap/)