

ABAP DEVELOPMENT

SAP ABAP 程式設計基礎課程 第四堂課：常用函數

歡迎您成為ABAP程式開發者,在本課程中,我們將瞭解到:

數學及文字處理的常用函數

製作：周台誠
noah.chou@outlook.com

THE BEST-RUN BUSINESSES RUN SAP



函數名稱	說明
ABS	返回輸入參數的絕對值
SIGN	返回輸入參數符號：正數返回1；0返回0；負數返回-1
CEIL	返回不小於輸入參數的最小整數
FLOOR	返回不大於輸入參數的最大整數
TRUNC	返回輸入參數的整數部分
FRAC	返回輸入參數的小數部分
ACOS(X)	反餘弦函數，結果範圍為 $[-\pi/2, \pi/2]$ ，X的範圍為from [-1, 1] (PI為三角函數：3.14)
ASIN(X)	反正弦函數，結果範圍為 $[0, \pi]$ ，X的範圍為from [-1, 1]；
ATAN	反正切函數，結果範圍為 $[-\pi/2, \pi/2]$

函數名稱	說明
COS	餘弦函數
SIN	正弦函數
TAN	正切函數
COSH	餘弦曲線函數
SINH	正弦曲線函數
TANH	正切曲線函數
EXP	底數為e的幕函數（ $e = 2.7182818284590452$ ）
LOG	底數為e的自然對數
LOG10	底數為10的對數
SQRT	平方根
STRLEN	獲取字串長度函數

```
1 DATA: I1 TYPE I, I2 TYPE I, I3 TYPE I,  
2         F1 TYPE F, F2 TYPE F,  
3         WORD1(10), WORD2(20),  
4         XSTR TYPE XSTRING.  
5 F1 = ( I1 + EXP( F2 ) ) * I2 / SIN( 3 - I3 ).  
6 COMPUTE F1 = SQRT( SQRT( ( I1 + I2 ) * I3 ) + F2 ).  
7 I1 = STRLEN( WORD1 ) + STRLEN( WORD2 ).
```

- **CONCATENATE** 合併字串

語法: **CONCATENATE** f1 ... fn **INTO** g [**SEPARATED BY** h].

```
1 DATA: ONE (10)    VALUE 'John',
2          TWO (3)    VALUE 'F.',
3          THREE (10) VALUE 'Kennedy',
4          NAME (20) .
5 CONCATENATE ONE TWO THREE INTO NAME.
6 *NAME變數的內容為"JohnF. Kennedy"
7
8 DATA: ONE (10)    VALUE 'John',
9          TWO (3)    VALUE 'F.',
10         THREE (10) VALUE 'Kennedy',
11         NAME (20) .
12 CONCATENATE ONE TWO THREE INTO NAME SEPARATED BY SPACE.
13 *SEPARATED BY 表示在連接字符串中加入分隔符號
14 *因此NAME變數的內容為"John F. Kennedy"
15
16 DATA SEPARATOR (4) VALUE 'USA'.
17 CONCATENATE SPACE ONE TWO THREE INTO NAME
18           SEPARATED BY SEPARATOR.
19 *定義分隔符號為'USA'因為宣告時是4個字元，因此顯示時會多一個空白
20 *因此NAME變數的內容為"USA JohnUSA F.USA Ke"
```

- **SPLIT** 字串分割

語法: **SPLIT f AT g INTO h1 ... hn.** 將分割字串的值分配給變數。

SPLIT f AT g INTO TABLE itab. 將分割字串的值分配給內表。

```

1  DATA: NAMES(30)      TYPE C VALUE 'HELLO,SAP',
2      NAMES2           TYPE STRING,
3      ONE(10)          TYPE C,
4      TWO(10)          TYPE C,
5      DELIMITER(1)     VALUE ' , '.
6  TYPES: BEGIN OF ITAB_TYPE,
7      WORD(20),
8      END OF ITAB_TYPE.
9  DATA: ITAB TYPE STANDARD TABLE OF ITAB_TYPE WITH HEADER LINE.
10 SPLIT NAMES AT DELIMITER INTO ONE TWO.
11 WRITE: ONE,TWO.
12 *使用「,」為分隔符號並將字串寫入ONE , TWO兩個變數內
13 *因此:ONE=Hello,TWO=SAP
14 SPLIT 'ONE STOP Two STOP Three STOP' AT 'STOP' INTO TABLE ITAB.
15 LOOP AT ITAB.
16     WRITE: ITAB.
17 ENDLOOP.

```

- **SHIFT** 將字串整體指定字串進行轉移

語法: **SHIFT c <LEFT/RIGHT/CIRCULAR>**. 向左、向右或循環移動一個字符, 預設向左

SHIFT c BY n PLACES. 指定移動字符位數

如果**SHIFT**操作的對象是**C**類型, 那麼它的所有字節都會向前移動一位, 最後一位用空格代替。

倘若為**String**類型, 所有字符都會向前移動一位, 最後一位刪除。

```

1  DATA: STR1 (10)   TYPE C VALUE 'ABCDEFGH I J',
2      STR2          TYPE STRING,
3      STR3 (10)     TYPE C, STR4 (10) TYPE C,
4      STR5 (10)     TYPE C, STR6 (10) TYPE C.
5  STR6 = STR5 = STR4 = STR3 = STR2 = STR1.
6
7  SHIFT STR1.      "執行結果: 'BCDEFGH I J', 整體長度不變, 左移一位, 最後為空
8  SHIFT STR2.      "執行結果: 'BCDEFGH I J', 左移一位, 最後一位刪除
9  SHIFT STR3 RIGHT. "執行結果: ' ABCDEFGH I', 整體長度不變, 右移一位, 首位為空
10 SHIFT STR4 CIRCULAR. "執行結果: 'BCDEFGH I J A', 整體長度不變, 字符循環移動
11 SHIFT ALPHA1 BY 5 PLACES. "轉換結果: 'FGH I J      ', 整體向左移動5位
12 SHIFT ALPHA2 RIGHT BY 2 PLACES. "轉換結果: '  ABCDEFGH I J', 整體向右移動2位

```

● 語法: **SHIFT c UP TO c1.**

可指定某字串從某一個字串進行位移，並將執行結果返回給系統參數SY-SUBRC，若執行成功時，返回值為"0"，倘若字串與原字串不匹配，返回值為"4"。

```

1  DATA: STR1(10) VALUE 'ABCDEFGHIJ',
2         STR2(10) VALUE 'ABCDEFGHIJ',
3         THREE(3) VALUE 'DEF',
4         FOUR(4)  VALUE 'DEF '.
5  SHIFT STR1 UP TO THREE. "執行結果: 'DEFGHIJ', SY-SUBRC = 0.
6  SHIFT STR2 UP TO FOUR.  "執行結果: 'ABCDEFGHIJ', SY-SUBRC = 4, STR2未發生變化
7  SHIFT STR1 LEFT DELETING LEADING 'ABCD'. "移除字符串左邊的子字符串。
8  SHIFT STR2 RIGHT DELETING TRAILING 'GHIJ'. "移除字符串右邊的子字符串。
9  DATA: STR1(15) VALUE '      ABCDEFGHIJ',
10         STR2(15) VALUE '      ABCDEFGHIJ',
11         STR3(15) VALUE '      ABCDEFGHIJ',
12         STR4 TYPE STRING,
13         M1(4)      VALUE 'ABCD',
14         M2(6)      VALUE 'HIJ'.
15         STR3 = STR1.
16  SHIFT STR1 LEFT DELETING LEADING SPACE.
17  *執行結果: 'ABCDEFGHIJ      ', 移除前面空格，後面補零
18  SHIFT STR2 RIGHT DELETING TRAILING M1.
19  *執行結果: '      ABCDEFGHIJ', 子串不符合條件，未發生變化
20  SHIFT STR3 RIGHT DELETING TRAILING M2.
21  *執行結果: '      ABCDEFG', 移除字符串右邊'HIJ'
22  SHIFT STR4 LEFT DELETING LEADING SPACE.
23  *執行結果: 'ABCDEFG', 直接刪除前面空格

```

- **CONDENSE** 重新整合分配字串。

語法: **CONDENSE** c < NO-GAPS >.

```
1 DATA NAME (30) .  
2     NAME (10)      = '   Dr.' .  
3     NAME+10 (10)   = 'Michael' .  
4     NAME+20 (10)   = 'Hofmann' .  
5 WRITE NAME.      "輸出結果: '   Dr.   Michael   Hofmann'  
6  
7 CONDENSE NAME.  
8 WRITE / NAME.    "輸出結果: 'Dr. Michael Hofmann'  
9  
10 CONDENSE NAME NO-GAPS.  
11 WRITE / NAME.   "輸出結果: 'Dr.MichaelHofmann'
```

- 變數將會被重新賦值，並以空格分開，字串前置的空格被刪除。若在函數後加上關鍵字“NO-GAPS”，那麼輸出結果將刪除所以空格。

- 要特別注意的是該語法同樣適用於資料結構群組(**Structure**)中, 但資料結構群組中的不同字段內容將會按字段長度被重新分配。

```

1 DATA: BEGIN OF NAME,
2     TITLE (8)          VALUE '  Dr.',
3     FIRST_NAME (10)    VALUE 'Michael',
4     SURNAME (10)       VALUE 'Hofmann',
5 END   OF NAME.
6 CONDENSE NAME.
7 WRITE NAME.  "輸出結果: 'Dr. Michael Hofmann'
8 WRITE: / NAME-title, / NAME-FIRST_NAME, / NAME-SURNAME.
9
10 *NAME-title:      'Dr. Mich'    Title字段被重新按長度賦值
11 *NAME-FIRST_NAME: 'ael Hofman'  FIRST_NAME字段被重新按長度賦值
12 *NAME-SURNAME:    'n'          SURNAME字段被重新按長度賦值
    Dr. Michael Hofmann
    12345678          NAME-title
          1234567890  NAME-FIRST_NAME
          |           |
          |           | 1 NAME-SURNAME

```

- TRANSLATE 字串轉換

語法: **TRANSLATE c TO UPPER CASE.** 將字串轉為大寫

TRANSLATE c TO LOWER CASE. 將字串轉為小寫

```
1 DATA: str1(3) TYPE C VALUE 'abc',
2      | Str2(4) TYPE C VALUE 'EFGH'.
3 TRANSLATE str1 TO UPPER CASE. "執行結果: 'ABC', 轉換為大寫。
4 TRANSLATE Str2 TO LOWER CASE. "執行結果: 'efgh', 轉換為小寫。
```

TRANSLATE c USING c1.

將一字串參照另一字串轉換。

將按逐個字符判斷，若原字串中的某一字符在參考字串中存在，那麼取參考字串中該字符出現第一次的位置的下一個字符作為轉換對象。若參考字串中不存在該字符或該字符首次出現在參考字串最後一位，則不進行轉換

```
5 DATA: str(20) TYPE C VALUE 'abcabcabcXabcf',
6      | change(15) TYPE C VALUE 'aXbaYBabZacZB'.
7 TRANSLATE str USING change. "執行結果: 'XaZXaZXaZXaZf'
```

來源: abcabcabcXabcf

參考: aXbaYBabZacZB

結果: XaZXaZXaZXaZf

- 語法: **TRANSLATE** c ...**FROM CODE PAGE** g1 ...**TO CODE PAGE**g2.

SAP可以應用於多種軟硬的系統，不同系統間的訊息傳遞和通信必然涉及代碼轉換問題。

例如，**HP-UX**系統與**IBM EBCDIC**系統中的代碼轉換可通過下列語句實現，具體的碼制請查看數據表**TCP00**。

```
9  TRANSLATE c FROM CODE PAGE '1110' TO CODE PAGE '0100'.
```

- **OVERLAY** 參考字串對空白字符進行填充。

語法: **OVERLAY C1 WITH C2.**

該函數對原字串及參考字串按字符逐一比較, 若原字串某位置字符值為空而參考字串有值存在, 則用參考字串字符值進行取代。

```
1 DATA: WORK(20) VALUE 'Th t h s ch ng d.',  
2      |      HELP(20) VALUE 'Grab a pattern'.  
3 OVERLAY WORK WITH HELP.  
4 *執行結果: WORK = 'That has changed'.
```

- **REPLACE** 按條件取代字串。

語法: **REPLACE f ...WITH g ...INTO h.**

```
6 DATA STR(10) VALUE '12345ABCDE'.  
7 REPLACE '12345' WITH '56789' INTO field.  
8 *執行結果: STR = '6789ABCDE'.
```

SEARCH 搜索指定字串

- **SEARCH f FOR g.:** **g**表示被搜索字符串。
... **ABBREVIATED:** 從指定字串中按順序搜索相匹配字符串。
... **STARTING AT n1:** 從字串第**n1**個字符開始搜索。
... **ENDING AT n2:** 搜索到字串第**n2**個字符為止。
... **AND MARK:** 從指定字串中模糊搜索相匹配字符串。
- **SEARCH itab FOR g.:** **g**表示被搜索字符串。
... **ABBREVIATED:** 從內表中按順序逐行搜索相匹配字符串。
... **STARTING AT line1:** 從內表中具體某行開始搜索匹配字符串。
... **ENDING AT line2:** 搜索最大範圍到內表中具體某行。
... **AND MARK:** 從內表中模糊搜索相匹配字符串。
- 通過系統變量**SY-SUBRC**可以回執字串查找的結果，若回執等於**"0"**，則表示查找成功。某段字符在字串中的具體位置保存在系統變量**SY-FDPOS**中。

```
1 DATA STR(50).  
2 MOVE 'Welcome to SAP world!' TO STR.  
3  
4 SEARCH STR FOR 'Wrd' ABBREVIATED.  
5 *"wrd"按順序包含在子字串"world"中  
6 *執行結果: SY-SUBRC = 0, SY-FDPOS = 15  
7 SEARCH STR FOR 'SAP' STARTING AT 10.  
8 *從第10位開始尋找, SY-SUBRC = 0, SY-FDPOS = 2  
9 SEARCH STR FOR 'SAP' ENDING AT 10.  
10 *尋找第10位結束, 字符不存在, SY-SUBRC = 4, SY-FDPOS = 0  
11 SEARCH STR FOR '*AP' AND MARK.  
12 *尋找包含"AP"子字串, "SAP"符合條件, 預設為尋找"SAP"  
13 *SY-SUBRC = 0, SY-FDPOS = 11  
14 SEARCH STR FOR '*A' AND MARK.  
15 *匹配字串必須是子字串以空格分開的最後幾位, SY-SUBRC = 4
```

- 在使用**SEARCH**函數搜索字串時可用一些特殊符號，如"*"、"@","."等，在其兩邊加上頓號作為修飾。如某字符串為"**AAAA*BBBB**"，查找符號"*"的位置。

```
1 DATA STR(50).  
2 MOVE 'AAAA*BBBB' TO STR.  
3  
4 SEARCH STR FOR '.*.' ABBREVIATED.  
5 *從內表第二行中可查找到，SY-SUBRC = 0，SY-FDPOS = 5
```

- 從內表中查找字符串方法與上面基本類似，若內表有多行，那麼字符串查找位置默認為從某行數據的第一位開始。

```
7 data: BEGIN OF T_INF OCCURS 0,  
8     LINE(80),  
9 END OF T_INF.  
10 APPEND 'HELLO SAP' TO T_INF.  
11 APPEND 'I am come from China' TO T_INF.  
12 *定義內表後增加兩行數據  
13 SEARCH T_INF FOR 'SAP' ABBREVIATED.  
14 *從內表第一行中可查找到，SY-SUBRC = 0，SY-FDPOS = 6  
15 SEARCH T_INF FOR 'am' ABBREVIATED.  
16 *從內表第二行中可查找到，SY-SUBRC = 0，SY-FDPOS = 2
```