String operations

Ref Pseudocode

LENGTH, UCASE and LCASE

0478 publishes four string 字符串 routines and no others. Three of them are here.

`LENGTH(<identifier>)` returns how many characters 字符 a string holds — spaces included.

```
OUTPUT LENGTH("Happy Days")
OUTPUT LENGTH("")
```

`UCASE(<identifier>)` returns the string in upper case 大写, `LCASE(<identifier>)` in lower case 小写. Neither changes the original 原值 — each hands back a new value.

```
DECLARE Name : STRING
Name ← "Happy"
OUTPUT UCASE(Name)
OUTPUT LCASE(Name)
OUTPUT Name
```

That last line still prints `Happy`: to keep the change you must assign it back.

```
DECLARE Name : STRING
Name ← "Happy"
Name ← UCASE(Name)
OUTPUT Name
```

Comparing without worrying about case

`UCASE` on both sides is the standard way to accept an answer however it was typed:

```
DECLARE Answer : STRING
Answer ← "yes"
IF UCASE(Answer) = "YES"
  THEN
    OUTPUT "Accepted"
ENDIF
```

Common mistakes

- Expecting `UCASE(Name)` to change `Name`. Assign the result back.

- `LENGTH` counts characters, not words. `LENGTH("Happy Days")` is 10.

SUBSTRING

`SUBSTRING(<identifier>, <start>, <length>)` pulls a piece out of a string — starting at position `start`, taking `length` characters.

```
OUTPUT SUBSTRING("Happy Days", 1, 5)
OUTPUT SUBSTRING("Happy Days", 7, 4)
```

△ **Counting starts at 1**, not 0. The first character of `"Happy Days"` is at position 1.

Taking one character, and the rest

```
DECLARE Word : STRING
Word ← "Computer"
OUTPUT "First letter: ", SUBSTRING(Word, 1, 1)
OUTPUT "Last letter: ", SUBSTRING(Word, LENGTH(Word), 1)
OUTPUT "Without the first: ", SUBSTRING(Word, 2, LENGTH(Word) - 1)
```

`LENGTH` inside `SUBSTRING` is what makes the last two lines work for a word of any size — hard-coding `8` would break the moment the word changed.

Reading a string one character at a time

```
DECLARE Word : STRING
DECLARE Index : INTEGER
Word ← "Cat"
FOR Index ← 1 TO LENGTH(Word)
    OUTPUT SUBSTRING(Word, Index, 1)
NEXT Index
```

Common mistakes

- Starting at 0. Position 1 is the first character.
- Reading the third argument as an *end position*. It is a **count** — `SUBSTRING(W, 2, 3)` takes three characters beginning at the second.
- Asking for more characters than the string has.