字符串操作

Ref Pseudocode

LENGTH, UCASE and LCASE

0478 发布四个字符串例程，除此之外没有其他。其中三个在这里。

`LENGTH(<identifier>)` 返回字符串包含的字符数量 — 包括空格。

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

`UCASE(<identifier>)` 将字符串转换为大写，`LCASE(<identifier>)` 转换为小写。两者都不会修改原始值 — 每个函数都会返回一个新值。

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

最后一行仍然输出 `Happy`：若要保留更改，必须将其赋值回原变量。

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

■ 忽略大小写进行比较

在两侧使用 `UCASE` 是接受任何输入方式答案的标准方法：

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

常见错误

- 期望 `UCASE(Name)` 改变 `Name`。需将结果重新赋值。
- `LENGTH` 计算的是字符数而非单词数。`LENGTH("Happy Days")` 是 10。

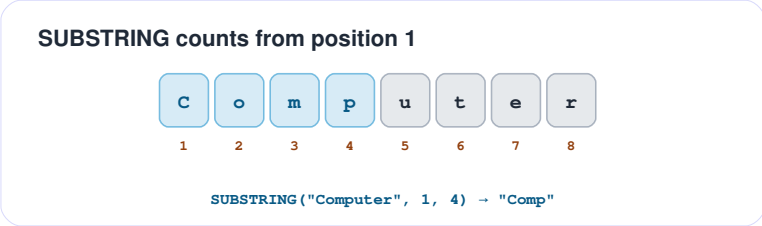
exercises.lol exercises.com.cn

SUBSTRING

`SUBSTRING(<identifier>, <start>, <length>)` 从字符串中提取一段 — 起始于位置 `start`，取 `length` 个字符。

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

△ **计数从 1 开始**，而非 0。`"Happy Days"` 的第一个字符位于位置 1。



字符从 *1* 开始计数：*SUBSTRING(“Computer”, 1, 4)* 结果为 *“Comp”*

■ 提取单个字符及剩余部分

```
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` 嵌套在 `SUBSTRING` 中，使得最后两行适用于任意长度的单词 — 硬编码 `8` 会在单词改变时立即失效。

■ 逐字符读取字符串

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

常见错误

- 从 0 开始。位置 1 是第一个字符。
- 将第三个参数读作 结束位置。它是一个 **计数** — `SUBSTRING(W, 2, 3)` 从第二个字符起取三个字符。
- 请求的字符数超过字符串实际长度。