

Programming Fundamentals

IGCSE Computer Science • Topic 8

IGCSE Computer Science

```
{ ____  
  ____  
  ____ }
```

Programming Fundamentals

What we will cover

- 1 Variables, constants, data types
- 2 The three structures
- 3 Operators & strings
- 4 Procedures & functions
- 5 Arrays
- 6 File handling

age

16

a variable is a named box
that holds a value

1 Data

Programming Fundamentals

Data

Variables, constants, data types

- **variable** 变量: a named store whose value can **change**
- **constant** 常量: a named value that is **fixed**

Five **data types** 数据类型: INTEGER, REAL, CHAR, STRING, BOOLEAN.

Pick the **smallest precise** type: a whole count is INTEGER, a price is REAL, a yes/no is BOOLEAN.

integer	whole number: 7
real	decimal: 3.5
char	one letter: 'A'
string	text: "hello"
boolean	true or false

Programming Fundamentals

Data

The three basic structures



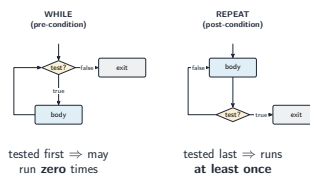
A **data type** 数据类型 says what kind of value a variable holds.

Every program is built from three control structures.

Programming Fundamentals

Data

Loops and counting



- **totalling** 求和 – keep adding values to a running total: 'total ← total + value'.
- **counting** 计数 – add 1 to a counter each time: 'count ← count + 1'.

2 Operators and strings

Operators

arithmetic
+ - × /
DIV (whole),
MOD (remainder)

comparison
= ≠ < > ≤ ≥

logical
AND, OR, NOT

17 DIV 5 = 3 (whole part), 17 MOD 5 = 2 (remainder) – MOD is how you test “is it even?”

String handling

- **LENGTH(s)** – number of characters
- **SUBSTRING(s, start, n)** – part of a string
- **UCASE / LCASE** – change case
- **& – concatenation** 拼接 (join)

String positions count from 1, not 0. And a **nested** 嵌套 statement is just one structure *inside* another – an IF within a FOR.

3 Subroutines

Procedures and functions

- **procedure** 过程: a named block that *does an action*, returns nothing
- **function** 函数: a named block that *returns a value*

Library routines 库例程 are ready-made subroutines (ROUND, RANDOM).

A **function** call sits inside an expression (it gives a value back); a **procedure** call is a statement on its own.

Writing a maintainable program

meaningful names
score, not x

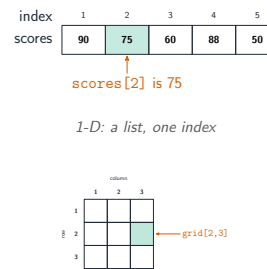
comments
explain *why*,
not the obvious

subroutines
split the work into
named blocks

Maintainable code is code the *next person* (or you, next month) can read and change without fear.

4 Data structures

Arrays



An **array** 数组 holds many values of the same type under one name, picked out by an **index** 索引.

A 1-D array is a list (Marks[3]);
a 2-D array is a grid (Grid[2,3])
– perfect for a FOR loop over every element.

File handling

Store data on disk so it survives when the program ends:

- OPENFILE for READ / WRITE
- READFILE / WRITEFILE
- CLOSEFILE when done

Variables live in RAM and vanish at the end; a **file** keeps the data between runs. Always CLOSEFILE – unsaved writes are lost otherwise.

5 Recap

Topic 8 – key takeaways

1 Structures

sequence, selection, iteration

2 Loops

FOR known; WHILE before; REPEAT after

3 Subroutines

procedure acts; function returns a value

4 Arrays & files

array = same type by index; file survives runs

Check yourself

- 1 Which loop may run zero times?
- 2 What is 17 MOD 5?
- 3 Procedure or function: which returns a value?
- 4 From which index do string positions count?

{ ____
____ }

Answers 1. WHILE 2. 2 3. a function 4. 1