R 11 Assignment, Variables, Constants & Sequences - Reading

Computer programs are normally written in **high level languages** that are close to how humans think rather than computers.

In computer programs we often want to store **values**. For instance, we may want to store a player's name or score in a game. The values that we store might need to change in the program so we store them in **variables** (as the values can *vary*).

A variable is an identifier (name) that points to a memory location in RAM which stores a value that can change when the program is run.

The rules as to how we write computer code are known as **syntax**. Here we will use syntax that is not for a specific language but is easy to understand no matter what language you decide to actually program in.

Putting a value into a variable is known as **assignment**. If we do this when the variable is first set up, it is known as **initialisation**.

Syntax for assignment

```
variableName ← value
```

Example of assignment

```
score ← 17
```

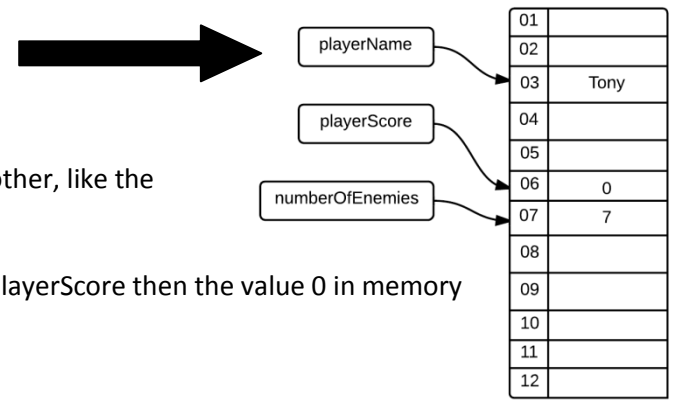
The ← symbol is the **assignment operator** in this situation. For the above example we say that "the variable score is **assigned** the value 17". Some language will use an = symbol for assignment.

In general, variables are written with no spaces and in lowercase. They can be written with an underscore separating words, which is known as **snake case**. Alternatively, words can be joined with each word starting with a capital letter, and this is known as **camel case**.

Example snake case variable names	Example camel case names
player_name	playerName
player_score	playerScore
number_of_enemies	numberOfEnemies

The following code will set up three variables. The variable names, pointers, memory locations and values in RAM are shown on the right as they would be at the end of the three lines of code running.

```
playerName ← 'Tony'
playerScore ← 0
numberOfEnemies ← 7
```



Lines of code which execute one after the other, like the above program, are known as a **sequence**.

If the following assignment were made to playerScore then the value 0 in memory would become replaced by 10.

```
playerScore ← 10
```

If we want to store a value that doesn't change while the program is running then we store it in a **constant**. Constants are normally written with capital letters, e.g. MAX_NUMBER_OF_PLAYERS

```
constant VAT_RATE ← 20
```