

# 6502 Program Example 01

Most assembly language programs have the same general layout. You can divide the page up into columns, and it is a good idea to try and stick to that format:

Label	Mnemonic Code or Assembler Directive	Operand	Comment
-------	--	---------	---------

Here is a simple program:

```
; Program to add two 8 bit numbers
; The numbers being added are 4 and 6
; Add0801.65s

        .ORG $0200    ; Store machine code starting here

        LDA #$04      ; Store first number (4) in
        STA no1        ;   byte labelled no1
        LDA #$06      ; Store second number (6) in
        STA no2        ;   byte labelled no2

        CLC           ; Clear the carry flag
        LDA no1        ; Load first number into accumulator register
        ADC no2        ; Add with carry second number
        STA res        ; Store the result in byte labelled res

        BRK           ; Stop running the program

no1:     .DB $00        ; The .DB directive instructs the assembler to
                        ; reserve one byte (8 bits)
no2:     .DB $00        ; of memory, and allows the programmer to refer
                        ; to it by using a label.
res:     .DB $00        ; This is how you declare variables.
```

This program, when assembled by the assembler, will produce the following machine code:

Address   Contents

0200	A9 04 8D 16 02 A9 06 8D
0208	17 02 18 AD 16 02 6D 17
0210	02 8D 18 02 00 00 00 00

## Note:

- Use of semicolon to indicate comments. These are ignored by the assembler.
- Use of assembler directives (.ORG and .DB). Directives for this assembler always start with a full stop. These are not assembly language commands which need to be translated into binary machine code instructions. These are instructions to the assembler which tell it how to go about the translation process itself.
- Use of labels to refer to memory locations (no1, no2, res) which enable me to use variables, and not have to worry about precisely where in memory these values are being stored.