„„Location Counter

Segment data
a db 17, -2, 0ffh, 'xyz',…
db ….
db….
;lga db $-a (mov [lga],….) ok – lga is a variable
;lga EQU $-a (mov [lga],… - illegal !!!) – lga is a CONSTANT
;lga dw $-$$; ok ! – IF a is THE FIRST identifier to be defined in the data segment !!!!
;lga dw lga-a ; !!!!!!!

b EQU 27
c dd 12345678h
lg dw b-a ; NO !!! b is NOT an address !!! syntax error
lg db c-a ; OK !!!
lga dw $-a-4 ; ok !!!
lg dw $-a ; length (a) + 4 !!!

If no SECTION directive is explicitly used, The $$ symbol will evaluate implicitly to the offset of the beginning of the current segment.

":" are mandatory to be present when DEFINING CODE labels (ex: "start:") but they must be absent when defining DATA labels (ex: defining variables "a db 17")

The format of a source line isn't specific only to the code segment, but is general applicable for ant source line independently of the type of that segment (inclusively a data segment)

[***label[:]***] [***prefixes***] [***mnemonic***] [***operands***] [***;comment***]

The offset of any label is a constant value determinable at assembly time. In any programming language the location of an allocated variable (its address) remain fixed; that is why the offsets of variables represents constant values determinable at assembly/compile time.

The SEGMENT address is also fixed but determinable ONLY at loading time.

Any offset used only by itself in a program (without the segment part) will be finally completed BY THE ASSEMBLER to a FAR address by prefixing it with a corresponding segment value. This IMPLICIT value will be always one of the CS, DS or SS segment registers and the rules for these implicit associations are:

- **CS** for code labels target of the control transfer instructions (jmp, call, ret, jz etc);
- **SS** in SIB addressing when using EBP or ESP as *base* (no matter of *index* or *scale*);
- **DS** for the rest of data accesses;