Allocating Storage Space for Initialized Data

The syntax for storage allocation statement for initialized data is −

```
[variable-name]    define-directive    initial-value   [,initial-value]...
```

Where, *variable-name* is the identifier for each storage space. The assembler associates an offset value for each variable name defined in the data segment.

There are five basic forms of the define directive −

Directive	Purpose	Storage Space
DB	Define Byte	allocates 1 byte
DW	Define Word	allocates 2 bytes
DD	Define Doubleword	allocates 4 bytes
DQ	Define Quadword	allocates 8 bytes
DT	Define Ten Bytes	allocates 10 bytes

Following are some examples of using define directives −

```
choice          DB      'y'

number          DW      12345

neg_number      DW      -12345

big_number      DQ      123456789

real_number1    DD      1.234

real_number2    DQ      123.456
```

Please note that −

- Each byte of character is stored as its ASCII value in hexadecimal.

- Each decimal value is automatically converted to its 16-bit binary equivalent and stored as a hexadecimal number.
- Processor uses the little-endian byte ordering.
- Negative numbers are converted to its 2's complement representation.
- Short and long floating-point numbers are represented using 32 or 64 bits, respectively.