

# GAS Tutorial - 3. Syntax

Young W. Lim

2016-03-01 Tue

## 1 Syntax

## “Using as”, Dean Elsner, Jay Fenlason & friends

I, the copyright holder of this work, hereby publish it under the following licenses: GNU head Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled GNU Free Documentation License.

CC BY SA This file is licensed under the Creative Commons Attribution ShareAlike 3.0 Unported License. In short: you are free to share and make derivative works of the file under the conditions that you appropriately attribute it, and that you distribute it only under a license compatible with this one.

- 1+ characters chosen from the set of
  - all letters (both upper and lower case)
  - digits
- '\$' On most machines, you can also use \$

in symbol names; exceptions are noted in Chapter 9 [Machine Dependencies], page 79.

- may not begin with a digit
- case is significant
- no length limit
- delimited
  - by characters not in that set
  - by the beginning of a file

- A statement ends at a newline character (") or line separator character(;).
- A statement begins with zero or more labels
- A statement optionally followed by a key symbol which determines what kind of statement it is.
  - the symbol begins with a dot '.' : an assembler directive
  - the symbol begins with a letter : an assembly language instruction:
- A label is a symbol immediately followed by a colon (:)

# Number Constants

- Integers : int in the C language
  - binary : '0b' or '0B'
  - octal : '0'
  - decimal : starts with a non-zero digit
  - hexadecimal : '0x' or '0X'
- Bignums : int more than 32 bits
  - the same syntax and semantics as int
- Flonums : floating point numbers
  - The digit '0'
  - A letter e
  - ('+-')(int part).(frac part)(exp part)

## Configuring listing output : '-listing'

This feature combines the input source file(s) with a hex dump of the corresponding locations in the output object file

The format of this listing can be controlled by directives inside the assembler source

- .list
- .title
- .sbttl
- .psize
- .eject

## Join Text and Data Sections : '-R'

- treat all data-section data in the text section.
- the data section part of your object file is zero bytes long
- generate shorter address displacements



