

# Asm80i アセンブラー

## 実行方法

コマンドプロンプトで実行します。

```
asm80i オプション ソースファイル名
```

## オプション

-v2	バージョン 2 の機能を有効にする <ul style="list-style-type: none"><li>SCOPE~ENDSCOPE ディレクティブ</li></ul>
-----	--

## コメント

セミコロンから行末まではコメントとなります。

## 定数

10 進定数	1 つ以上の 0~9
16 進定数	0~9 で始まり 0~9 または A~F と末尾の H または\$で始まり 0~9 または A~F
文字定数	シングルのまたはダブルクォーテーションで囲まれた文字

## シンボル

シンボルに使える文字はアルファベットと\_ . ? @ です。2 文字目以後は数字も使えます。

## 演算子

以下の演算子を使用できます。

### 単項

+	正符号
-	負符号
HIGH	上位バイト
LOW	下位バイト
NOT	反転(1 の補数)

### 二項

+	加算
-	減算
*	乗算
/	除算
MOD	剰余
SHL	左シフト
SHR	右シフト
AND	論理積
OR	論理和
XOR	排他的論理和

優先順位は以下の通りです。

優先順位	演算子
1	<b>*, /, MOD</b>
2	<b>+, -</b>
3	<b>SHL, SHR</b>
4	<b>AND</b>
5	<b>OR, XOR</b>

## 疑似命令

<b>INCLUDE</b>	他のファイルを挿入します。
<b>PUBLIC</b>	シンボルを他のモジュールから参照可能にします。
<b>EXTERN</b> <b>EXT</b>	他のモジュールのシンボルを参照します。
<b>CSEG</b>	この命令以後をコードセグメントに配置します。
<b>DSEG</b>	この命令以後をデータセグメントに配置します。
<b>EQU</b>	定数を定義します。
<b>DEFB</b> <b>DB</b>	バイト定数配置します。
<b>DEFW</b> <b>DW</b>	ワード定数配置します。
<b>DEFS</b> <b>DS</b>	メモリ領域を確保します。

## -v2 オプション指定時

<b>SCOPE~ENDSCOPE</b>	ラベルの有効範囲を指定します。ただし <b>PUBLIC</b> 宣言されたラベルを除きます。
-----------------------	---

## 独自機能

### 複数命令

| で区切ることで、1 行に複数の命令を記述できます。

## 構造化命令

### IF-ELSE-ENDIF

記述	機能
<b>IF</b> 条件 ブロック 1	条件が真の場合はブロック 1、偽の場合はブロック 2 を実行します。 <b>ELSE</b> ブロックは省略可能です。
<b>ELSE</b> ブロック 2	

ENDIF
-------

DO-WHILE-WEND

記述	機能
DO	ブロック 1 を実行し、条件が成立する限りブロック 2 を繰り返します。
ブロック 1	
WHILE 条件	
ブロック 2	
WEND	

LinkLE リンカー

実行方法

コマンドプロンプトで実行します。

linkle 出力ファイル名 コードセグメントアドレス データセグメントアドレス オブジェクトファイル名...

セグメントアドレスには、4000-4fff,2000 のように複数範囲を指定できます。

拡張子の指定により、出力ファイルの形式は以下のようになります。

拡張子	形式
CMT	PC-8001 エミュレーター用
P6	PC-6001 エミュレーター用(ローダー付き)
MZT	MZ エミュレーター用
CAS	MSX エミュレーター用
RAM	PASOPIA エミュレーター用 RAMPAK2
PRG	JR-100 エミュレーター用
CJR	JR-200 エミュレーター用
L3	ベーシックマスターレベル 3 エミュレーター用
CAS	MSX エミュレーター用
T64	Commodore64,VIC-20 エミュレーター用
HEX	インテル HEX フォーマット
S	モトローラ S レコードフォーマット
上記以外	コードセグメントの内容そのまま(アドレス情報なし)