

# Documentação Linguagem ASTEKA

Carine Viecili, Tiago Debastiani

19 de outubro de 2014

## Conceitos Básicos

Asteka é uma linguagem procedural, ou seja, é semelhante a C. Todas as linhas, com exceção das iniciadas por **se**, **senao** ou **laco**, devem terminar com dois pontos (:).

Os elementos da linguagem Asteka são diferentes dos comumente usados em outras linguagens. Veja as diferenças:

- Atribuição = !

Os operadores matemáticos:

- Soma = \*
- Subtração = /
- Divisão = -
- Multiplicação = +
- Mod = %

Operadores lógicos:

- Igual = \$
- Diferente = §
- *And* = #
- *Or* = @
- Maior = >
- Menor = <

## Declarando Variáveis

A declaração das variáveis na linguagem Asteka é simples e pode ser feito de duas formas:

Declarando a variável e já atribuindo a ela um valor, sendo ele numérico ou não:

```
var nome_da_variavel ! valor:
```

Ou apenas declarando a variável, que será atribuído o valor `null`:

```
var nome_da_variavel:
```

## Leitura e Impressão

Para ler um valor qualquer do teclado, usa-se o comando `leia`, seguido pela variável entre aspas simples(' '). Atribui apenas uma variável de cada vez. Exemplo:

```
leia 'a':
```

A impressão na tela é feita pelo comando `imprime` seguido pela variável ou, pela frase entre aspas simples. A impressão em nova linha é feita inserindo o caractere “`_`” em qualquer lugar da frase, ou após a variável. Veja os exemplos:

Comandos	Tela
<code>var nome ! Tiago:</code>	<code>nome = Tiago</code>
<code>imprime 'nome = ':</code>	
<code>imprime nome_:</code>	<code>fi</code>
<code>imprime 'fi_m':</code>	<code>m</code>

## Controle de Fluxo

O controle de fluxo na linguagem Asteka é feito pelos comandos `se` e `senao`. A condição deve ficar entre colchetes ([ ]).

O fim dos comandos são definidos por `endse` ou `endsenao`, respectivamente ou, outro comando `senao`. Veja os exemplos:

```
se [a % 2 $ 0]
    imprime 'É divisível por 2_':
senao [a % 3 $ 0]
    imprime 'É divisível por 3_':
senao
    imprime 'Não é divisível por 2 nem por 3_':
endsenao:

se [a $ 2]
    imprime a_:
endse:
```

## Laço

O laço é feito pelo comando `laço` seguido pela condição entre colchetes (`[` `]`). O fim do laço é definido pelo comando `endlaco`. Veja exemplo:

```
var i ! 0:
laço [i < 10]
    imprime i_:
    i ! i * 1:
endlaco: o
```