

DOCUMENTAÇÃO DA LINGUAGEM COMPLICADORES

Sintaxe Básica:

***Observação:** para todos os valores esperados, como `<#start`, `var`, `#>`, `safe` e `show`, poderão ser digitados tanto na forma maiúscula quanto na forma minúscula que o compilador irá aceitar, pois foi feito o processo de transformação caso entre com valores maiúsculos. Outro detalhe é em cada linha para o compilador saber diferenciar os tokens é necessário espaço entre eles.

Início: para iniciar o código, é necessária a presença do símbolo “`<#`” seguido da palavra ***start***, como mostra o exemplo a baixo:

`<#start`

Erro palavra-chave Início: caso o usuário digite o termo de forma errada ou esqueça de digitar, o compilador irá expor o erro causado, como mostrado a seguir:

Codigo Fonte	Erros
<pre>var <u>inta</u>, <u>intb</u>, <u>intresultado</u>; a = 6; b = 7; safe(a); safe(b); resultado = a + b; safe(resultado); show(resultado); #></pre>	<pre><#start não encontrado.</pre>

Declaração de Variáveis: a declaração de variáveis é feita utilizando a notação “**var**”, seguido de um dos dois tipos numéricos presentes, `int` e/ou `float`, como mostrado no exemplo a baixo:

Ex: `var float: a, int: b;`

Também pode ser adiciona valor no cabeçalho de variaveis:

Ex: `var float: a = 7.0, int: b =7;`

Erro declaração de variáveis: se o usuário não entrar com a notação var, o compilador irá alertá-lo sobre a falta do mesmo, e caso o usuário digite *varfloat: a=3.5;* (sem a presença do espaço entre o var e o tipo da variável) também irá dar erro. O erro que irá aparecer é o seguinte:

Código Fonte	Erros
<pre><#start inta, int:b, int:resultado; a = 6; b = 7; safe(a); safe(b); resultado = a + b; safe(resultado); show(resultado); #></pre>	<p>Palavra chave var ausente ou não encontrada.</p>

Operações de adição e subtração: No caso da operação de adição o símbolo utilizado é “+” e para subtração é utilizado “-“. No exemplo abaixo é possível observar a demonstração de um cálculo efetuado no sistema:

	Resultado
<pre><#start var inta, intb, int:resultado, floatx, floaty, floatres2; a = 6; b = 7; safe(a); safe(b); resultado = a + b; safe(resultado); show(resultado); x = 1.5; y = 9.2; safe(x); safe(y); res2 = y - x; safe(res2); show(res2); #></pre>	<p>O valor da variável resultado é: 13 ,O valor da variável res2 é: 7.7</p>

Erro ao remover operador: se o usuário não entrar com um operador, o compilador irá alertá-lo sobre a falta do mesmo. O erro que irá aparecer é o seguinte:

Código Fonte	Erros
<pre><#start var int:a, int:b, int:resultado, float:x, float:y, float:res2; a = 6; b = 7; safe(a); safe(b); resultado = a b; safe(resultado); show(resultado);</pre>	<p>A linha resultado = a b contém erro.</p>

Comando de Escrita: O comando responsável pela escrita é o comando “show”, que deve ser escrito na forma minúscula, como mostra o exemplo:

show(resultado que será mostrado na tela);

Comando de Leitura: O comando responsável pela leitura dos dados é o comando “safe”, que também deve ser escrito na forma minúscula, conforme demonstrado no exemplo:

safe(p1);

Símbolo de encerrar comando: para que o compilador entenda que um comando foi finalizado, ele espera encontrar o símbolo “;” como mostra o exemplo a seguir:

show(resultado);

Erro no símbolo de encerramento: caso o compilador não encontre o símbolo “;” no final de qualquer comando, o seguinte erro irá aparecer:

Codigo Fonte	Erros
<pre> <#start var int:a, int:b, int:resultado; a = 6; b = 7; safe(a); safe(b) resultado = a + b; safe(resultado); show(resultado); #> </pre>	<div>Erro encontrado o ; na linha safe(b)</div>

Fim de programa: para ser feita a finalização do código, o compilador espera a presença do símbolo, como no exemplo abaixo:

#>

Erro de fim de programa: caso o usuário digite o termo de forma errada ou esqueça de digitar, o compilador irá expor o erro causado, como mostrado abaixo:

Codigo Fonte	Erros
<pre> <#start var int:a, int:b, int:resultado; a = 6; b = 7; safe(a); safe(b); resultado = a + b; safe(resultado); show(resultado); </pre>	<div>#> não encontrado.</div>