

Universidade Federal da Fronteira Sul

Programação I – Trabalho I

Anderson Barberini anderson.aeb@gmail.com

Manual da Linguagem Hak

Declaração de Variáveis:

Sintaxe:

`$<nome_variavel> = <valor da variável>;`

As variáveis não possuem tipagem, e podem receber quatro tipos primitivos de valores:

- **String:** `$variavel = "valor da variável";`
- **Int:** `$variavel = 10;`
- **Double:** `$variavel = 10.5;`
- **Boolean:** `$variavel = true; / $variavel = false;`

Além disso, elas podem assumir valores lógicos e aritméticos, conforme a tabela abaixo:

Operações Matemáticas	
$a + b$	Soma de a com b
$a - b$	Subtração de a com b
$a * b$	Multiplicação de a por b
a / b	Divisão de a por b
$a \% b$	Resto da divisão de a por b

Operações Lógicas	
$a > b$	a maior que b
$a < b$	a menor que b
$a \geq b$	a maior ou igual a b
$a \leq b$	a menor ou igual a b
$a \neq b$	a diferente de b
$a == b$	a igual a b

Só são aceitos dois operandos, tanto nas operações matemáticas quanto nas lógicas. Vale para declaração de variáveis, controlador de fluxo e laços.

Controlador de Fluxo

Sintaxe:

```
if(<condição>) {  
    <comandos>  
} else {  
    <comandos>  
}
```

As condições e propriedades usadas no controlador de fluxo são de acordo com a tabela de operações lógicas, mostradas no item “Declaração de Variáveis”.

Laço

Sintaxe:

```
while(<condição>) {  
    <comandos>  
}
```

As condições e propriedades usadas no laço são de acordo com a tabela de operações lógicas, mostradas no item “Declaração de Variáveis”.

Exemplo:

```
$i = 0;  
while($i < 3) {  
    $i = $i + 1;  
}
```

Comandos de saída

Sintaxe:

```
print <valor> // Imprime um valor na tela  
println <valor> // Imprime um valor na tela e depois pula uma linha
```

O valor de um comando de saída pode ser uma variável, uma operação matemática, uma operação lógica ou uma string.

Ex:

```
println $a + $b;  
println “Imprimindo uma string”;
```

```
print $a < $b;
```

Comandos de entrada

Sintaxe:

```
scan <variavel>
```

Como a linguagem não possui tipagem, não é necessário declarar a variável antes, sendo assim, segue o exemplo abaixo:

```
scan $nome;
```

O comando irá fazer a leitura de entrada e armazenará o valor na variável \$nome.

Comentários

Sintaxe:

```
// <comando>
```

Ao adicionar duas barras na frente de uma linha qualquer o programa entende com um comentário, e ignora o comando;

Observações sobre o interpretador

- O programa é case sensitive, todos os comandos são em letras minúsculas;
- O programa aceita aninhamento de comandos;
- Possui sintaxe flexível, permitindo espaços entre comandos;
- Só reconhece comandos linha por linha, não podendo separá-los;