

Documentação

A linguagem definida permite 11 comandos básicos, sendo eles: comentários, identificadores, atribuição, repetição enquanto, leitura, escrita, condicional se, condicional ternário, repetição faça enquanto, funções, condicional switch e um comando inovador. Todos os comandos precisam terminar com ; (ponto e vírgula) ou } (fecha chaves).

Comentários: podem ser escritos em uma linha ou em um bloco.

Comentários de uma linha: começam com /c/ e terminam com um enter.

Exemplo: /c/ esse é um comentário válido

Comentários em bloco: iniciam com /c* e finalizam com */.

Exemplo: /c* esse é um comentário

válido em

bloco */

Identificadores: um identificador deve começar com seu tipo, que pode ser number, string ou boolean, e em seguida o nome da variável, que deve iniciar obrigatoriamente por uma letra. Podem ser criados um ou vários identificadores na mesma linha, assim como inicializar a variável.

Exemplos: string palavra;

string palavra2, palavra3;

number valor = 2;

Atribuição: a atribuição deve começar com o nome da variável, que deve iniciar obrigatoriamente por uma letra. Após isso, é inserido o símbolo =, seguido por uma variável, string, número ou um cálculo aritmético.

Exemplos: palavra = "palavra";

number = 2 + 1;

palavra2 = palavra;

Condição: a condição é utilizada por alguns comandos, sendo iniciada por uma variável ou número, seguida de um operador relacional e outra variável ou número.

Exemplo: valor > 5

Repetição enquanto: a repetição inicia com o while, seguido por uma condição, depois é aberto uma chave ({), que conterá um comando dentro, por fim a chave é fechada (}).

Exemplos: while valor > 5 {

 write "maior que 5";

}

Leitura: a leitura serve para ler um valor do usuário e armazenar em uma variável, ou seja, é necessário informar o comando read seguido por uma variável.

Exemplo: read teste;

Escrita: a escrita serve para mostrar um valor ao usuário. O comando inicia com write, seguido por um número, string ou valor booleano.

Exemplo: write "olá mundo";

Condicional se: esse condicional inicia por if, seguido de uma condição, abertura de chaves, comandos, fechamento de chaves. O comando pode ter ou não a parte do else, que inicia por else, abertura de chaves, comandos, fechamento de chaves.

Exemplo: if valor > 0 {
 write "positivo";
}

```
if valor > 0 {  
    write "positivo";  
} else {  
    write "negativo";  
}
```

Condicional ternário: esse condicional inicia por ternary, seguido de uma condição, dois pontos (:), um valor (dígito, string ou booleano), ponto de exclamação (!) e outro valor.

Exemplo: ternary 2 > 0 : true ! false;

Repetição faça enquanto: a repetição inicia com o do, seguido por abertura de chaves, comandos, fechamento de chaves, depois é informado o while, seguido de uma condição.

Exemplos: do {
 write "olá mundo";
} while 2 > 0;

Funções: uma função é declarada iniciando por function, seguida por uma variável, abertura e fechamento de parênteses. Dentro dos parênteses podem ou não ser informados os parâmetros, que são variáveis. Após isso, há uma abertura de chaves, comandos, fechamento de chaves.

Exemplos: function exemplo1(){
 write "exemplo 1";
}

```
function funcao2(valor){  
    if valor > 0 {  
        write "valor positivo";  
    } else {  
        write "valor negativo";  
    }  
}
```

Condicional switch: esse condicional inicia pela palavra switch, seguida de uma variável. Há uma abertura de chaves para informar os cases, que podem haver quantos forem necessários. Os cases são compostos por case, valor, dois pontos (:), comandos e um break opcional.

No final é necessário haver um nocase:, seguido por comandos e um break opcional. Há o fechamento de chaves no fim.

Exemplo: switch variavel {

case 1:

read teste;

write "caso1";

break;

case 2:

write "caso2";

break;

nocase:

write "nenhum";

break;

}

Comando inovador: esse comando executa uma ação enquanto uma condição for verdadeira, o diferencial é que faz isso a cada tempo determinado. O comando inicia com o innovative, seguido por uma condição, depois é aberto uma chave ({), que conterà um comando dentro, por fim a chave é fechada (}). No final é inserido um número em segundos, que será o tempo que cada execução será feita.

Exemplos: innovative valor > 5 {

write "execucao a cada 2 segundos";

} 2

Aprendizados e dificuldades

Esse foi um trabalho com uma certa dificuldade, pois envolveu muitos conteúdos, funções, tokens, etc, e por isso foi um trabalho demorado para fazer e gerou algumas dúvidas.

Com o auxílio dos slides e das aulas disponibilizadas pelo professor, essas dúvidas foram respondidas, mas ainda é um compilador básico, visto que estamos no início da matéria e possuímos limitações.

Além disso, houve muito aprendizado, principalmente na questão de criar comandos um pouco mais complexos, como um switch, que demandou um esforço maior.