Lógica de programação

É disciplina que ensina como um computador entende um código. A esses códigos damos o nome de algoritmos. Um algoritmo é uma sequencia de passos que qunado bem organizados e estruturados resolvem um problema.

Exemplo

algoritmoAtravessarRua

Olha para a direita

Olhar para a esquerda

Se estiver vindo carro, não atravessar

Se não estiver vindo carro, atravessar.

Um algoritmo pode ser escrito de diversas formas, ou seja, um mesmo problema pode ter varias soluções diferentes.

Variaveis

Variaves são espaços resevados na memoria do computador para armazenar dados. A esses dados damos o nome de TIPO PRIMITIVO. Existem 4 tipos principais

Inteiro – Armazena números do tipo inteiro. Ex: 5 , 4 ,10 ,150 ,10000

Real – Armazena um numero decimal (float). Ex: 4.5 , 10.58, 150.987. obs: SEMPRE SEPARAR NUMEROS DECIMAIS POR UM PONTO, JAMAIS POR VIRGULA.

Lógico – É um valor booleano. Verdadeiro ou falso.

Caracteres – Vai armazenar dados em texto, palavra, frase (String). Ex: “Victor”, “A vida é bela”

Regras para criar variáveis

1 – Uma variável NUNCA deve começar com um numero, SEMPRE com uma letra. A partir do segundo caractere pode ser colocado um numero. Ex: n1, n2

2 – Uma variável não pode ter caracteres especiais. EX: dinheiro$. Exceto underline(\_). Porém underline não é utilizado, então caso preciso de variáveis com nomes compostos utilize o padrão camelCase. Bascicamente a primeira letra da primeira palavra em minúsculo e a primeira letras das palavas subsequentes em maiúsculo. Ex: primeiroNome

3 – Variaveis não podem ter espaços. Ex: primeiro nome

4 – Variaveis não podem ser uma palavra reservada, ou seja, uma palavra que faça parte da sintaxe da linguagem.

Operadores Aritméticos

A<- 5

B<- 3

|  |  |  |
| --- | --- | --- |
| Soma | + | A + B = 8 |
| Subtração | - | A – B = 2 |
| Divisao | / | A / B = 1.6 |
| Multiplicacao | \* | A \* B = 15 |
| Exponenciação | ^ | A ^ B = 125 |
| Módulo(Resto) | % | A % B= 2 |

Ordem de precedência

|  |  |
| --- | --- |
| Parentesis | () |
| Exponenciação | ^ |
| Multiplicação/divisao | \*/ |
| Soma/subtraçao | +- |

Estruturas condicionais

Estruturas condicionais são utilizdas em momentos em que uma situação pode erar resultados diferentes. Um exemplo disso é o exemplo do inicio do documento onde você so vai atravessa a rua se não estiver vindo carro, caso contrario você noa vai atravessar. Perceba que tem uma situação para cada codição diferente. Na programação temos duas estruturas condicionais. If..else(Se..entao) e a switch..case(escolha...caso).

Estrutura condicional If..else(Se..entao)

Se uma situação for verdadeira, execute um bloco de código. Caso não seja verdadeira execute outro bloco de código.

SE(condição for verdadeira) então

//Execute este bloco de código

Senão

//Execute este bloco de código

Fimse

Para criar condições utilizaremos dos OPERADORES RELACIONAIS. Estes operadores SEMPRE irão retornar um valor lógico.

|  |  |  |
| --- | --- | --- |
| Maior | > | A > B |
| Menor | < | A < B |
| Maior ou igual | >= | A >= B |
| Menor ou igual | <= | A <= B |
| Igual | = | A = B |
| Diferente | <> | A <> B |

Operadores Lógicos

Os operadores lógicos serão utilizados em situação em que temos mais de uma condição logica. São eles: E, OU e NÃO.

**Operador E:**

|  |  |  |
| --- | --- | --- |
| **P** | **Q** | **P E Q** |
| **V** | **V** | **V** |
| **V** | **F** | **F** |
| **F** | **V** | **F** |
| **F** | **F** | **F** |

**Operador OU:**

|  |  |  |
| --- | --- | --- |
| **P** | **Q** | **P OU Q** |
| **V** | **V** | **V** |
| **V** | **F** | **V** |
| **F** | **V** | **V** |
| **F** | **F** | **F** |

**Operador Não:**

|  |  |
| --- | --- |
| Não V | **F** |
| Não F | **V** |