

Trabalho Prático (valor 2.0)
Manual de Referência Rápido - Pseudo-código, C e Python

Entrega: 16/05/2016

Alunos (máximo 3): Eduardo Luiz Schade Soares, Matheus Ribeiro Gomes, João Eduardo Miranda da Silva, Lucas Pielak.

Este manual de referência rápido traz os principais comandos de pseudo-código, linguagem C e linguagem Python. Assim como um primeiro exemplo usando até estruturas de condicionais (decisão).

| Item | Pseudo-código | Linguagem C | Python |
|--|---|--|--|
| Tipos de dados | <u>inteiro</u> , <u>real</u> , <u>caractere</u> , <u>lógico</u> | int, float, char | <u>Int</u> , float, bool, <u>complex</u> |
| Regra geração de nomes (identificadores) | <ul style="list-style-type: none"> - começar por letra; - o segundo caractere pode ser letra, dígito ou <i>underscore</i>. - não pode conter espaço ou caractere especial | <ul style="list-style-type: none"> - começar por letra; - o segundo caractere pode ser letra, dígito ou <i>underscore</i>; - não pode conter espaço ou caractere especial. | <ul style="list-style-type: none"> - nomes de variáveis não podem ser iguais a palavras reservadas; - nomes de variáveis devem possuir como primeiro caractere uma letra; - os demais caracteres podem ser letras, números ou sublinhado; - nomes não podem ter espaços em branco; - não há diferença entre letras maiúsculas e minúsculas; |
| Constantes | podem ser <u>inteira</u> : - exemplos: 10, -10, 0 podem ser <u>real</u> : - exemplo: 1.5, 0, -3.25 podem ser <u>caractere</u> : - exemplo: "casa bonita", "juca 10"; "z"; podem ser <u>lógica</u> : V ou F | Podem ser int : - ex: 10, -10, 0; Podem ser float : - ex: 1.5, 0, -3,25; Podem ser char : - ex: "z", ou char [nº de carac.] - ex: "juca"; Não há constantes lógicas. | Podem ser <u>int</u> : - exemplos: 1, 7, 42 podem ser <u>float</u> : - exemplos: 0.5, 1.05 podem ser <u>bool</u> : - exemplos: true, false podem ser <u>complex</u> : - exemplos: a=42+7z |
| Declaração de variáveis | <u>tipo</u> <lista de variáveis>; Exemplos: <u>inteiro</u> A, B; <u>real</u> X, Y; <u>lógico</u> Teste; <u>caractere</u> Nome, Sexo; | Exemplos: int A, B; float X, Y; Não há lógicos; char | Não se declara variáveis. |
| Operadores aritméticos | Adição (+) Subtração (-) Produto (*) | Adição (+) Subtração (-) Produto (*) | Adição (+) Subtração (-) Produto (*) |

| | | | |
|------------------------|--|---|--|
| | Divisão (/) Resto (mod) Potenciação (**) Radiciação (//) | Divisão (/) Resto (%) Potenciação (**) Radiciação (//) | Divisão (/) Resto (%) Potenciação (**) Radiciação (//) |
| Operadores relacionais | Igual (=) Diferente de (≠) Maior (>) Menor (<) Maior igual (≥) Menor Igual (≤) | Igual (==) Diferente de (!=) Maior (>) Menor (<) Maior igual (>=) Menor Igual (<=) | Igual (==) Diferente de (!=) Maior (>) Menor (<) Maior igual (>=) Menor Igual (<=) |
| Operadores lógicos | Conjunção (e) Disjunção (ou) Negação (não) | && ! | and or not |
| Atribuição | ← | = | = |
| Entrada | <u>leia</u> (<variáveis>); Exemplos: <u>leia</u> (A); <u>leia</u> (A, B); | scanf ("variáveis"); Exemplo: scanf ("%f", &A); scanf ("%d, %f", &A, &B); | >>> user_input =("variável") Exemplo: >>> user_input =("%A") |
| Saída | <u>escreva</u> (); Exemplos: <u>escreva</u> (A); <u>escreva</u> ("Valor :", A); <u>escreva</u> (A+B); <u>escreva</u> ("Resultado = ", A*2); | printf (""); Exemplos: printf ("%c", A); printf ("Valor: %d", A); printf ("%c", A+B); printf ("Resultado=%f ", A*2) | print "" Exemplos: print "%c", % (A) print "Valor: %d", % (A) print "%c", % (A+B) print "Resultado= %f", % (A*2) |
| Condição simples | <u>se</u> (condição) <u>então</u> <comandos>; <u>fim-se</u> | if (condição) <comandos>; | >>> if (condição) <comando>; |
| Condição composta | <u>se</u> (condição) <u>então</u> <comandos>; <u>senão</u> <comandos>; <u>fim-se</u> | if (condição) <comandos>; else <comandos>; | >>> if (condição) <comando>; else if (condição) <comando>; |
| Múltipla escolha | <u>escolha</u> (var) <u>caso</u> v ₁ : <comandos>; <u>caso</u> v ₂ : <comandos>; <u>caso</u> v _n : <comandos>; <u>caso contrário</u> : <comandos>; <u>fim-escolha</u> ; obs: v _i deve ser um inteiro ou um caractere. | switch (var) { case 1: <com>; break ; case 2: <com>; break ; case 3: <com>; break ; default : <com>; } | def case_1(): print 'Numero 1' def case_2(): print 'Numero 2' def case_3(): |

| | | | |
|------------------------------------|---|--|--|
| | | | <pre>print 'Numero 3'</pre> <pre>def case_default():</pre> <pre>print 'Numero fora do intervalo'</pre> |
| Repetição com teste no início | <u>enquanto</u> (condição) <u>faça</u> <comandos>; <u>fim-enquanto</u> | while (condição) { <comandos>; } | >>>while (condição): <comandos>; |
| Repetição com teste no final | <u>repita</u> <comandos>; <u>até</u> (condição); ou ainda pode ser: <u>faça</u> <comandos>; <u>enquanto</u> (condição); | while (condição) { <comandos>; } do { <comandos>; } while (condição); | Não existe. |
| Repetição com variável de controle | <u>para</u> V <u>de</u> I <u>até</u> F <u>passo</u> P <u>faça</u> <comandos>; <u>fim-para</u> onde: V: variável de controle; I: valor inicial; F: valor final; P: passo (incremento ou decremento) | for (V=I; V==F; P) <comando>; | for (V=I; V==F; P) <comando>; |
| Vetores | | | |
| Matrizes | | | |
| Registros | | | |
| Arquivos | | | |

Obs:

<comandos> : significa lista de comandos, um ou mais separados por ponto e virgula.

<lista de variáveis>: uma ou mais variáveis separadas por virgula.

Exemplos: usando as três linguagens apresentadas acima resolva o seguinte problema:

Faça um algoritmo que leia o nome e o valor de compra de um determinado produto. Com base no valor de comprar calcular o valor de venda, conforme tabela fornecida. Imprimir o nome do produto e o valor de venda que foi calculado.

| Valor de compra (VC) | Valor de Venda (VV) |
|------------------------|--|
| Zero ou negativo | Valor de compra informado com problema |
| $0 < VC \leq 100$ | VC acrescido de 30% |
| $100 < VC \leq 1000$ | VC acrescido de 20% |
| $1000 < VC \leq 10000$ | VC acrescido de 10% |
| $VC > 10000$ | VC acrescido de 5% |

Pseudocódigo

Início

caractere produto;

real VC, VP, VT;

leia produto

leia VC

se ($0 < VC$ e $VC \leq 100$)

$VT \leftarrow VC * (30/100);$

$VV \leftarrow VT + VC;$

escreva ("O valor de venda do produto %c é: %1.f", produto, VV);

senão se ($100 < VC$ e $VC \leq 1000$)

$VT \leftarrow VC * (20/100);$

$VV \leftarrow VT + VC;$

escreva ("O valor de venda do produto %c é: %1.f", produto, VV);

senão se ($1000 < VC$ e $VC \leq 10000$)

$VT \leftarrow VC * (10/100);$

$VV \leftarrow VT + VC;$

escreva ("O valor de venda do produto %c é: %1.f", produto, VV);

senão se ($VC > 10000$)

$VT \leftarrow VC * (5/100);$

$VV \leftarrow VT + VC;$

escreva ("O valor de venda do produto %c é: %1.f", produto, VV);

senão escreva ("Valor de compra informado com problema");

fim se

fim se

fim se

fim se

fim se
fim

C:

```
#include <stdio.h>
#include <stdlib.h>

int main () {
    float nome, VC, VV, VT;
    printf("Insira o nome do produto: ");
    scanf("%f", &nome);
    fflush(stdin);
    printf("Insira o preço do produto: ");
    scanf("%f", &VC);
    if (VC <= 0)
        printf("Valor de compra informado com problema.\n");
    else if (VC > 10000) {
        VT = VC*0.05;
        VV = VC+VT;}
    else if (VC > 1000 && VC <= 10000){
        VT = VC*0.1;
        VV = VC+VT;}
    else if (VC > 100 && VC <=1000){
        VT = VC*0.2;
        VV = VC+VT;}
    else if (VC > 0 && VC <=100){
        VT = VC*0.3;
        VV = VC+VT;}

    printf(" Produto: %f\n Valor de venda: %.1f\n", nome, VV);
    system("pause");
    return 0;
}
```

Python:

```
produto = str(input('Informe o nome do produto: '))
VC = float(input('Informe o preço do produto: '))

if VC <= 0:
    print("Valor de compra informado com problema.")

elif VC > 10000:
    VT = VC*0.05
    VV = VC + VT
    print ("produto %s, Valor %f" %(produto) %(VV))

    elif VC > 0 and VC <= 100
    VT = VC*0.3
    VV = VC + VT
    print ("produto %s, Valor %f" %(produto) %(VV))

        elif VC > 100 and <= 1000
        VT = VC*0.2
        VV = VC + VT
        print ("produto %s, Valor %f" %(produto) %(VV))

            else VC > 1000 and <= 10000
            VT = VC*0.1
            VV = VC + VT
            print ("produto %s, Valor %f" %(produto) %(VV))
```