

Se

Aqui veremos como dizer a um algoritmo quando um conjunto de instruções deve ser executado. Esta determinação é estabelecida se uma condição for verdadeira. Mas o que seria esta condição? Ao executar um teste lógico teremos como resultado um valor verdadeiro ou falso. A condição descrita anteriormente nada mais é que um teste lógico.

Se este teste lógico resultar verdadeiro, as instruções definidas dentro do desvio condicional serão executadas. Se o teste for falso, o algoritmo pulará o trecho e continuará sua execução a partir do ponto onde o desvio condicional foi finalizado.

O desvio condicional que foi acima apresentado é considerado simples e conhecido como o comando se.

A sintaxe é respectivamente a palavra reservada se, a condição a ser testada entre parênteses e as instruções que devem ser executadas entre chaves caso o desvio seja verdadeiro.

Exemplo de Sintaxe

```
logico condicao = verdadeiro
se (condicao)
{
    //Instruções a serem executadas se o desvio for verdadeiro
}

inteiro x = 5
se (x > 3)
{
    //Instruções a serem executadas se o desvio for verdadeiro
}
```

Exemplo:

```
programa
{
    funcao inicio()
    {

        inteiro num

        escreva ("Digite um número: ")
        leia (num)

        se (num==0)
        {
            escreva ("O número digitado é 0")
        }

    }
}
```

Se-senao

Agora vamos imaginar que se a condição for falsa um outro conjunto de comandos deve ser executado. Quando iremos encontrar esta situação?

Imagine um programa onde um aluno com média final igual ou maior a 6 é aprovado. Se quisermos construir um algoritmo onde após calculada a média, seja mostrada na tela uma mensagem indicando se o aluno foi aprovado ou reprovado. Como fazer isto? Utilizando o comando se junto com o senao.

Sua sintaxe é simples, basta no termino do comando se ao lado do fechamento de chaves, colocar o comando senao e entre chaves colocar as instruções a serem executadas caso o comando se for falso

Exemplo de Sintaxe

```
logico condicao = falso
```

```
se (condicao)
```

```
{
```

```
    //Instruções a serem executadas se o desvio for verdadeiro
```

```
}
```

```
senao
```

```
{
```

```
    //Instruções a serem executadas se o desvio for falso
```

```
}
```

Exemplo:

```
programa
```

```
{
```

```
    funcao inicio()
```

```
{
```

```
    inteiro hora
```

```
    escreva ("Digite a hora: ")
```

```
    leia (hora)
```

```
    se (hora >= 6 e hora <= 18)
```

```
{
```

```
        escreva ("É dia")
```

```
}
```

```
    senao
```

```
{
```

```
        escreva ("É noite")
```

```
}
```

```
}  
}
```

Se-senao se

Agora imagine que você precise verificar a nota da prova de um aluno e falar se ele foi muito bem, bem, razoável ou mau em uma prova como fazer isto ?

Quando você precisa verificar se uma condição é verdadeira, e se não for, precise verificar se outra condição é verdadeira uma das formas de se fazer esta verificação é utilizando do se ... senao se;

A sua sintaxe é parecida com a do senao, mas usando o comando se imediatamente após escrever o comando senao.

Exemplo de Sintaxe

```
logico condicao = falso  
logico condicao2 = verdadeiro  
se (condicao)  
{  
    //Instruções a serem executadas se o desvio for verdadeiro  
}  
senao se (condicao2)  
{  
    //Instruções a serem executadas se o desvio anterior for falso e  
    este desvio for verdadeiro  
}
```

Também pode-se colocar o comando senao no final do ultimo senao se, assim quando todos os testes falharem, ele irá executar as instruções dentro do senao

Exemplo de Sintaxe

```
se (12 < 5)  
{  
    //Instruções a serem executadas se o desvio for verdadeiro  
}  
senao se ("palavra" == "texto")  
{  
    //Instruções a serem executadas se o desvio anterior for falso e  
    este desvio for verdadeiro  
}  
senao  
{  
    //Instruções a serem executadas se o desvio anterior for falso  
}
```

O exemplo a seguir ilustra a resolução do em Portugol de avisar se o aluno foi muito bem, bem, razoável ou mau em uma prova.

Exemplo

```
programa
{
    funcao inicio()
    {
        real nota
        leia(nota)
        se(nota >= 9)
        {
            escreva("O aluno foi um desempenho muito bom na prova")
        }
        senao se (nota >= 7)
        {
            escreva("O aluno teve um desempenho bom na prova")
        }
        senao se (nota >= 6)
        {
            escreva("O aluno teve um desempenho razoável na prova")
        }
        senao
        {
            escreva("O aluno teve um desempenho mau na prova")
        }
    }
}
```

Escolha-caso

Qual a melhor forma para programar um menu de, por exemplo, uma calculadora? Esta tarefa poderia ser executada através de desvios condicionais se e senão, porém esta solução seria complexa e demorada. Pode-se executar esta tarefa de uma maneira melhor, através de outro tipo de desvio condicional: o escolha junto com o caso. Este comando é similar aos comandos se e senão, e reduz a complexidade do problema.

Apesar de suas similaridades com o se, ele possui algumas diferenças. Neste comando não é possível o uso de operadores lógicos, ele apenas trabalha com valores definidos, ou o valor é igual ou diferente. Além disto, o escolha e o caso tem alguns casos testes, e se a instrução pare não for colocada ao fim de cada um destes testes, o comando executará todos casos existentes.

A sintaxe do escolha é respectivamente o comando escolha a condição a ser testada e entre chaves se coloca os casos

A sintaxe para se criar um caso é a palavra reservada caso, o valor que a condição testada deve possuir, dois pontos e suas instruções. Lembre-se de termina-las com o comando pare

Exemplo de Sintaxe

```
inteiro numero
leia(numero)
escolha(numero)
{
    caso 1:
        //Instruções caso o numero for igual a 1
    pare

    caso 2:
        // Instruções caso o numero for igual a 2
    pare

    caso 50:
        // Instruções caso o numero for igual a 50
    pare

    caso contrario:
        // Instruções caso nenhum dos casos anteriores não seja verdadeiro
}
}
```

```
caracter simbolo  
leia(simbolo)  
escolha(simbolo)  
{  
    caso 's':  
        // Instruções caso o caracter for igual a 's'  
    pare  
  
    caso '[':  
        // Instruções caso o caracter for igual a '['  
    pare  
  
    caso '*':  
        // Instruções caso o caracter for igual a '*'  
    pare  
}
```

Exemplo

```
programa
{
    funcao inicio()
    {
        inteiro valor=1
        escolha (valor)
        {
            caso 0: //testa se o valor é igual a 0
                escreva ("o valor é igual a 0")
            pare

            caso 1: //testa se o valor é igual a 1
                escreva ("o valor é igual a 1")
            pare

            caso 2: //testa se o valor é igual a 2
                escreva ("o valor é igual a 2")
            pare

            caso contrario:
                escreva ("o valor não é igual a 0, 1 ou 2")
        }
    }
}
```