in4mation HUB

Estruturas de Repetição

Conceito e Sintaxe

in4mation HUB

A IMPORTÂNCIA DE ESTRUTURAS DE REPETIÇÃO

01 **Capaz de Repetir trechos de códigos multiplas vezes**

Enquanto. **(while)**
Fazer Enquanto **(do-while)**
Para **(for)**

02 **Reaproveitamento de código**

Evitando redundância e aumentando a eficiência.

03 **Modularidade e Legibilidade**

Permite que tarefas sejam descritas de forma mais clara e concisa.

in4mation HUB

NECESSIDADE DE ESTRUTURAS DE REPETIÇÃO

Avaliação de condições para determinar se, o que e como algo deve ou não ser feito.

01 Processamento de Dados em Bancos

Bancos utilizam estruturas de repetição para processar grandes volumes de transações financeiras, como pagamentos, transferências e saques.

02 Controle de Estoque em Supermercados

Estruturas de repetição para atualizar o estoque de produtos, realizar inventários e monitorar a disponibilidade de itens nas prateleiras.

03 Análise de Dados

Pesquisadores utilizam estruturas de repetição para analisar grandes conjuntos de dados em pesquisas computacionais.

04 Processamento de Texto

Editores de texto utilizam estruturas de repetição para percorrer cada caractere ou linha de um documento, permitindo diferentes operações.

in4mation HUB

QUAL O PAPEL PRÁTICO DESSAS ESTRUTURAS?

01 **Automação de Tarefas Repetitivas:** As estruturas de repetição automatizam tarefas que precisam ser realizadas várias vezes, evitando a redundância e simplificando o código.

02 **Processamento de Dados:** Permitem o processamento de grandes volumes de dados de forma eficiente, permitindo a iteração sobre listas, arrays e outros tipos de estruturas de dados.

03 **Iteração e Controle de Fluxo:** Permitem a iteração sobre uma sequência de elementos, como caracteres em uma string, itens em uma lista ou linhas em um arquivo, proporcionando um controle detalhado sobre o fluxo de execução do programa.

in4mation HUB

EXEMPLOS TEÓRICOS

01 Validação de Usuário: (Testa correspondência de senha)

senha digitada =! senha correta

senha correta = x
senha digitada = y

senha digitada == senha correta

FIM

senha digitada = ?

in4mation HUB

02 Contador: (Conta do numero digitado até 0)

contador = x

contador > 0

contador = contador - 1

contador == 0

FIM

in4mation HUB

03 Calculo de horas de estudo: (Conta horas de estudo ao longo de uma semana)

```
dia = 1
horas_estudo = 0
total_horas = 0
```

dia < 7

dia == 7

FIM

total_horas

```
horas_estudo = x
total_horas = total_horas + horas_estudo
dia = dia + 1
```

in4mation HUB

SINTAXE DE ESTRUTURA DE REPETIÇÃO WHILE

```
1  while (condição) {
2      // Corpo do loop
3      // Código a ser repetido enquanto a condição for verdadeira
4  }
5
```

in4mation HUB

EXEMPLOS PRÁTICOS:

01 Validação de Usuário: (Testa correspondência de senha)

```c
#include <stdio.h>

int main() {
    char senha_correta[] = "senha123";
    char senha_digitada[20];

    printf("Digite sua senha: ");
    scanf("%s", senha_digitada);

    while (strcmp(senha_digitada, senha_correta) != 0) {
        printf("Senha incorreta. Tente novamente: ");
        scanf("%s", senha_digitada);
    }

    printf("Senha correta! Bem-vindo!\n");

    return 0;
}
```

in4mation HUB

02 Contador: (Conta do numero digitado até 0)

```c
#include <stdio.h>

int main() {
    int contador = 10;

    printf("Contagem regressiva:\n");
    while (contador >= 0) {
        printf("%d\n", contador);
        contador--;
    }

    printf("Fogo!\n");

    return 0;
}
```

in4mation HUB

03 Calculo de horas de estudo: (Conta horas de estudo ao longo de uma semana)

```c
#include <stdio.h>

int main() {
    int horas_estudo, dia = 1, total_horas = 0;

    printf("Contador de Horas de Estudo\n\n");

    // Loop para registrar as horas de estudo por dia
    while (dia <= 7) {
        printf("Digite o número de horas de estudo no dia %d: ", dia);
        scanf("%d", &horas_estudo);

        total_horas += horas_estudo; // Adiciona as horas de estudo ao total
        dia++; // Incrementa o contador de dias
    }

    printf("\nTotal de horas de estudo na semana: %d\n", total_horas);

    return 0;
}

```

in4mation HUB

DESAFIO

Crie um programa simples que conte e exiba a quantidade de números pares dentro de um intervalo fornecido pelo usuário.

a. O programa deve solicitar ao usuário que insira dois números inteiros que representem o início e o fim do intervalo.
b. O programa deve contar e exibir a quantidade de números pares dentro desse intervalo, incluindo os próprios limites do intervalo, se aplicável.