

Apostila de Dart: Estruturas Condicionais

Bem-vindo à apostila sobre estruturas condicionais em Dart! Este material foi projetado para ensinar como controlar o fluxo de um programa, permitindo que ele tome decisões e execute ações diferentes com base em condições específicas.

1. O que são Estruturas Condicionais?

Em programação, nem sempre queremos que o código seja executado de forma linear, do início ao fim. Muitas vezes, precisamos que o programa avalie uma situação e decida qual caminho seguir.

As estruturas condicionais permitem exatamente isso: elas testam se uma determinada condição é **verdadeira** (true) ou **falsa** (false) e, com base nesse resultado, executam um bloco de código específico.

2. A Estrutura if, else if e else

A estrutura if é a forma mais fundamental de controle de fluxo. Ela pode ser usada sozinha ou em conjunto com else if e else para criar lógicas mais complexas.

2.1. if Simples

Executa um bloco de código **apenas se** a condição for verdadeira.

Sintaxe:

```
if (condicao) {  
    // Código a ser executado se a condição for verdadeira  
}
```

Exemplo Prático: Verificador de Maioridade

```
void main() {  
    int idade = 20;  
  
    if (idade >= 18) {  
        print("Você é maior de idade. Acesso permitido!");  
    }  
}
```

Neste caso, como a variável idade (20) é maior ou igual a 18, a mensagem será impressa no console.

2.2. if / else

Oferece um caminho alternativo para quando a condição do if é falsa.

Sintaxe:

```

if (condicao) {
    // Código executado se a condição for VERDADEIRA
} else {
    // Código executado se a condição for FALSA
}

```

Exemplo Prático: Verificação de Média Escolar

```

void main() {
    double media = 5.5;

    if (media >= 6.0) {
        print("Aluno Aprovado!");
    } else {
        print("Aluno Reprovado.");
    }
}

```

Aqui, como a media (5.5) não satisfaz a condição `media >= 6.0`, o bloco de código dentro do `else` será executado.

2.3. if / else if / else

Permite testar múltiplas condições em sequência. O programa executa o bloco da **primeira condição verdadeira** que encontrar e ignora as demais.

Sintaxe:

```

if (condicao1) {
    // Executado se a condicao1 for verdadeira
} else if (condicao2) {
    // Executado se a condicao1 for falsa E a condicao2 for verdadeira
} else {
    // Executado se NENHUMA das condições anteriores for verdadeira
}

```

Exemplo Prático: Classificação de Média

```

void main() {
    double media = 8.5;

    if (media < 6.0) {
        print("Reprovado.");
    }
}

```

```

    } else if (media < 9.0) {
        print("Aprovado. Bom desempenho!");
    } else {
        print("Aprovado com Distinção! Excelente desempenho!");
    }
}

```

O programa primeiro testa se $media < 6.0$ (falso). Em seguida, testa se $media < 9.0$ (verdadeiro) e executa o código correspondente, ignorando o else final.

Exercícios if/else

1. **Verificador de Saldo:** Crie uma variável saldo e outra valorSaque. Use if/else para verificar se o saldo é suficiente. Se for, imprima "Saque realizado com sucesso!"; senão, imprima "Saldo insuficiente".
2. **Par ou Ímpar:** Crie uma variável numero e use o operador de módulo (%) com if/else para imprimir se o número é "Par" ou "Ímpar".

3. A Estrutura switch / case

A estrutura switch é uma alternativa ao if/else if/else, ideal para quando você precisa comparar uma única variável com múltiplos valores possíveis.

Vantagens:

- Pode ser mais legível quando há muitas condições.
- Ideal para comparar valores exatos (números, strings, constantes).

Sintaxe:

```

switch (variavel) {
    case valor1:
        // Código para o valor1
        break;
    case valor2:
        // Código para o valor2
        break;
    // ...outros cases
    default:
        // Código executado se nenhum case corresponder
}

```

Importante: A palavra-chave break é crucial. Ela encerra a execução do switch assim que um case correspondente é encontrado. Sem ela, o código continuaria a executar os case seguintes (comportamento chamado de "fall-through").

Exemplo Prático: Menu de Opções

```

void main() {
    String opcao = "ajuda";

    switch (opcao) {
        case "abrir":
            print("Abrindo o arquivo...");
            break;
        case "salvar":
            print("Salvando as alterações...");
            break;
        case "ajuda":
            print("Exibindo menu de ajuda.");
            break;
        default:
            print("Opção inválida!");
    }
}

```

Exercícios switch/case

1. **Dia da Semana:** Crie uma variável dia (com um número de 1 a 7). Use um switch para imprimir o nome do dia da semana correspondente (1 para "Domingo", 2 para "Segunda", etc.). Inclua um default para números inválidos.
2. **Status do Pedido:** Crie uma variável status do tipo String. Use um switch para os valores "Pendente", "Enviado" e "Entregue", imprimindo uma mensagem apropriada para cada um.

4. O Operador Ternário

O operador ternário é uma forma compacta e elegante de escrever uma estrutura if/else simples em uma única linha.

Sintaxe:

```
condicao ? expressao_se_verdadeiro : expressao_se_falso;
```

Como funciona:

1. A condicao é avaliada.
2. Se for true, o resultado de toda a expressão será expressao_se_verdadeiro.
3. Se for false, o resultado será expressao_se_falso.

Exemplo Prático: Mensagem de Status do Usuário

```

void main() {
    bool usuarioLogado = true;

    // Usando if/else tradicional
    String mensagemTradicional;
    if (usuarioLogado) {
        mensagemTradicional = "Bem-vindo!";
    } else {
        mensagemTradicional = "Por favor, faça o login.";
    }
    print(mensagemTradicional);

    // Usando o operador ternário
    String mensagemTernaria = usuarioLogado ? "Bem-vindo!" : "Por
favor, faça o login.";
    print(mensagemTernaria);
}

```

Ambos os blocos de código produzem o mesmo resultado, mas o operador ternário é muito mais conciso.

Exercícios com Operador Ternário

1. **Seleção de Cor:** Crie uma variável booleana `ehModoNoturno`. Use o operador ternário para atribuir a uma variável `corDeFundo` o valor "Preto" se for true ou "Branco" se for false. Imprima o resultado.
2. **Permissão de Voto:** Crie uma variável `idade` e use o operador ternário para atribuir a uma String a mensagem "Pode votar" ou "Não pode votar" com base na idade ser maior ou igual a 16.

Resumo e Boas Práticas

Estrutura Quando Usar

if / else if / else Para lógicas complexas, comparações de faixas (>, <, >=, <=) e múltiplas variáveis.

switch / case Para comparar uma única variável contra múltiplos valores exatos e melhorar a legibilidade.

Operador Ternário Para atribuições condicionais simples que cabem em uma única linha.