

## O comando switch

Agora que aprendemos sobre o `if`, é importante aprendermos onde ele não nos ajuda. O `if` possui um ponto fraco, que é a quantidade e complexidade de leitura do código quando precisamos de muitos `ifs`.

Imagine um programa que imprima em nossa tela o nome de um mês baseado no número que ele representa no calendário, por exemplo, Janeiro = 1, Fevereiro = 2 etc.

Vamos escrevê-lo utilizando o `if`:

```
void main() {  
  int mes = 8;  
  
  if (mes == 1) {  
    print("Janeiro");  
  }  
  if (mes == 2) {  
    print("Fevereiro");  
  }  
  if (mes == 3) {  
    print("Março");  
  }  
  if (mes == 4) {  
    print("Abril");  
  }  
  if (mes == 5) {  
    print("Maio");  
  }  
  if (mes == 6) {  
    print("Junho");  
  }  
  if (mes == 7) {  
    print("Julho");  
  }  
  if (mes == 8) {  
    print("Agosto");  
  }  
  if (mes == 9) {  
    print("Setembro");  
  }  
  if (mes == 10) {  
    print("Outubro");  
  }  
  if (mes == 11) {  
    print("Novembro");  
  }  
  if (mes == 12) {  
    print("Dezembro");  
  }  
}
```

Viram como o código fica um pouco bagunçado? Para piorar, se não tratarmos de todos os casos, o programa não vai funcionar de forma esperada. Por exemplo, nós sabemos que não existe mês de número 13, mas nosso programa não. O que ele faria?

Nada! E apesar de isso não ser tão preocupante neste caso, o usuário do seu programa poderia acreditar que o programa não funciona ou que o dispositivo dele tem algum problema.

Felizmente, o Dart tem um comando muito legal que nos permite escrever várias opções em um formato mais compacto e fácil de ler, que ainda possui uma forma de tratar todos os casos que não caem nestas opções.

Esse comando se chama `switch`, e esta é a forma padrão de uso:

```
void main() {  
  int mes = 8;  
  
  switch(mes) {  
    case 1:  
      print("Janeiro.");  
      break;  
    case 2:  
      print("Fevereiro.");  
      break;  
    case 3:  
      print("Março.");  
      break;  
    case 4:  
      print("Abril.");  
      break;  
    case 5:  
      print("Maio.");  
      break;  
    case 6:  
      print("Junho.");  
      break;  
    case 7:  
      print("Julho.");  
      break;  
    case 8:  
      print("Agosto.");  
      break;  
    case 9:  
      print("Setembro.");  
      break;  
    case 10:  
      print("Outubro.");  
      break;  
    case 11:  
      print("Novembro.");  
      break;  
    case 12:  
      print("Dezembro.");  
      break;  
    default:  
      print("Este mês não existe!");  
      break;  
  }  
}
```

```
}  
}
```

O comando `switch` permite o tratamento de vários casos diferentes de forma modular. Percebam que dentro de suas chaves ( `{ }` ) existem vários trechos de código que se repetem, alterando apenas o valor do número depois da palavra `case` , seguindo o seguinte padrão:

```
case X:  
    Y;  
    break;  
case A:  
    B;  
    break;
```

Isso pode ser traduzido em uma frase da seguinte maneira:

Caso o valor seja X, faça Y. Caso o valor seja A, faça B.

Notem o `break` após cada comando também. Ele está lá para parar a execução do `switch` , caso tenha encontrado um valor buscado.

E o que torna o uso do `switch` mais atrativo? O `default` . O `default` pode ser observado na última opção no exemplo dado, e ele é responsável por executar os comandos depois dele, caso nenhuma das opções dadas dentro do `switch` sejam acionadas. Nesse exemplo, ele imprime “Esse mês não existe!” em todos os casos que o `switch` não receba um valor entre 1 e 12.