

Linguagem Dart.

<https://dart.dev>

Aqui estão os 20% dos tópicos mais importantes de aprendizados sobre Dart ajudarão a entender 80% dele.

Tipagem: Dart é uma linguagem com tipagem estática, o que significa que as variáveis devem ser declaradas com um tipo específico e não podem ser alteradas posteriormente.

Variáveis: As variáveis em Dart podem ser declaradas usando as palavras-chave `var`, `dynamic` ou `final`. A palavra-chave `var` permite que o tipo da variável seja inferido pelo compilador, enquanto `dynamic` permite que o tipo seja determinado em tempo de execução.

Funções: As funções em Dart são definidas usando a palavra-chave `func` e podem ter um ou mais parâmetros e um tipo de retorno.

Classes e objetos: Dart é uma linguagem orientada a objetos, o que significa que o código é organizado em classes que definem objetos. Os objetos são instâncias de classes e possuem propriedades e métodos.

Herança: As classes em Dart podem herdar de outras classes, permitindo que os objetos tenham as propriedades e métodos de sua classe pai.

Interfaces: As interfaces em Dart definem um conjunto de métodos que uma classe deve implementar. Isso é útil para garantir que várias classes tenham um conjunto comum de comportamentos.

Mixins: Mixins são uma forma de compartilhar código entre várias classes sem herança. Eles são semelhantes às interfaces, mas podem ter implementações de métodos.

Coleções: As coleções em Dart incluem Listas, Mapas e Conjuntos. Listas são usadas para armazenar uma coleção ordenada de elementos, Mapas armazenam pares chave/valor e Conjuntos armazenam uma coleção de valores únicos.

Controle de fluxo: O controle de fluxo em Dart inclui condicionais como `if` e `else`, bem como laços como `while` e `for`.

Exceções: As exceções em Dart são usadas para sinalizar erros em tempo de execução e podem ser tratadas com a palavra-chave `try` e `catch`.

Bibliotecas: Bibliotecas em Dart são usadas para organizar o código em unidades independentes que podem ser importadas em outros arquivos.

Async/await: Dart oferece suporte a programação assíncrona por meio das palavras-chave "async" e "await", permitindo que o código execute operações de E/S sem bloquear a execução do programa.

Operadores: Dart oferece uma ampla gama de operadores, incluindo aritméticos, de atribuição, lógicos e bit a bit.

Métodos estáticos: Os métodos estáticos em Dart são métodos que pertencem à classe, não ao objeto. Eles podem ser chamados sem criar uma instância da classe.

Parâmetros nomeados: Dart oferece suporte a parâmetros nomeados, o que significa que os parâmetros podem ser identificados pelo nome, em vez da posição.

Parâmetros opcionais: Os parâmetros opcionais em Dart podem ser usados com colchetes [] ou chaves {} e podem ter valores padrão.

Programação funcional: Dart oferece suporte à programação funcional com funções de ordem superior, que podem ser passadas como parâmetros e retorn.

- Exemplo simples de código em Dart que imprime os números de 1 a 10 usando um laço for:

Código:

```
void main() {  
  for (int i = 1; i <= 10; i++) {  
    print(i);  
  }  
}
```

Retorno:

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```