

DART

O GUIA

PROFESSOR KAIAN NOVAIS

Sumário

Quem é o Professor Kaian Novais.....	3
O que é o DART.....	4
Sintaxe do DART.....	5
Instalação do DART.....	6
Instalação do VS Code.....	7
Variáveis.....	8

Quem é o professor Kaian?

Sou formado em Análise e Desenvolvimento de Sistemas, tenho mais de 3 anos de experiência de ensino, principalmente na rede de educação do estado de Minas Gerais. Sou mineiro por natureza e amante pela habilidade de ensino, habilidade essa que busco sempre aperfeiçoar, gastando tempo considerável para poder entregar um ensino de qualidade aos meus alunos, bem como um conteúdo que venha ser útil na vida de cada um.

Em destaque, menciono a Happy Code, a maior escola de programação de crianças e adolescentes do Brasil. Atualmente leciono nela.

Esse é meu primeiro ano na Udemy, e quero me dedicar na construção de conteúdos que venham atender meus alunos. Fico muito feliz por ter adquirido esse curso, espero muito receber umas 5 estrelas de você, hehehehe. Além das 5 estrelinhas, que aqui você possa encontrar material de qualidade, aulas de qualidade. Um forte abraço!



Essa foto eu tirei no studio da Happy Code em Campinas SP.

Chega de falar de mim e vamos ao que nos interessa.

O que é o Dart

O Dart é uma linguagem de programação fortemente tipada, inicialmente criada pela Google em 2011.

Fortemente tipada??? O que é isso ??? Calma lá, vou te explicar:

Linguagem fortemente tipada, resumindo, seria aquela em que os objetos/variáveis tem um tipo bem definido e que precisa ser informado no momento de sua declaração.

Um exemplo de linguagem que não é fortemente tipada é o PHP. Nela não é preciso definir o tipo das variáveis ao declará-las, e esse tipo pode ser alterado em tempo de execução. Assim, uma mesma variável pode receber valores numéricos, texto, booleanos, etc.

A missão inicial do Dart era substituir o JavaScript para desenvolvimento de scripts em páginas web. Porém, com a evolução da linguagem e com o passar dos anos, ela hoje pode ser considerada uma linguagem multi-paradigma, embora a linguagem apresente fortes estruturas típicas de linguagens orientadas a objeto.

Hoje, sabemos que o Dart não obteve muito sucesso em sua missão inicial em substituir o JavaScript nos navegadores. Porém, o desenvolvimento e posterior sucesso do Flutter, que é fundamentado no Dart, fez com que a linguagem voltasse à tona, atraindo a atenção de muitos desenvolvedores.

A sintaxe do Dart

O Dart possui uma sintaxe com estilo baseado no C. Isso faz com que sua sintaxe seja muito similar à linguagens atualmente populares, como Java e C#. Porém, o Dart tenta reduzir um pouco os ruídos característicos de linguagens baseadas no C.

Um simples *“Hello World”* em Dart poderia ser escrito da seguinte forma:

```
main() {  
  
  print('Hello World!');  
  
}
```

Instalação do Dart

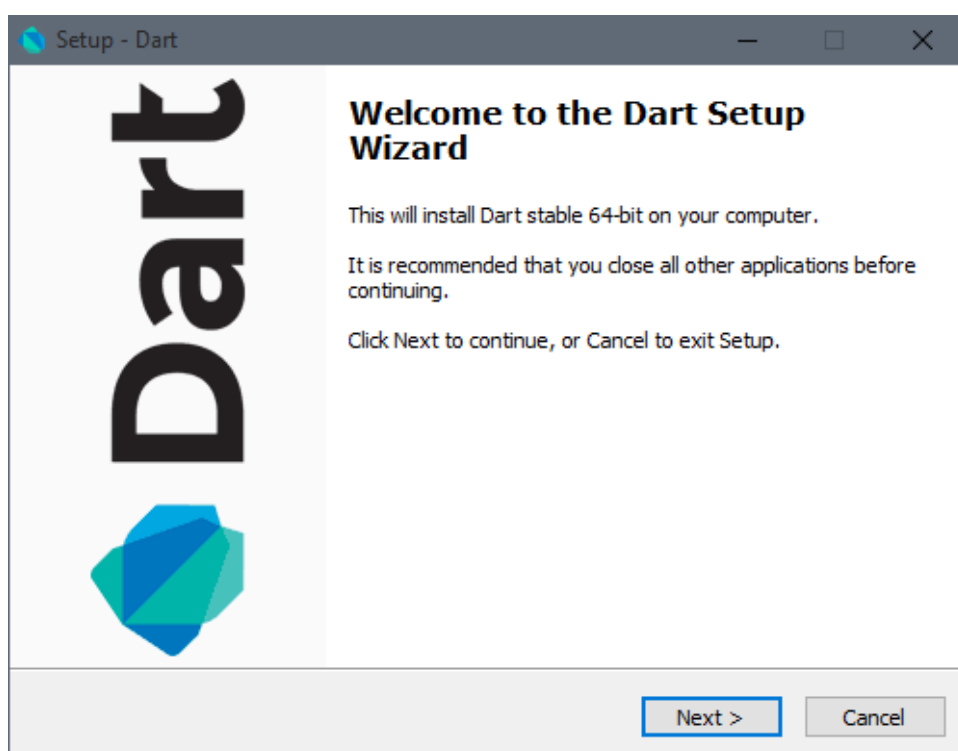
- Para realizar a instalação do Dart, você deverá ir até o site oficial

<https://dart.dev/get-dart>

- No site, existem duas formas para a instalação, a que eu escolhi foi por meio do download do arquivo, exatamente nessa guia:

<https://gekorm.com/dart-windows/>

É só escolher o seu Sistema Operacional e baixar.

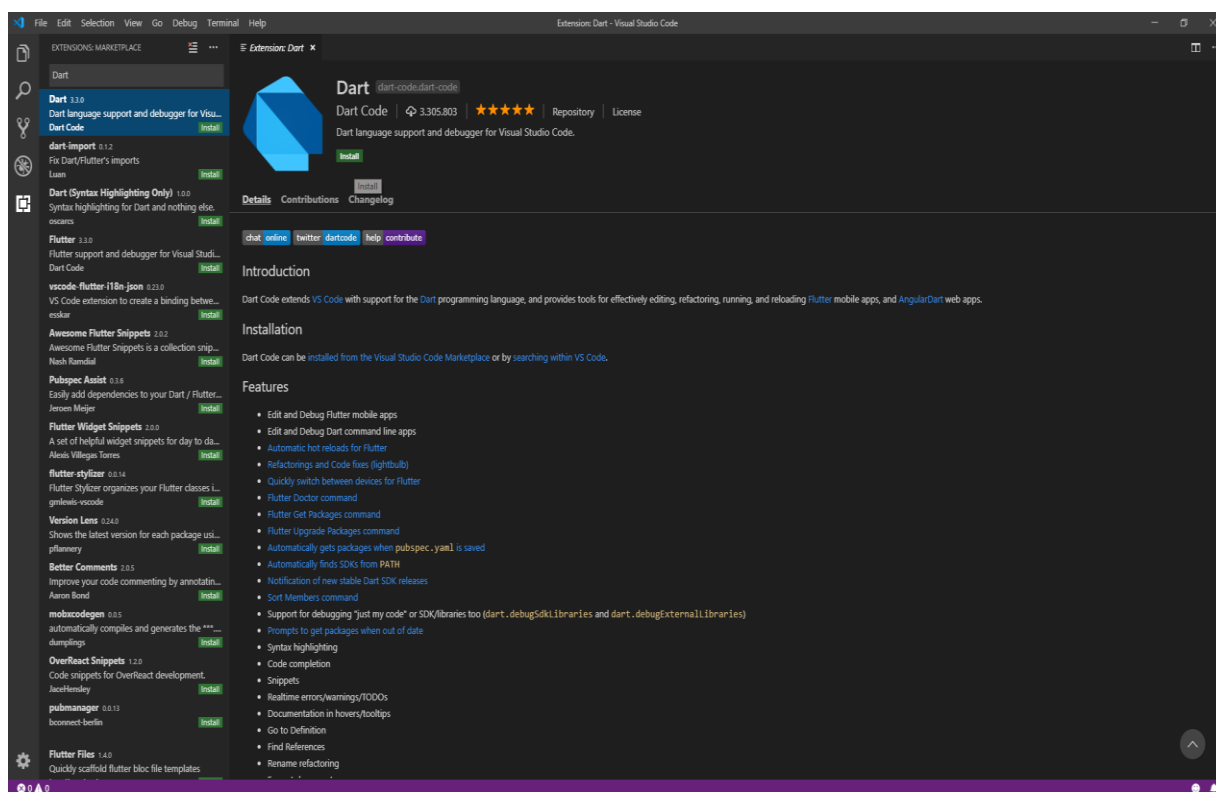


- Após baixar o arquivo, realize a instalação;
- Após a instalação, é hora de conferirmos se está tudo ok. Para isso, vá até o seu prompt de comando e digite `dart --version`;
- Estando tudo certo, deverá aparecer a versão do dart instalada.

Instalação do VS Code

Vamos utilizar o VS Code por ser uma IDE muito utilizada no mercado, potente e leve.

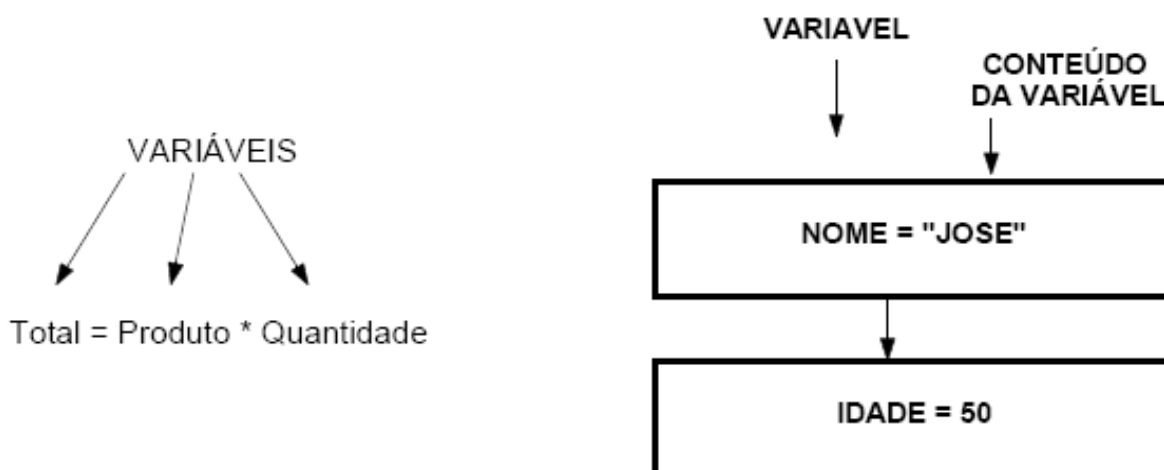
- Faça download do Visual Studio Code;
- Instale-o na configuração padrão;
- Abra o VS Code e aperte CTRL + SHIFT + X para abrir buscar extensões. Digite Dart na barra de busca e instale a versão do autor Dart Code (tem um botão verde escrito install);
- Após a instalação do plugin, feche o VS Code e abra-o novamente.



Variáveis

O aprendizado de qualquer nova ciência se inicia com o domínio de conceitos fundamentais. Para a programação, compreender o que é uma variável é essencial, pois é através delas que vamos armazenar e acessar dados em memória.

Na programação, uma **variável** é um objeto (uma posição, frequentemente localizada na memória) capaz de reter e representar um valor ou expressão.



Quando programamos em determinada linguagem, precisamos declarar uma variável, ou seja, dar existência a ela, informando ao programa o tipo dessa variável (texto ou número por exemplo), o nome de variável e o valor da mesma.

Na linguagem DART, bem como também em muitas outras, temos alguns tipos de variáveis, cada uma com uma função diferente. Cada tipo de variável é também um objeto, permitindo que o programador acesse um conjunto de funções para cada tipo de variável e, a partir dessas funções, realizar determinadas tarefas. Vejamos os tipos:

1- Number

2 - String

3 - Boolean

4 - Dynamic

5 - Function

6 - List

7 - Map

1 - Number

Dart oferece três tipos para armazenar valores numéricos. O primeiro deles é `int`, utilizado para o armazenamento de qualquer número inteiro, seja ele negativo ou positivo. O segundo é `double`, que é utilizado para o armazenamento de números de pontos flutuantes. Ambos, `int` e `double`, são subtipos de `num`. Ao declarar uma variável como `num` ela pode ser tanto um inteiro quanto um número de ponto flutuante.

Com esse tipo de variável, é possível utilizar operadores `+` (adição), `-` (subtração), `*` (multiplicação), `/` (divisão) e outros.

```
main() {  
  
  int n1 = 35;  
  
  Double n2 = 45.67;  
  
  print(n1); //resultado 35  
  
  print(n2); // resultado 45.67  
  
}
```

2 - String

Strings são cadeias de caracteres, que recebem tanto texto, quanto números ou símbolos, que podemos representar com aspas duplas ou aspas simples.

Na programação de computadores, uma **cadeia de caracteres** ou ***string*** é uma sequência de caracteres, geralmente utilizada para representar palavras, frases ou textos de um programa.

Podemos juntar duas strings através da concatenação. Veja:

```
main(){  
    String nome = 'Kaian';  
    String sobrenome = 'Novais';  
    String nomeCompleto = nome + " " + sobrenome; //o + é usado para concatenar  
    print(nomeCompleto); //resultado Kaian Novais  
}
```

3 – Boolean

Para representarmos valores booleanos no Dart utilizamos o tipo `bool` que pode ser representado por dois valores: `true` (verdadeiro) e `false` (falso). Veja um exemplo onde usamos o boolean, para nos informar se um determinado número é ímpar ou par.

```
main(){  
    bool impar;  
    num n1 = 61;  
  
    if (n1%2 == 1){  
        impar = true;  
        print(impar);  
    } else {  
        impar = false;  
        print(impar);  
    }  
}
```

```
}  
}
```

Nesse código acima, criamos uma variável do tipo bool com o nome ímpar, que poderá receber dois valores: true (verdade) e false (mentira). Essa variável irá receber esses valores, de acordo com o resultado da operação, que é: se o resto da divisão por 2 for igual a 1 ela recebe true, caso contrário, recebe false.

4 – Dynamic

No Dart existe tipo chamado dynamic, onde podemos atribuir valores de todos os outros tipos, e até mesmo modificar esses valores em tempo de execução, ou seja, declarando uma variável do tipo dynamic, ela poderá receber números, textos, etc.

```
main(){  
    dynamic nome = "Kaian Novais";  
    dynamic idade = 26;  
    print(nome); // Kaian Novais  
    print(idade); // 26  
    idade = "26 anos";  
    print(idade); // 26 anos  
  
}
```

5 – Function

Uma função é um trecho de código, um processo ou rotina, responsável pela execução de uma tarefa específica, podendo ser executado múltiplas vezes durante a execução do programa. No Dart, funções são um tipo de dado e podemos tratá-las como qualquer outro dado: podemos passá-las como parâmetro, atribuir a variáveis e mais.

Uma função pode receber parâmetros (dados necessários para sua execução) e retornar outros valores gerados a partir de um processamento qualquer. Entretanto, nenhum dos dois é obrigatório: uma função pode não precisar de parâmetros e não retornar nada.

```
main(){  
    mensagem(); // Esse é o seu curso de Dart com o professor Kaian;  
}  
mensagem(){  
    print('Esse é o seu curso de Dart com o professor Kaian');  
}
```

Para recebermos parâmetros declararamos cada uma das variáveis que a função pode receber, e seus tipos, dentro dos parênteses:

```
main(){  
    mensagem("Kaian Novais", "Professor da Udemy");  
}  
void mensagem(String nome, String profissao) {  
  
    print("Nome: $nome");  
    print("Profissão: $profissao");  
}
```

```
}
```

No exemplo acima temos um exemplo de passagem de parâmetros obrigatórios, e posicionais (mudar a ordem de passagem dos parâmetros influencia a execução da função). Se um desses parâmetros não for passado na execução da função, um erro será gerado.

No Dart é possível definir parâmetros opcionais posicionais e nomeados, mas não ambos. Vejamos um exemplo de cada:

Parâmetro Opcional Posicional

Para definir parâmetros opcionais posicionais, devemos agrupar esses parâmetros entre colchetes.

Atenção: Parâmetros opcionais posicionais precisam ser declarados após os parâmetros obrigatórios:

```
main(){  
  mensagem("Kaian Novais");  
}  
void mensagem(String nome, [String profissao]) {  
  
  print("Nome: $nome");  
  
}
```

6 – List

Quando necessitamos trabalhar com arrays (estrutura de dados que armazena valores identificador a partir de um index), podemos contar com o objeto List.

```
main(){  
    List listas = ["Manga", "Uva", "Laranja"];  
    print(listas);  
  
}
```

7 – Map

Um Map é uma estrutura similar à uma List. Entretanto, diferentemente da List onde cada valor tinha um index (número inteiro) correspondente, em um Map cada valor terá uma chave (que pode ser um objeto qualquer) correspondente. Assim como a List, o Map também pode ser criado através da criação de uma instância ou de forma literal.

