**Aula IV - DevAppII**

**Lógica em Dart**

**Fixando os conteúdos**

**Trabalhando com Funções**

**Biblioteca Math**

**Math Random**

**Return**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Fontes:***

[*https://github.com/montonurb/dart-exercises/tree/master/fundamentos*](https://github.com/montonurb/dart-exercises/tree/master/fundamentos)

[*https://www.ibm.com/docs/pt-br/pureapplication-service/2.3.1.0?topic=language-operators-expressions*](https://www.ibm.com/docs/pt-br/pureapplication-service/2.3.1.0?topic=language-operators-expressions)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Fixando Conteúdos**

**Review de Conteúdos - Aulas 2 e 3**

1. **Operadores Aritméticos**
2. **Operadores Lógicos**
3. **Operadores Relacionais**
4. **IF, Else e Else IF**
5. **Switch Case**

|  |
| --- |
| **Observação quanto aos operadores:** |

**Operadores Aritméticos ( +, -, \*, /, \*\*, % )**

Os operadores aritméticos executam operações matemáticas, como adição e subtração com operandos. **Há dois tipos de operadores matemáticos: unário e binário. Os operadores unários executam uma ação com um único operando.** **Operadores binários executam ações com dois operandos.** Em uma expressão complexa, (dois ou mais operandos), a ordem de avaliação depende de regras de precedência.

**Operadores Aritméticos Unários**

Operadores unários são operadores aritméticos que desempenham uma ação em um único operando. A linguagem de script reconhece o negativo do operador unário (-).

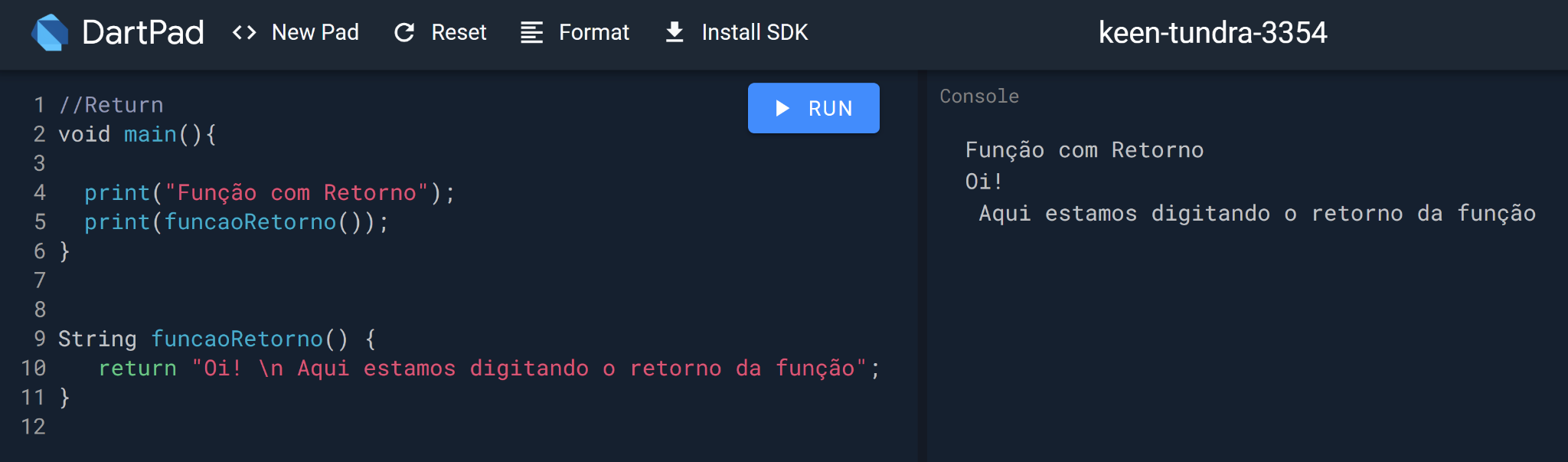
O operador unário negativo inverte o sinal de uma expressão de positivo para negativo ou vice-versa. O efeito geral é multiplicar o número por -1.

**Funções & Function**

As funções no Dart se comportam de maneira muito semelhante às funções no JavaScript. Eles podem ser definidos como um método semelhante à nossa void função, bem como, eles se comportam como objetos de primeira classe, o que significa que podem ser armazenados em uma variável , passados ​​como um argumento ou retornados como um valor de retorno normal de uma função.

**Return**

Às vezes, podemos querer que uma função retorne algum valor ao ponto de onde foi chamada. No Dart como nas outras linguagens, há uma palavra-chave de retorno que permite que uma função retorne o valor. A instrução de retorno é opcional; se não for especificada, a função retornará nulo. **Só pode haver uma instrução de retorno em uma função.**



**Dart:math library - Biblioteca Matemática**

**Classe Random**

Um gerador de valores aleatórios bool, int ou double. A implementação padrão fornece um fluxo de bits pseudo-aleatórios que não são adequados para fins criptográficos.