

Variáveis, Valores & Tipos

Total de pontos 31/32

Esse questionário vai reforçar os conceitos que você está aprendendo. Ao resolver estas questões você estará se tornando um programador mais competente!

✓ Qual o menor elemento em um programa que expressa uma ação a ser executada? *

1/1

☒ Uma declaração (statement) ✓

☐ Uma expressão

✓ A combinação de um ou mais valores, constantes, variáveis, operadores e funções que a linguagem interpreta e usa para produzir outro valor é: *

1/1

☐ Uma declaração (statement)

☒ Uma expressão ✓

✓ Quais são parênteses? *

1/1

☒ () ✓

☐ []

☐ {}

✓ Quais são colchetes? *

1/1

☐ ()

☒ [] ✓

☐ {}



✓ Quais são chaves? *

1/1

☐ ()

☐ []

☒ {}

✓

✓ A abrangência de uma variável designa onde no código você pode acessar esta variável, e atribuir ou ler valores dela. *

1/1

☒ Verdade

✓

☐ Mentira

✓ Um tipo de dados primitivo é um tipo composto, criado a partir dos outros tipos básicos que já vem de fábrica na linguagem.

1/1

*

☐ Verdade

☒ Mentira

✓

Feedback

Tipo primitivo e tipo básico são a mesma coisa. Tipos compostos são criados a partir de tipos primitivos/básicos, mas tipos primitivos/básicos não são tipos compostos.

✓ O tipo int é um tipo de dados primitivo. *

1/1

☒ Verdade

✓

☐ Mentira



✗ O tipo string é um tipo de dados composto. *

0/1

☒ Verdade

✗

☐ Mentira

Resposta correta

☒ Mentira

Feedback

String é um tipo de dados primitivo.

✓ Um tipo de dados composto permite que você crie estruturas formadas de outros tipos. *

1/1

☒ Verdade

✓

☐ Mentira

✓ Quando declaramos uma variável com a palavra-chave var e não atribuímos nenhum valor a esta variável, o compilador designa para esta um valor padrão, chamado de valor zero. *

1/1

☒ Verdade

✓

☐ Mentira



✓ Palavras-chave servem a propósitos específicos, mas fora isso podem ser usadas livremente ao longo do programa. * 1/1

☐ Verdade

☒ Mentira



Feedback

Palavras-chave somente podem ser utilizadas para seus propósitos específicos. Não podem ser utilizadas livremente.

✓ Palavras-chave e palavras reservadas designam coisas diferentes. * 1/1

☐ Verdade

☒ Mentira



Feedback

Palavras-chave e palavras reservadas são dois nomes para a mesma coisa.

✓ Uma palavra-chave somente pode ser usada pra seu propósito específico e para declarar variáveis. * 1/1

☐ Verdade

☒ Mentira



Feedback

Uma palavra-chave somente pode ser usada somente para seu propósito específico. Nada mais.



✓ Em "2 + 2", os números 2 são os operadores. *

1/1

- ☐ Verdade
- ☒ Mentira

**Feedback**

Os números são os operandos. O sinal é o operador.

✓ O termo "package" é uma palavra-chave. *

1/1

- ☒ Verdade
- ☐ Mentira



✓ O termo "variable" é uma palavra-chave. *

1/1

- ☐ Verdade
- ☒ Mentira

**Feedback**

O termo "var" consta na lista de palavras chave, mas "variable" não.

✓ O "ponto de entrada" para todos os programas é a função main(), que deve ficar dentro do package main. *

1/1

- ☒ Verdade
- ☐ Mentira



✓ O operador curto de declaração pode ser usado ao invés de var em todas as situações. * 1/1

☐ Verdade

☒ Mentira



Feedback

O operador curto não pode ser utilizado fora de um code block. Para criar variáveis com package-level scope utiliza-se a palavra chave "var".

✓ Quando vemos "fmt.Println()", isto está chamando a função Println() que pertence ao package "fmt". * 1/1

☒ Verdade

☐ Mentira



✓ "Identificador" é o nome atribuído a uma variável, função ou constante. * 1/1

☒ Verdade

☐ Mentira



✓ Para utilizar uma função, variável ou constante de outro package utiliza-se o formato "package-ponto-identificador". Por exemplo, fmt.Println(). * 1/1

☒ Verdade

☐ Mentira



✓ Qual o caractere que permite "jogar fora" um valor? Ou seja, qual caractere permite que você diga ao programa que não vai utilizar o valor retornado por uma função? *

1/1

☐ #☐ @☒ -

✓

☐ \$☐ Isso é pegadinha...

✓ Em Go, não é permitindo haver variáveis que não sejam utilizadas. *

1/1

☒ Verdade

✓

☐ Mentira

✓ Uma função cujo parâmetro é "...interface{}" é uma função variádica. Isso significa que você pode passar à função um número pré-determinado de valores. *

1/1

☐ Verdade☒ Mentira

✓

Feedback

Você pode passar um número ilimitado de valores a uma função variádica.

✓ Todo valor em Go pertence também ao tipo interface vazia, representado pela notação "interface{}". *

1/1

☒ Verdade

✓

☐ Mentira

✓ "2+3" é uma declaração/statement, não uma expressão. * 1/1

☐ Verdade

☒ Mentira

✓

Feedback

"2 + 3" é uma expressão.

✓ "2+3" é uma expressão, não uma declaração/statement. * 1/1

☒ Verdade

✓

☐ Mentira

✓ Se eu quiser salvar o resultado de um format printing em uma variável, posso usar a função `fmt.Sprintf()`. * 1/1

☒ Verdade

✓

☐ Mentira

✓ Em Go, podemos criar nossos próprios tipos. * 1/1

☒ Verdade

✓

☐ Mentira



✓ Falando de tipos, em Go utilizamos o termo "coerção," diferentemente de Java, por exemplo, onde utiliza-se "conversão." *

1/1

☐ Verdade

☒ Mentira



Feedback

Em Go utiliza-se o termo conversão.

✓ Todo tipo criado pelo programador possui um tipo subjacente. 1/1 *

☒ Verdade

☐ Mentira



Este conteúdo não foi criado nem aprovado pelo Google. - Termos de Serviço - Termos Adicionais

Google Formulários

