

REVISÃO

Sintaxe Java

Variáveis Java

Tipos de dados Java

Operadores Java

Sequências Java

Java Matemática

Booleanos Java

Java se...outro

Troca Java

Loops Java

Matrizes Java

Métodos Java

Classes/objetos Java

Exceções Java

Criar variável

Criar telas

Ler

Operações

Funções / métodos

Navegação entre telas

Banco

Sintaxe Java

Insira a parte que falta no código abaixo para gerar "Hello World".

```
public class MyClass {  
    public static void main(String[] args) {  
        ..("Hello World");  
    }  
}
```

Sintaxe Java

Comentários em Java são escritos com caracteres especiais. Insira as peças que faltam:

This is a single-line comment

This is a multi-line comment

Variáveis Java

Crie uma variável chamada `carName` e atribua o valor `Volvo` a ela.

```
  =  ;
```

Variáveis Java

Crie uma variável chamada `maxSpeed` e atribua o valor `120` a ela.

```
  =  ;
```

Variáveis Java

Exiba a soma de `5 + 10`, usando duas variáveis: `x` e `y`.

```
int x = 5;  
int y = 10;  
System.out.println(x + y);
```

Variáveis Java

Crie uma variável chamada `z`, atribua `x + y` a ela e exiba o resultado.

```
int x = 5;  
int y = 10;  
  = x + y;  
System.out.println();
```

Variáveis Java

Preencha as partes que faltam para criar três variáveis do mesmo tipo, usando uma **lista separada por vírgulas** :

```
_____ x = 5 _____ y = 6 _____ z = 50;  
System.out.println(x + y + z);
```


Tipos de dados Java

Adicione o tipo de dados correto para as seguintes variáveis:

```
 myNum = 9;  
 myFloatNum = 8.99f;  
 myLetter = 'A';  
 myBool = false;  
 myText = "Hello World";
```

Tipos de dados Java

`byte`, `short`, `int`, `long`, `float`, `double`, `boolean` e `char` são chamados:

data types.

Tipos de dados Java

Conversão de tipo - converta o seguinte `double` tipo (`myDouble`) em um `int` tipo:

```
double myDouble = 9.78d;  
int myInt =  myDouble;
```

Operadores Java

Multiplique 10 por 5 e imprima o resultado.

```
System.out.println(10 * 5);
```

Operadores Java

Divida 10 por 5 e imprima o resultado.

```
System.out.println(10 / 5);
```

Operadores Java

Use o operador correto para aumentar o valor da variável `x` em `1`.

```
int x = 10;
```

```
x;
```

Operadores Java

Use o operador **de atribuição de adição** para adicionar o valor 5 à variável x.

```
int x = 10;
```

```
x  5;
```

Java Strings

Preencha a parte que falta para criar uma `greeting` variável do tipo `String` e atribua a ela o valor `Hello`.

```
 greeting = ;
```


Java Strings

Converta o valor de `txt` para maiúsculas.

```
String txt = "Hello";  
System.out.println();
```

Java Strings

Use o operador correto para **concatenar** duas strings:

```
String firstName = "John ";  
String lastName = "Doe";  
System.out.println(firstName  lastName);
```

Java Strings

Use o método correto para **concatenar** duas strings:

```
String firstName = "John ";  
String lastName = "Doe";  
System.out.println(firstName.(lastName));
```

Java Strings

Retorne o **índice** (posição) da primeira ocorrência de "**e**" na seguinte string:

```
String txt = "Hello Everybody";  
System.out.println(txt.());
```

Java Matemática

Use o método correto para encontrar o **valor mais alto** de `x` e `y`.

```
int x = 5;  
int y = 10;  
Math.(x, y);
```

Use o método correto para encontrar a **raiz quadrada** de `x`.

```
int x = 16;  
Math.(x);
```

Java Matemática

Use o método correto para retornar um número aleatório entre 0 (inclusivo) e 1 (exclusivo).

```
Math.  ;
```

Booleanos Java

Preencha as partes que faltam para imprimir os valores `true` e `false` :

```
 isJavaFun = true;  
 isFishTasty = false;  
System.out.println(isJavaFun);  
System.out.println(isFishTasty);
```


Booleanos Java

Preencha as partes que faltam para imprimir o valor `true`:

```
int x = 10;  
int y = 9;  
System.out.println(□ □ □);
```

Java If...Else

Imprima "Hello World" se `x` for **maior que** `y` .

```
int x = 50;  
int y = 10;  
☐ (x ☐ y) {  
    System.out.println("Hello World");  
}
```

```
int x = 50;  
int y = 10;  
if (x > y) {  
    System.out.println("Hello World");  
}
```

Java If...Else

Imprima "Hello World" se for `x` igual `a` `y` .

```
int x = 50;  
int y = 50;  
☐ (x ☐ y) {  
    System.out.println("Hello World");  
}
```

Java If...Else

Imprima "Sim" se `x` for igual a `y`, caso contrário imprima "Não".

```
int x = 50;  
int y = 50;  
if (x == y) {  
    System.out.println("Yes");  
} else {  
    System.out.println("No");  
}
```

Java If...Else

Imprima "1" se `x` for igual a `y`, imprima "2" se `x` for maior que `y`, caso contrário imprima "3".

```
int x = 50;
int y = 50;
if (x == y) {
    System.out.println("1");
} else if (x > y) {
    System.out.println("2");
} else {
    System.out.println("3");
}
```

Java If...Else

Insira as partes que faltam para completar a seguinte "instrução abreviada `if...else`" (**operador ternário**):

```
int time = 20;  
String result = time < 18  "Good day."  "Good evening.";   
System.out.println(result);
```

Java Switch

Insira as partes que faltam para completar a seguinte `switch` declaração.

```
int day = 2;  
switch () {  
     1:  
        System.out.println("Saturday");  
        break;  
     2:  
        System.out.println("Sunday");  
        ;  
}
```

Java Switch

Complete a `switch` instrução e adicione a **palavra-chave** correta no final para especificar algum código a ser executado se não houver correspondência de maiúsculas e minúsculas na `switch` instrução.

```
int day = 4;
switch ( ) {
    1:
        System.out.println("Saturday");
        break;
    2:
        System.out.println("Sunday");
        ;
    :
        System.out.println("Weekend");
}
```


Java Loops

Imprima `i` até que `i` seja menor que 6.

```
int i = 1;  
while (i < 6) {  
    System.out.println(i);  
    i++;  
}
```

Java Loops

Use o `do/while` loop para imprimir `i` enquanto `i` for menor que 6.

```
int i = 1;  
 {  
    System.out.println(i);  
    i++;  
}  
 (i < 6);
```

Java Loops

Use um `for` loop para imprimir "Sim" 5 vezes:

```
for (int i = 0; i < 5; i++) {  
    System.out.println("Sim");  
}
```

Java Loops

Percorra os itens da `cars` matriz.

```
String[] cars = {"Volvo", "BMW", "Ford"};
for (String i : cars) {
    System.out.println(i);
}
```

Java Loops

Print a loop se for i 5.

```
for (int i = 0; i < 10; i++) {  
    if (i == 5) {  
        ;  
    }  
    System.out.println(i);  
}
```

Java Loops

No loop, quando o valor for "4", pule diretamente para o próximo valor.

```
for (int i = 0; i < 10; i++) {  
    if (i == 4) {  
        ;  
    }  
    System.out.println(i);  
}
```

Matrizes Java

Crie um array do tipo `String` chamado `cars`.

```
String[] cars = {"Volvo", "BMW", "Ford"};
```

Matrizes Java

Imprima o segundo item da `cars` matriz.

```
String[] cars = {"Volvo", "BMW", "Ford"};  
System.out.println();
```


Matrizes Java

Altere o valor de "Volvo" para "Opel", na `cars` matriz.

```
String[] cars = {"Volvo", "BMW", "Ford"};  
 = ;  
System.out.println(cars[0]);
```

Matrizes Java

Descubra quantos elementos o `cars` array possui.

```
String[] cars = {"Volvo", "BMW", "Ford"};  
System.out.println();
```

Matrizes Java

Percorra os itens da `cars` matriz.

```
String[] cars = {"Volvo", "BMW", "Ford"};
for (String i : cars) {
    System.out.println(i);
}
```

Matrizes Java

Insira as partes que faltam para criar uma matriz **bidimensional** .

```
 myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };
```

Java Methods

Insira a parte que falta para `myMethod` chamar `main`.

```
static void myMethod() {  
    System.out.println("I just got executed!");  
}  
  
public static void main(String[] args) {  
    ;  
}
```

Java Methods

Insira a parte que falta para ligar `myMethod` duas `main` **vezes** .

```
static void myMethod() {  
    System.out.println("I just got executed!");  
}  
  
public static void main(String[] args) {  
    ;  
    ;  
}
```

Java Methods

Adicione um `fname` parâmetro do tipo `String` e `myMethod` produza "John Doe".

```
static void myMethod( ) {  
    System.out.println( + " Doe");  
}  
  
public static void main(String[] args) {  
    myMethod("John");  
}
```

Java Methods

Insira a parte que falta para imprimir o número 8 `main`, usando uma **palavra-chave** específica dentro `myMethod`:

```
static int myMethod(int x) {  
     5 + x;  
}  
  
public static void main(String[] args) {  
    System.out.println(myMethod(3));  
}
```


Java Classes/Objects

Crie uma classe chamada `MyClass` .

```
public class
```

Java Classes/Objects

Crie um objeto `MyClass` chamado `myObj` .

```
  = new  ();
```

Java Classes/Objects

Use `myObj` para acessar e imprimir o valor do `x` atributo de `MyClass`.

```
public class MyClass {  
    int x = 5;  
  
    public static void main(String[] args) {  
        MyClass myObj = new MyClass();  
        System.out.println(.);  
    }  
}
```

Java Classes/Objects

Chame `myMethod` o objeto.

```
public class MyClass {  
    public void myMethod() {  
        System.out.println("Hello World");  
    }  
  
    public static void main(String[] args) {  
        MyClass myObj = new MyClass();  
        myObj.myMethod();  
    }  
}
```

Java Classes/Objects

Crie e chame um **construtor de classe** de `MyClass`

Siga os comentários para inserir as partes que faltam no código abaixo:

```
// Create a MyClass class
public class _____ {
    int ____; // Create a class attribute x

    // Create a class constructor for the MyClass class
    public _____() {
        x = ____; // Set the initial value for the class attribute x to 5
    }

    public static void main(String[] args) {
        // Create an myObj object of class MyClass (This will call the constructor)
        MyClass _____ = new MyClass();
        // Print the value of x
        System.out.println(_____.__);
    }
}
```

Java Classes/Objects

A classe abaixo não deve ser herdada por outras classes. Adicione o modificador correto:

```
 class MyClass
```

Java Classes/Objects

Preencha as partes que faltam para importar a `java.util.Scanner` classe da API Java:

```
  .  .  ;
```

Java Classes/Objects

A `Car` classe deve herdar os atributos e métodos da `Vehicle` classe. Adicione a palavra-chave correta para tornar isso possível.

```
class Car  Vehicle
```