

Lógica de Programação: Aula 02 - Uga, Uga!

Instruções primitivas, variáveis e tipos de dados



Olá! Revisão e Objetivos



- Bem-vindo(a) de volta!
- Na aula anterior, aprendemos que um algoritmo é uma receita (sequência de passos) e a Linguagem de Programação é o idioma que usamos para ensiná-la ao computador.

Objetivos de hoje



- Definir o que são instruções primitivas.
- Conhecer os comandos para mostrar dados (**saída**) e receber dados (**entrada**).
- Entender o que são variáveis e os tipos de dados.
- Começar a utilizar variáveis em nossos programas.

**Acessem:
ide-potigol-brasil.duckdns.org**





Digite seu código Potigol abaixo e clique em Executar.

```
1
```

Entrada do Programa (se houver):

Digite aqui os valores que o 'leia' vai pedir, um por linha.

Executar

Saída:

Instruções Primitivas



- Espera aí, terei de voltar ao tempo das cavernas?
- Calma! As instruções primitivas são apenas os **comandos mais básicos e essenciais de uma linguagem**.



Instruções Primitivas



- Elas permitem que nossos programas:
 - Mostrem informações na tela (**Saída**).
 - Recebam dados do usuário (**Entrada**).
 - Guardem esses dados na memória do computador (Armazenamento).
 - Coloquem um valor **dentro** de um local de armazenamento (Atribuição).
- Sem elas, nossos programas seriam **estáticos** e só poderiam ser usados por quem soubesse programar!

Comandos de Saída: Mostrando Coisas na Tela



- Os **comandos de saída** fazem com que sejam exibidos na tela do computador os dados e mensagens de saída de um algoritmo.
- Há dois comandos de saída no *Potigol*, os quais **permitem "imprimir" números ou textos na tela**. Conheça o primeiro comando: o *escreva*. Você já viu esse comando na aula anterior, no exemplo do "*Olá, mundo!*", recorda-se dele? Reveja como é o comando:

```
1 escreva "Olá, Mundo!"
```

Desafio 1

- Na sua IDE, vamos recriar um algoritmo utilizado na aula passada. Crie um programa, ou algoritmo, que imprima na tela “ Olá, Novo Mundo!”
- Utilize o comando *escreva* “”



Comandos de Saída



- Agora que você aprendeu como funciona o comando escreva, veja como funciona o comando *imprima* “ ”.
- Este também serve para **imprimir informações de seu algoritmo na tela**, mas há uma pequena diferença entre ele e o escreva.

Desafio 2

- Copie o seguinte algoritmo e execute.
- Qual a diferença do comando `escreva` “” para o comando `imprima””?`

```
1 # Exemplo de código.  
2 imprima "Olá, Mundo!"  
3  
4 escreva "Como vai você?"  
5  
6 imprima "Hoje é um belo dia para estudar Lógica de Programação."  
7  
8 escreva "Gostei desses comandos!"
```



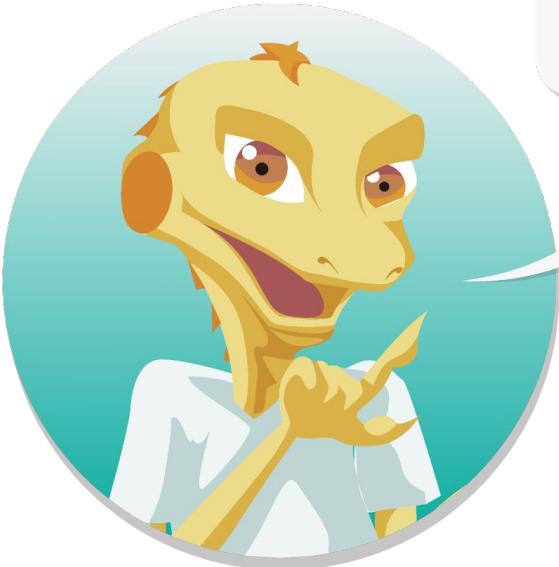
Desafio 3

- Observe o código ao lado.
- O que você acha que vai aparecer na tela quando ele for executado?

```
1 imprima "Lógica de "
2 imprima "Programação "
3 escreva "é "
4 escreva "demais!"
```



Pense por um minuto e discuta com seu colega!



Você percebeu a diferença entre os comandos **escreva** e **imprima**?



A Resposta: A Quebra de Linha

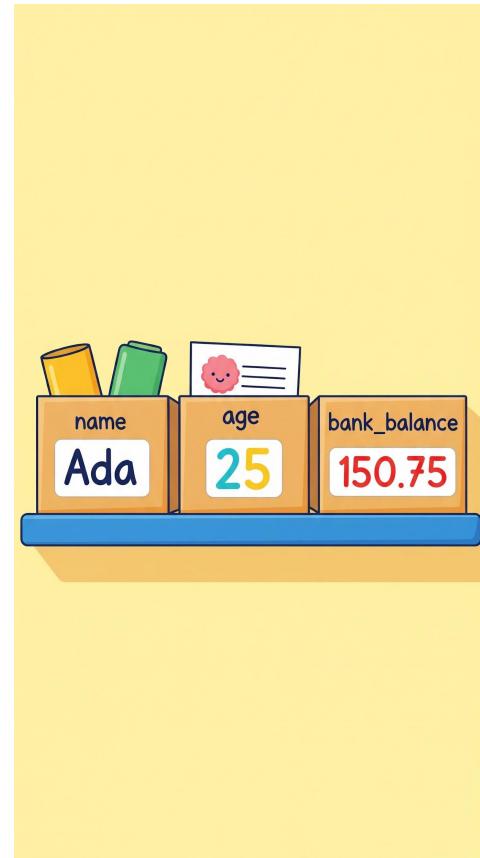


- A Diferença Fundamental:
 - *imprima*: Escreve o texto e mantém o cursor na mesma linha.
 - *escreva*: Escreve o texto e depois pula para a linha de baixo (realiza uma "quebra de linha").
- Por isso *Lógica de Programação* é ficou tudo junto e *demais!* foi para a linha seguinte.

Guardando Dados: O que são Variáveis?



- Para **receber** e **manipular** **dados**, precisamos de um lugar para **guardá-los**. Esse lugar é a variável.
- Pense em uma variável como uma **caixa etiquetada** na memória do computador, pronta para armazenar uma informação.
 - A **etiqueta** é o **nome** da variável.
 - O **conteúdo** da **caixa** é o **valor** da variável.





Declarando Variáveis em Potigol

- Para criar (ou "declarar") uma variável em Potigol, a sintaxe é simples:

```
1 var nome_da_variavel := valor
```

- Vamos quebrar em partes...



Declarando Variáveis em Potigol

var: Palavra-chave que avisa: "**Vou criar uma variável!**".

nome_da_variavel: O nome (etiqueta) que você escolhe para a sua "caixa".

:= : É o operador de atribuição. Significa "recebe" ou "guarda".

valor: A informação que você quer guardar na caixa.

```
1 var nome_da_variavel := valor
```



Exemplo

```
1 var ano_atual := 2025
2 var saudacao := "Olá, turma!"
3 var nome := "Fulano"
4 var idade := 10
5 var aniversario 10/19/1999
```

As Regras para Nomes de Variáveis



- Para criar *nomes (identificadores)*, existem algumas regras importantes, válidas para a maioria das linguagens:
- Pode usar (recomendação):
 - Usar letras e números.
 - Usar o caractere underline (_).
 - Começar sempre com uma letra ou underline.
 - Ex: *nota1, nome_completo, _idade*

As Regras para Nomes de Variáveis



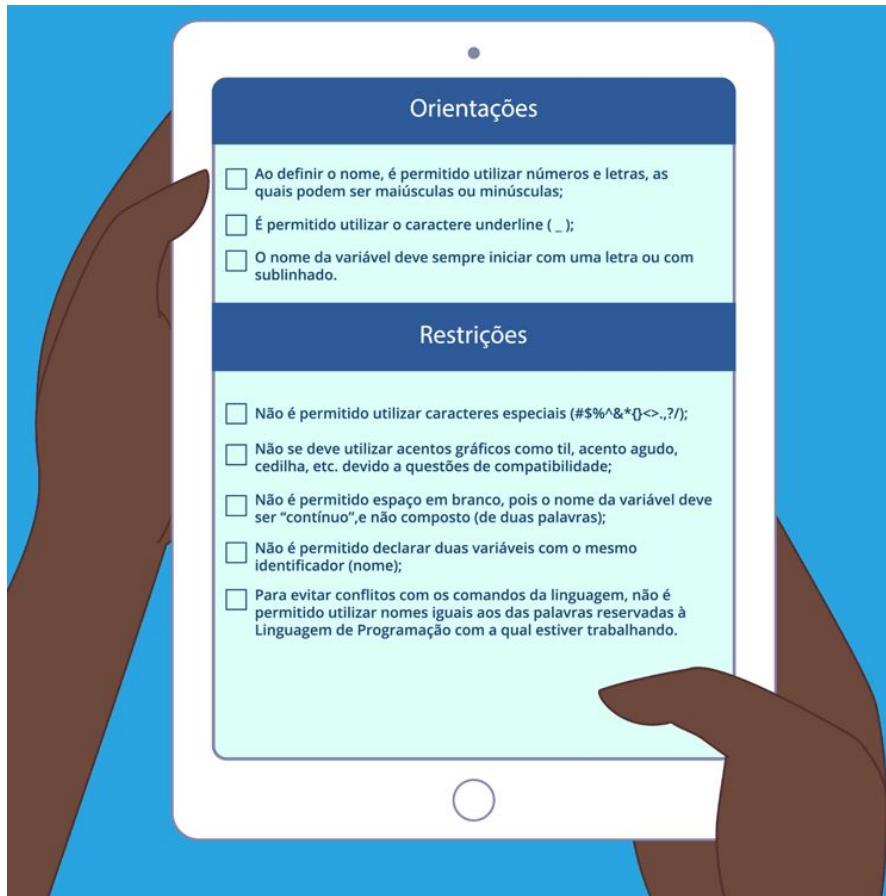
- X Restrições (Não pode fazer):
 - Usar caracteres especiais como !, @, #, \$, %.
 - Usar acentos ou cedilha (~, ´, ç).
 - Usar espaços em branco.
 - Usar o mesmo nome para duas variáveis diferentes.
 - Usar palavras reservadas da linguagem (como var, escreva, leia_inteiro).
 - Ex: *1nota, nome completo, valor\$*

Desafio 4

- 1) Quais dos nomes de variáveis abaixo são VÁLIDOS em Potigol?
- a) 2alunos
 - b) cidade_natal
 - c) endereço
 - d) valor%
 - e) nomeAluno
 - f) var



Viu como foi fácil?



Desafio 5

- Crie 6 variáveis que recebem dados diferentes, como:
 - um nome qualquer;
 - uma idade qualquer;
 - uma apresentação qualquer;
 - um endereço qualquer;
 - uma pergunta qualquer;
 - uma resposta;



Comandos de Entrada e Tipos de Dados



Agora que temos onde guardar (variáveis), como pedimos para o usuário nos dar uma informação? Com os comandos de entrada!

Cada comando é especializado em ler um tipo de dado diferente.

Tipo de Dado	O que é?	Comando em Potigol	Exemplo de Uso
Inteiro	Números sem casas decimais	<i>leia_inteiro</i>	Idade, quantidade
Real (Ponto Flutuante)	Números com casas decimais	<i>leia_numero</i>	Altura, preço, média
Texto (String/Literal)	Sequência de caracteres (letras, números, símbolos)	<i>leia_texto</i>	Nome, endereço, frase

 **Atenção!**
Uma Curiosidade Crucial!





Atenção!



Na matemática no Brasil, usamos a vírgula para decimais (ex: 2,5). Na programação, incluindo o Potigol, usamos o **PONTO** (ex: 2.5). Usar vírgula pode causar erros!

Desafio 6 – Qual o Tipo?

Se você quisesse pedir e armazenar cada uma das informações abaixo, qual comando de leitura e qual tipo de variável seria o mais adequado?

1. O nome da sua cidade natal?
2. A sua idade em anos?
3. O preço de um litro de gasolina?
4. O seu número de telefone (com DDD)?

<i>Inteiro</i>
<i>Real (Ponto Flutuante)</i>
<i>Texto (String/Literal)</i>



Tipagem Estática: Cada Caixa com seu Tipo



No Potigol, quando você declara uma variável, ela nasce com um tipo específico e não pode mudar. Isso se chama **Tipagem Estática**.

```
1 var minha_idade := 25
2
3 minha_idade := "Vinte e cinco" // <-- ISTO NÃO FUNCIONA!
```

Analogia: É como ter um cofrinho com um buraco só para moedas (Inteiro). Você não consegue forçar a entrada de uma nota de papel (Texto) nele.

Juntando todo conhecimento



- Tripulante, agora você já conhece alguns comandos do potigol e podemos juntar eles e montar nossos primeiros programas.
- Vamos fazer um junto!

Atividades práticas





Observação

Nossa IDE utilizaremos uma nova funcionalidade, que é de interação com o programa.

Para isso, iremos colocar as respostas, caso precise, por linha.

Entrada do Programa (se houver):

```
um nome
123
14.5
olá
```



Interação com usuário

- A base de interação com usuários é ter uma pergunta e poder capturar uma resposta.
- exemplo:

```
1 escreva "Qual o seu nome?"  
2 var nome := leia_texto  
3 escreva "Olá, {nome}"
```

Qual o seu nome?
Olá, douglas

Desmembrando...



- Lembram da aula passada que falamos sobre o caso da caneta?
- Antes de pegar a caneta, tínhamos que abrir a gaveta.
- No exemplo, é do mesmo jeito!
- Antes de mostrar o nome temos que perguntar o nome.

1 escreva "Qual o seu nome?"

Desmembrando...



- Na linha 2, usamos uma **variável** chamada *nome* para podermos armazenar o valor de nome.
- Perguntamos qual o nome e guardamos a resposta.

```
2 | var nome := leia_texto
```

Desmembrando...



- Por último, agora temos que mostrar o que tínhamos guardado dentro da variável nome.
- Para mostrar, temos que usar um comando de escrever na tela e depois temos que falar com variável tenho que mostrar.
- Utilizamos as chaves {} para colocar a variável dentro.

3 escreva "Olá, {nome}"

Desafio 7

O uso das instruções primitivas é essencial no desenvolvimento de algoritmos. Raras serão as situações que permitirão construir algoritmos que não utilizem dados de entrada ou não possuam saída de dados.

Sua primeira missão, como programador, será construir um pequeno programa que irá receber 3 dados distintos:



Desafio 7

- Um valor X do tipo texto
- Um valor Y do tipo inteiro
- Um valor Z do tipo número

Esses 3 dados serão, respectivamente, um nome, uma idade e uma altura em metros.



Desafio 7

Após o seu programa receber essas informações o seu programa deverá escrever na tela no seguinte formato:

Eu sou X

Tenho Y anos

Minha altura é de Zm





- **Instruções Primitivas:** Os blocos de construção básicos (entrada, saída, atribuição).
- **Saída:** escreva (pula linha) e imprima (não pula linha).
- **Variáveis:** "Caixas" na memória para guardar dados.
- **Declaração:** var nome := valor.
- **Tipos de Dados:** Inteiro, Real e Texto.
- **Entrada:** leia_inteiro, leia_numero, leia_texto.
- **Tipagem Estática:** Em Potigol, o tipo de uma variável é fixo.

Referências



- POTIGOL é... Disponível em: <http://potigol.github.io/>. Acesso em: 25 set. 2017.
- Material de estudos do Instituto Metropole Digital