

Lógica de Programação

Aula 02 - Uga, Uga! - Instruções primitivas, variáveis e tipos de dados



Apresentação da Aula

Olá! Seja bem-vindo(a) à aula 02 da disciplina Lógica de Programação. Na aula anterior, você estudou algoritmos e linguagens de programação e viu que o algoritmo é a definição de passos para a execução de uma atividade ou resolução de problemas. Aprendeu também que a Linguagem de Programação é a ferramenta utilizada para definir esses passos.

Nessa aula, você conhecerá os primeiros comandos da linguagem Potigol para utilizá-la nos exercícios. Ela é uma linguagem moderna, frequentemente usada por alunos iniciantes. No decorrer da disciplina, você estará cada vez mais familiarizado com essa linguagem. Vamos lá?



Objetivos

- Definir instruções primitivas;
- Definir os comandos de entrada e saída;
- Definir tipos de dados e tipos de variáveis;
- Utilizar variáveis.

Instruções Primitivas



Espera aí, instruções primitivas? Será que terei de voltar ao tempo das cavernas? Não, nada disso! **As instruções primitivas são os comandos básicos existentes nas linguagens de programação.** Esses comandos permitem que o seu algoritmo possa obter ou apresentar algum dado através da Linguagem de Programação, além de armazenar esses dados em espaços de memória do computador. E, para isso, também é necessário utilizar os comandos de atribuição.

Sem as instruções primitivas, os seus programas seriam extremamente limitados, impossibilitando a construção de programas com estruturas dinâmicas. Sempre que fosse necessário realizar um procedimento, seria preciso informar os valores de entrada antes do processo de execução do programa. Desse modo, somente pessoas que possuísem conhecimento de programação conseguiriam utilizar programas.

Assim, as linguagens de programação possuem as **instruções primitivas de entrada, as instruções de saída, a declaração de variáveis e os comandos de atribuição.** Você conhecerá agora as instruções primitivas do Potigol.

Comandos de Saída

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:

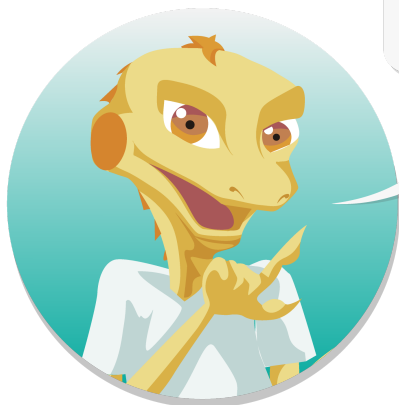
Conteúdo interativo, acesse o Material Didático.

O comando **escreva** irá imprimir, na região destinada à impressão das informações, o texto que estiver entre aspas (""). Quer experimentar o comando **escreva**? Utilize o editor de código abaixo e altere o texto entre as aspas, logo em seguida clique no botão "Executar" e avalie o resultado apresentado pelo seu simulador.

Conteúdo interativo, acesse o Material Didático.

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**. Execute o código abaixo para, então, analisar a saída do algoritmo.

Conteúdo interativo, acesse o Material Didático.



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

Esses comandos funcionam de maneira semelhante, pois imprimem na tela as informações presentes entre as aspas.

Apesar disso, possuem uma pequena diferença: um realiza a quebra de linha logo após imprimir o texto na tela, já o outro não faz essa quebra. Você já conseguiu identificar qual dos comandos realiza a quebra e qual não a realiza?

O comando **imprima** não realiza a quebra da linha após apresentar o conteúdo que estava entre suas aspas. Caso um novo comando de impressão seja utilizado, o conteúdo desse novo comando ficará "colado" na mesma linha do texto impresso pelo **imprima**. Tente agora modificar o conteúdo do programa acima e observe o que acontece.

#FicaDica

Altere a ordem dos comandos e veja como as mensagens serão impressas. Em seguida, avalie o resultado. Caso queira, você também pode modificar o conteúdo dos comandos, sintá-se à vontade! :)

Comandos de Entrada

Números Inteiros

Os comandos existentes no Potigol permitem que o algoritmo receba dados de entrada. Cada comando é responsável pela leitura de um tipo específico de dado. O primeiro comando de leitura que você irá conhecer é o **leia_inteiro**. Este permite que o algoritmo realize a leitura de números inteiros, ou seja, números que não possuem casas decimais. Números do tipo inteiro são geralmente utilizados para representar informações como: idade (18 anos), representação de datas (dia, ano ou mês) ou quantidade de algum objeto (3 livros, 2 telefones, 6 cadeiras etc.), pois nesses casos não faria sentido ter "2,5 livros", por exemplo. Veja como utilizar o comando **leia_inteiro** e observe bem o código apresentado:

Conteúdo interativo, acesse o Material Didático.

O programa visto realizou a leitura de dois números inteiros, armazenando-os em **n1** e **n2**. Em seguida, por meio da utilização do comando **escreva**, os números foram apresentados pelo seu algoritmo. Você deve tentar modificar o código acima para que ele obtenha mais dois novos números e realize a impressão destes. Além disso, que tal experimentar o comando **imprima** e verificar se ele também funciona?

Espero que você tenha compreendido o primeiro comando de entrada a partir do exemplo apresentado e das modificações que você realizou. O correto entendimento dos comandos de entrada é importante para a construção dos programas que serão apresentados no decorrer da disciplina.

Números Reais

Agora conheça o comando que permite ao seu algoritmo obter números do tipo real, ou seja, números que possuem casa decimal. Esse tipo de número é também conhecido como "ponto flutuante". Os números reais podem ser utilizados para representar informações que estejam relacionadas a valores fracionados. Por exemplo, a média de uma disciplina ou, então, o valor de algum produto ou serviço.



Atenção

Os números reais, conhecidos na programação como números de "ponto flutuante" possuem, na matemática, a vírgula (,) separando a parte inteira da parte fracionária. Já na maioria das linguagens de programação utiliza-se o ponto (.) como o separador decimal, inclusive o Potigol.

Para que você possa obter números do tipo real na linguagem Potigol é necessário utilizar o comando **leia_numero**, cujo uso é semelhante ao **leia_inteiro**, visto anteriormente. Note abaixo como ocorre a sua utilização:

Conteúdo interativo, acesse o Material Didático.

Viu como os comandos são bem parecidos? Neste momento, você deve estar se perguntando se o comando **leia_numero** é capaz de ler números sem casas decimais, certo? E a resposta é sim, ele é capaz de ler números inteiros (sem casas decimais). Altere o programa acima e faça o teste. Informe alguns números reais utilizando vírgulas (,) e outros utilizando o ponto (.) como separador e, então, observe o resultado do programa.

Mas será que o **leia_inteiro** é capaz de ler números reais (com casas decimais)? Posso lhe dizer que ele é capaz sim. Na verdade, mais ou menos... Ao se utilizar o comando **leia_inteiro** e, então, fornecer um número real, a parte fracionária do número (o valor que fica à direita da vírgula) é perdida. Quando isso ocorre, você está "truncando" o número real em um número inteiro. Que tal testar e realizar essa troca do **leia_numero** pelo **leia_inteiro**? Altere o programa acima e teste essa substituição do comando.

Dados Literais ou Cadeias de Caracteres

Além dos comandos de entrada responsáveis por obter dados do tipo numérico no Potigol, há também o comando que permite fornecer dados do tipo textual. Este, também conhecido como literal, cadeia de caracteres, ou *String*, pode ser

constituído por letras, números ou outros caracteres (como !(exclamação), @(arroba), \$(cifrão), %(símbolo de percentual), &(e comercial), *(asterisco), .(ponto), , (vírgula), entre outros).

Ou seja, nem só de números vivem os algoritmos. Para que o algoritmo construído com o Potigol leia dados do tipo literal, usa-se o comando **leia_texto**, o qual permite capturar sequências de letras e/ou palavras. O exemplo abaixo demonstra como o comando é estruturado para que sejam solicitadas duas sequências de caracteres e, então, elas sejam armazenadas em **p1** e **p2**.

Conteúdo interativo, acesse o Material Didático.

Que tal testar também o comando **leia_texto** com o comando **imprima**? Você pode modificar o exemplo acima e experimentar substituir o **escreva** pelo **imprima**.

Você viu até agora os comandos que são utilizados para que o seu algoritmo receba dados. Esses comandos são importantes e serão empregados em praticamente todos os próximos algoritmos. Caso você possua alguma dúvida a respeito do **leia_inteiro**, **leia_numero** e **leia_texto**, poste-a no fórum de dúvidas para ser auxiliado e entender melhor o conteúdo.

Definindo Variáveis

Agora que você já entendeu as instruções primitivas de entrada e saída do Potigol e verificou que com elas é possível obter ou apresentar os dados de seus algoritmos, conheça um dos elementos presentes nas linguagens de programação: **as variáveis**.

Mas o que são **as variáveis**? São porções da memória do computador reservadas para o armazenamento de algum dado dos programas. Quando se define uma variável, você está reservando um pequeno espaço na memória (geralmente, na memória [RAM](#) (A Memória de Acesso Aleatório (do inglês Random Access Memory - RAM) é utilizada como memória primária em sistemas eletrônicos digitais, pois permite tanto leitura quanto escrita de dados, diferentemente da Memória Somente de Leitura (Read Only Memory – ou ROM))).

Duas variáveis não podem possuir o mesmo endereço, pois cada variável precisa de um endereço único. Quando uma variável é declarada no programa, este, ao ser executado, realizará de modo transparente os cuidados necessários para que cada variável declarada tenha seu próprio endereço. Assim, é evitado que uma variável tome o espaço ocupado por outra variável.

Você reparou que os exemplos dos comandos de entrada **leia_inteiro**, **leia_numero** e **leia_texto** estavam sempre precedidos de algo? Se notou, parabéns! É um bom sinal de que está lendo o material didático com bastante atenção. Caso não tenha notado, deverá ficar mais atento aos códigos de exemplo, pois há sempre algo a ser observado. De todo modo, irei facilitar um pouco para você, pois ainda estamos no início dessa disciplina, não é mesmo? Nesses exemplos foram utilizados alguns identificadores, os quais receberam os nomes **n1**, **n2**, **n3**, **n4**, **p1** e **p2**. Cada identificador é uma variável diferente. A sintaxe da linguagem Potigol para declarar uma variável é formada por 4 itens:

- A palavra **var**, indicando que será declarada uma variável;
- O nome da variável;
- O sinal de atribuição (**:=**);
- O valor da variável ou o comando de leitura do tipo de dado.

Ou seja, para declarar uma variável com o Potigol você precisa apenas utilizar o comando **var**, definir o nome da variável e, então, atribuir um valor a ela, seja por um comando de entrada (**leia_inteiro**, **leia_numero** ou **leia_texto**), seja pela atribuição de um valor inicial.

Neste exemplo declarei três variáveis, respectivamente, dos tipos inteiro, real (ponto flutuante) e texto/literal.

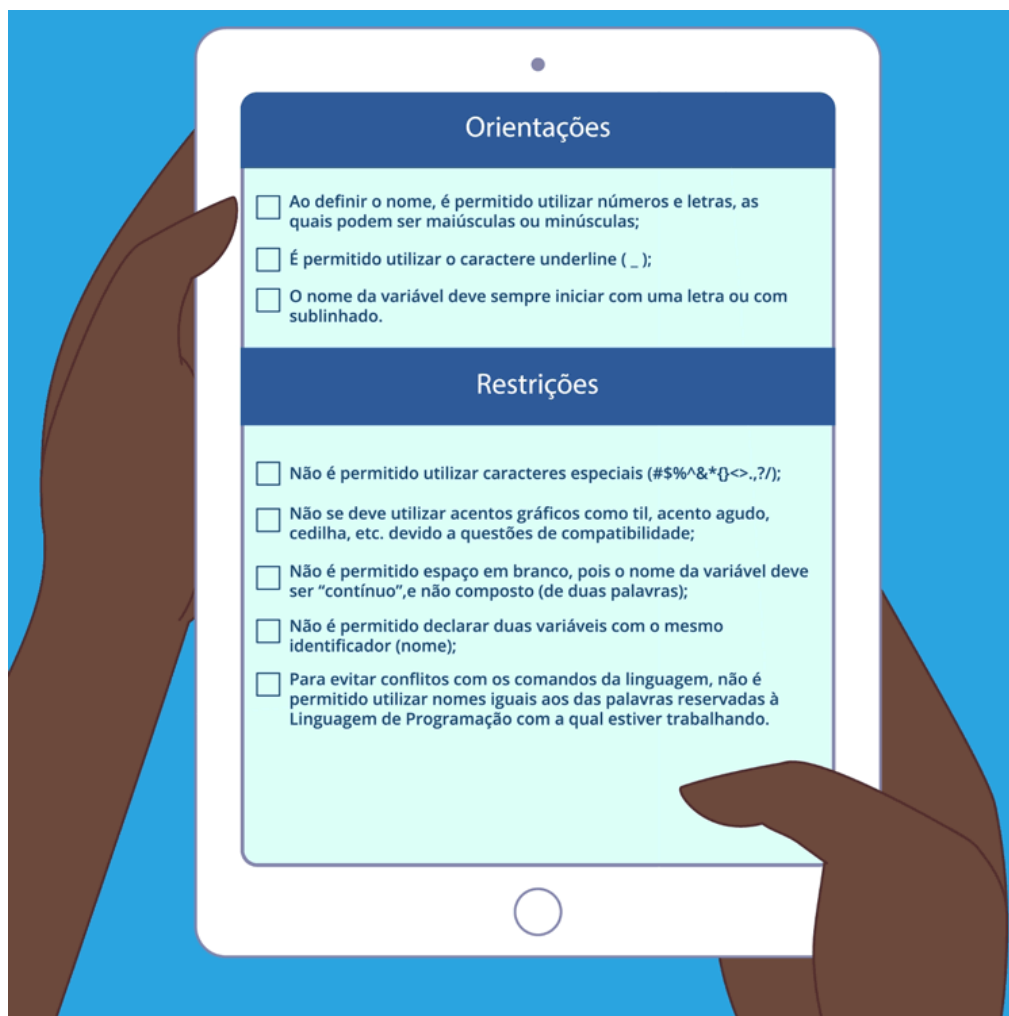
Conteúdo interativo, acesse o Material Didático.

Outra maneira de declarar uma variável é, em vez de utilizar os comandos de entrada (**leia_inteiro**, **leia_numero** ou **leia_texto**), atribuir valores iniciais para as variáveis. O exemplo a seguir declara as variáveis de tipos inteiro, real e texto, respectivamente.

Conteúdo interativo, acesse o Material Didático.

Percebeu como foi fácil declarar as primeiras variáveis? Agora que você já viu o essencial das variáveis, observe algumas orientações e restrições para definição do nome delas. Elas são válidas para a maioria das linguagens de programação existentes.

Figura 01 - Checklist



Tipos de Variáveis

Acredito que você já percebeu a existência de vários tipos de dados, para os quais, de acordo com o tipo, é necessário utilizar um comando de entrada específico. Você também notou que, para declarar uma variável, é necessário atribuir um dado a ela, através de um comando de entrada ou da atribuição direta de um valor.

Quando uma variável é declarada no Potigol, ela torna-se de um tipo específico. Por esse motivo, só é possível atribuir a ela os dados com os quais o seu o tipo for compatível. Por exemplo, você não pode atribuir a uma variável do tipo inteiro um dado do tipo textual. Veja o exemplo:

```
1 var minha_variavel := 2017
2 minha_variavel := leia_texto
3
```

O código acima não funcionará, pois **minha_variavel** foi declarada, inicialmente, através da atribuição de um número do tipo inteiro. Por esse motivo, será possível armazenar apenas números inteiros na variável **minha_variavel**. A esse comportamento damos o nome de “Tipagem Estática”.

- **Tipagem Estática:** Uma vez definido o tipo de uma variável, não é possível modificá-lo no decorrer do programa.
- **Tipagem Dinâmica:** É possível modificar o tipo da variável a partir da atribuição de um novo tipo de dado, portanto, a qualquer momento da execução do programa.



Atividade 01

Conteúdo interativo, acesse o Material Didático.



Resumo

Nesta aula você conheceu os conceitos relativos às instruções primitivas, compreendeu a importância da definição desses comandos e a maneira como declarar variáveis. O comando de entrada é utilizado para receber dados digitados pelo usuário. Já o comando de saída é utilizado para mostrar dados na tela. Além disso, você aprendeu os tipos de dados (inteiro, real e texto/cadeia de caracteres) e observou que, quando falamos sobre variável, nos referimos a uma "região" de memória do computador, a qual tem por objetivo armazenar algum tipo de dado. Ao final, também viu que, para definir o nome de identificação de uma variável, são necessárias algumas restrições e orientações.

Até a próxima aula!



Referências

POTIGOL é... Disponível em: <<http://potigol.github.io/>>. Acesso em: 25 set. 2017.