

PERSPECTIVAS

Programando o Futuro

Os robôs estão fazendo o seu trabalho de forma adequada?

Desafio de programação - Python



Uma grande empresa de vendas online deu início a um novo projeto de expansão com o objetivo de melhorar o trabalho realizado para entrega de produtos. Se o projeto tiver sucesso, a empresa conseguirá maximizar a produtividade dos armazéns dos centros de distribuição, proporcionando entregas mais eficientes e atendendo melhor os seus clientes.

A primeira fase do projeto consiste em inserir os chamados “veículos guiados automaticamente”, ou AGVs, como são conhecidos pelas pessoas que trabalham com esse tipo de tecnologia. Esses veículos podem ser considerados um tipo de **robô de serviço** e são usados em substituição a outros equipamentos que necessitam de operação manual, como as empilhadeiras. Quando os AGVs são colocados em operação, eles podem auxiliar os trabalhadores e as trabalhadoras da empresa no transporte de materiais dentro dos grandes armazéns.

Os robôs escolhidos pela empresa são capazes de navegar pelas instalações do armazém seguindo rotas pré-estabelecidas que são marcadas por tiras magnéticas coladas ao chão. Essas tiras magnéticas são percebidas por sensores acoplados aos robôs e juntos, tiras e sensores, habilitam os robôs a navegarem de forma autônoma (sem necessidade de serem controlados, por exemplo, por controles remotos). Veja exemplos na figura 1.

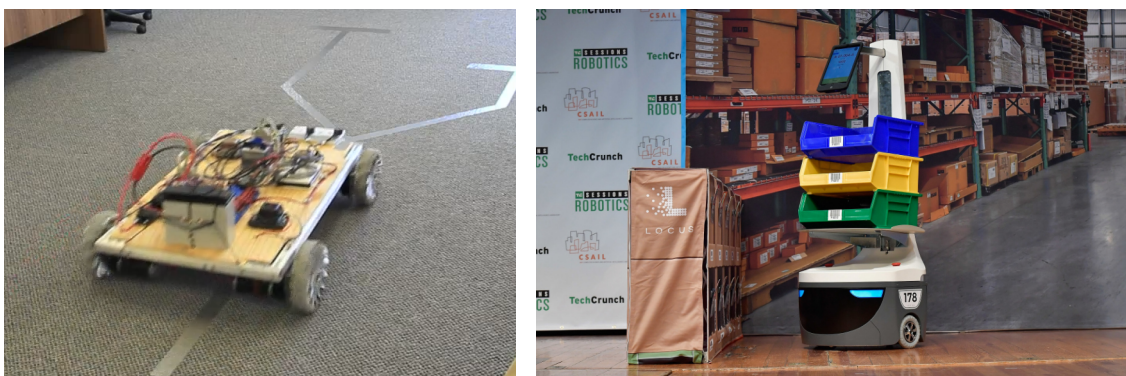


Figura 1: (direita) veículo guiado automaticamente navegando em uma rota; (esquerda) robô LocusBot da empresa Locus Robotics

Imagens de:

- <https://www.flickr.com/photos/scsibug/5423797749/>
- <https://www.flickr.com/photos/techcrunch/35155061164/>



*Há outras formas de implementar a navegação de robôs! **Continue estudando com a equipe do projeto Perspectivas e você vai aprender mais sobre todo esse universo.***

Quando um robô é colocado em operação para realizar um serviço ele deve ser usado em benefício tanto da empresa quanto das pessoas que trabalham na empresa. Mas, em alguns casos, ele fará o trabalho no lugar de uma pessoa, que precisará aprender uma nova função.

Aprender a trabalhar com os robôs e com outros aparelhos similares é uma boa opção, por isso é importante aumentar os nossos conhecimentos sobre essas tecnologias e estudar como elas funcionam.

Nesta fase inicial do projeto, os robôs ainda estão em teste e é necessário avaliar o desempenho deles em relação ao que eles estão fazendo dentro dos armazéns. Nessa fase de inicial, os robôs não são ainda capazes de enviar informações sobre o seu trabalho para uma central de processamento de dados, então é necessário que algumas pessoas observem os robôs trabalhando e registrem em um sistema alguns dados sobre o desempenho dos robôs em cada um dos setores do armazém.

Mas esse sistema não está pronto e como toda a equipe de trabalho responsável pelo projeto de implantação dos robôs está trabalhando duro na adaptação das instalações do armazém, o gerente responsável pelo projeto pediu a ajuda da equipe do projeto PeRSpectivAs. Você, como parte desta equipe, está alocada (ou alocado) para desenvolver um programa para ser usado na entrada de dados e para geração de um relatório.



Observe na figura 3, o exemplo de relatório que o gerente gostaria de obter a partir dos dados fornecidos pelas pessoas que estão observando o trabalho dos robôs:

RELATÓRIO DE DESEMPENHO AGV

Código do robô:
158

Data (dd/mm/aaaa):
11/10/2021

Setor: 1. Recebimento () 2. Armazenamento () 3. Separação () 4. Expedição ()
1

Desempenho: 1. Satisfatório () 2. Insatisfatório ()
2

Resumo do desempenho:
O robô não está conseguindo navegar corretamente sobre a tira magnética. A tira magnética foi colocada de forma que curvas com ângulos muito pequenos foram criadas. O robô não é capaz de fazer curvas tão fechadas.

Figura 3: Relatório fornecido pelo gerente do projeto

A sua tarefa é construir um programa que transforme o exemplo de relatório fornecido pelo gerente do projeto em um interface modo texto por meio da qual os funcionários da empresa que estão observando os robôs possam digitar os dados que preencherão o relatório. Esses dados ficarão armazenados na memória RAM do computador.



Revisando o conceito de memória RAM: A memória RAM (Random Access Memory ou Memória de Acesso Aleatório) é um componente de hardware presente em computadores. Nesse componente, o sistema operacional e os programas (por exemplo, os aplicativos de um smartphone) podem armazenar os dados que utilizam durante o seu funcionamento.

*Então, os dados fornecidos como entrada para o programa não estarão mais armazenados quando o programa terminar sua execução. **Continue estudando com a equipe do projeto Perspectivas** e você ainda vai aprender como fazer para armazenar os dados de forma permanente no computador usando Python.*

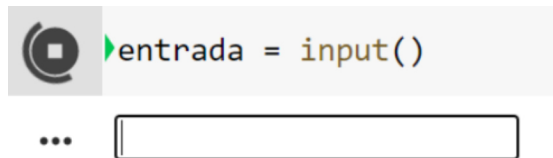
A linguagem Python possui recursos específicos para construir programas que resolvam esse tipo de problema. Isso tornará o seu trabalho muito mais fácil!!!! UFA!!!! :-)

A grande questão aqui é: como tornar possível que uma pessoa (o usuário de nosso programa) digite um dado e o valor desse dado seja atribuído a uma variável?

O comando ***input()*** serve justamente para esse propósito. Esse comando indica ao programa que algum valor será informado. Ao ser executado, ele cria uma “caixa de entrada” onde o usuário pode inserir um valor que será recebido pelo programa. É um comando de entrada de dados.

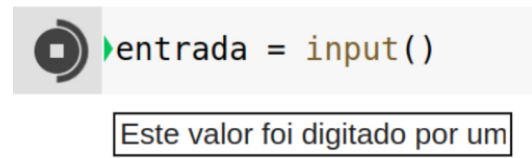


Observe os exemplos nas figuras 4 e 5:



```
entrada = input()
```

Figura 4: É executada a célula com o comando `input()`

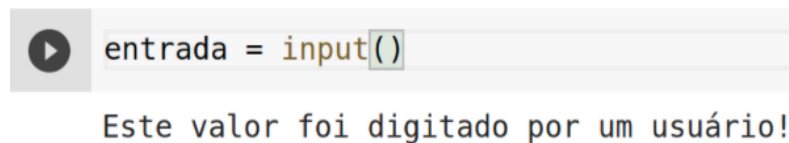


```
entrada = input()
```

Este valor foi digitado por um

Figura 5: O usuário insere uma entrada

Ao executar a célula de código (figura 6), o programa cria uma caixa na qual o usuário ou a usuária deve digitar o dado que ele quer armazenar na variável chamada “entrada”, e então ao pressionar a tecla Enter. Ao realizar esses passos, a execução do programa continua seguindo para os próximos comandos, se eles existirem.

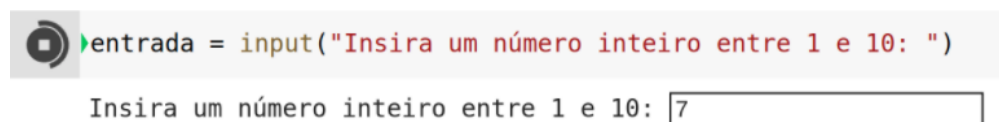


```
entrada = input()
```

Este valor foi digitado por um usuário!

Figura 6: O Google Colab exibe o conteúdo fornecido pelo usuário

O comando `input()` também aceita um parâmetro opcional bastante útil. Nós podemos inserir uma string dentro do parênteses que será exibida antes da caixa na qual algum valor será inserido. Isso é especialmente útil quando desejamos comunicar à usuária ou usuário o que é esperado que ela ou ele forneça no programa. Veja a figura 7:



```
entrada = input("Insira um número inteiro entre 1 e 10: ")
```

Insira um número inteiro entre 1 e 10: 7

Figura 7: Ao utilizar o parâmetro adicional é exibida uma mensagem ao usuário



Um detalhe importante do comando `input()` é que ele assume **por padrão** que o dado inserido na caixa é um valor do tipo string. Isso pode ser um problema quando desejamos receber valores de outros tipos, como inteiro ou de ponto flutuante.

Para ajustar o comportamento do comando, podemos utilizar os comandos de **conversão de tipo**, como `int()`, `str()` ou `float()`. Esses comandos transformam o tipo de dado do valor inserido no tipo de dado desejado para uso no programa.

Vamos usar os comandos `print()` e `type()` para assimilar melhor esses conceitos novos. Observe a figura 8.

```
▶ entrada = input("Insira um número inteiro: ")
print(type(entrada))

entrada_int = int(entrada)
print(type(entrada_int))

Insira um número inteiro: 2
<class 'str'>
<class 'int'>
```

Figura 8: Utilização do comando `type()`

Você deve se lembrar do comando `print()`. Ele foi um dos primeiros comandos que aprendemos nessa nossa jornada de aprendizado de programação. A função dele é exibir na tela o conteúdo que é passado a ele como parâmetro.

Esse conteúdo pode ser um valor explícito, como uma string, ou pode ser uma variável que contenha um valor. O conceito de variável também já deve ter sido estudado.



Você se lembra do conceito de “tipo de variável” e o que faz o comando `type()`?

Os tipos de dados são importantes, pois cada um deles possui características específicas que:

- definem o tamanho do espaço que deve ser reservado a ele na memória. Também dizemos que se trata da quantidade de memória necessária para armazená-lo.
- define a forma como eles podem ser utilizados.

O comando `type()` permite verificar o tipo do dado que está armazenado na variável.

Com esses novos conceitos que você aprendeu, você já consegue fazer a primeira parte do seu programa. Nessa parte você deve criar as condições para a entrada de dados sobre o desempenho de um dos robôs.

Agora que você já fez a primeira parte, nós vamos mostrar para você como o relatório com os dados pode ser exibido na tela, usando uma interface em modo texto.

Veja o exemplo da figura 9 para aprender como fazer essa exibição de dados.

```
▶ nome = input("Olá! qual o seu nome? ")
sobrenome = input("Olá "+nome+"! Qual é o seu sobrenome? ")
idade = input("Olá "+nome+" "+sobrenome+"! Qual é a sua idade? ")

print(nome+" "+sobrenome+" tem "+idade+" anos!")

Olá! qual o seu nome? João
Olá João! Qual é o seu sobrenome? Paiva
Olá João Paiva! Qual é a sua idade? 20
João Paiva tem 20 anos!
```

Figura 9: Exibindo as informações inseridas pelo usuário



Olhe com um pouco mais de detalhes para essa linha do programa para entender melhor o que foi programado:

```
sobrenome = input("Olá "+nome+"! Qual é o seu sobrenome ?")
```

Nessa linha de programação estamos solicitando o dado referente ao sobrenome da pessoa que está fornecendo dados. Por isso, nomeamos a variável que receberá a entrada fornecida na execução do comando *input()* como “sobrenome”.

Mas o parâmetro que foi passado para o comando *input* traz uma novidade. Para construir o parâmetros, nós usamos o operador “+” que concatena *strings*. Na realidade nós temos três *strings* diferentes.

Concatenação é um termo usado em computação para designar a operação de unir o conteúdo de duas strings.

Por exemplo, considerando as strings "casa" e "mento" a concatenação da primeira com a segunda gera a string "casamento".

```
"Olá " + nome + "! Qual é o seu sobrenome ?"
```

Essas strings são concatenadas em uma só para formar a string final que o comando *input* exhibe na tela antes da caixa de entrada de dados. Uma de nossas três *strings* está armazenada na variável **nome**. Esse armazenamento é resultado da linha anterior no programa.

Agora é a sua vez de criar um programa para exibição do relatório que o gerente do projeto solicitou!

