

Missão Prática | Nível 1 | Mundo 1

Material de **orientações** para desenvolvimento da **missão prática** do **1º nível de conhecimento**.

📌 **As práticas devem ser feitas individualmente.**

RPG0001 - Desvendando a Caixa Preta

Objetivos da prática

- Descrever a instalação e utilização do interpretador Python;
- Descrever a construção e execução da primeira instrução do Python;
- Descrever a configuração do ambiente local para desenvolvimento com Python;
- Descrever a declaração e utilização de variáveis com Python;
- Descrever a utilização de dados através de entrada do usuário via teclado com Python;

Entrega e Progresso

- As microatividades irão dar suporte para o desenvolvimento da Missão Prática. Elas têm apoio/gabarito para resolução no próprio documento;
- A entrega esperada é a Missão Prática, descrita neste documento após as Microatividades;
- A missão prática progride 5% na entrega e até 5% dependendo da nota atribuída pelo tutor em sua correção.

👉 Atividades práticas

Desvendando a Caixa Preta

Microatividade 1: Descrever a instalação e utilização do interpretador Python

- Material necessário para a prática

Privilegios de instalação de software no Sistema Operacional utilizado no PC/Notebook

- Procedimentos

1. Sendo um usuário do Sistema Operacional Windows:
 - a. Acesse o aplicativo Microsoft Store;
 - b. Pesquise por Python;

- d. Busque por Python;
 - e. Na lista de resultados selecione e instale a última versão disponível do "Python interpreter and runtime";
 - f. Após finalizada a instalação, abra uma nova janela do CMD (prompt de comando) ou do Windows PowerShell;
 - g. Na janela aberta, digite o comando: `python --version`
 - h. Como resultado do comando acima você deverá ver exibida a versão instalada do interpretador Python.
2. Sendo um usuário do Sistema Operacional Linux (ou MacOS):
 - a. Verifique, estando na janela do terminal, se o interpretador Python já está instalado com o seguinte comando:
 - i. `python --version`
 - b. Caso não visualize, ao executar o comando acima, a versão atual instalada, siga os passos a seguir;
 - c. Caso utilize uma distribuição Linux baseada em Debian e Ubuntu, use o gerenciador de pacotes para instalar o interpretador Python:
 - i. `sudo apt-get install python`
 - d. Caso utilize uma distribuição Linux baseada em RedHat e CentOS, use o gerenciador de pacotes para instalar o interpretador Python:
 - i. `sudo yum install python`
3. Sendo um usuário do Sistema Operacional Mac OS:
 - a. Abra um terminal e execute, em sequência, os comandos abaixo:

xcode-select --install

sudo easy_install pip

sudo pip install --upgrade pip

ruby -e "\$(curl -fsSL https://raw.githubusercontent.com/mxcl/homebrew/go)"

brew install python

- Resultados esperados ✨

O resultado esperado dessa microatividade é verificar se o aluno possui conhecimentos de fundamentos de hardware e software, sendo capaz de instalar aplicações através de gerenciadores de pacotes gráficos ou textuais.

Microatividade 2: Descrever a construção e execução da primeira instrução do Python

- Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;

- Procedimentos:

1. Abra uma janela do terminal e digite o comando: `python`
2. O comando anterior abrirá o interpretador Python, permitindo que comandos da linguagem sejam executados diretamente na janela do terminal. Nesse ponto você deverá ver em seu terminal o prompt, conforme mostrado a seguir (lembrando que as informações da imagem podem ser ligeiramente diferentes, dependendo do seu Sistema Operacional):
3. No prompt (ao lado do `>>>`), digite a instrução:


```
print("Hello Python Code")
```
4. Como saída para a instrução acima, você deverá ver, logo abaixo do comando, ainda no terminal, o texto "Hello Python Code";
5. Agora, execute essa nova instrução no prompt:

```
print(1 + 1)
```

6. A saída do comando acima deverá ser o número 2 (uma vez que a instrução indicava que deveria ser impresso o resultado da soma entre dois números;
7. A exemplo desse último comando, teste outros combinando o comando "print" e operações matemáticas.

- Resultados esperados ✨

O resultado esperado dessa microatividade é permitir ao aluno ter um contato inicial com o interpretador de códigos Python, assim como com alguns comandos básicos da linguagem.

Microatividade 3: Descrever a configuração do ambiente local para desenvolvimento com Python

- Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

- Procedimentos

1. Abra a IDE Vs Code;
2. Clique no menu/opção "Extensions";
3. Na barra de pesquisa, procure por "Python";
4. Na lista de resultados, selecione a opção "Python" do fornecedor/desenvolvedor Microsoft;
5. Na página da extensão, clique em "Install";
6. Após concluir a instalação, feche a página da extensão;
7. Navegue, no Vs Code, até a opção "Explorer";
8. Clique em "Open Folder";
9. Selecione uma pasta, no seu computador, para utilizar como local de armazenamento dos seus códigos Python. Você também poderá criar uma nova pasta nesse momento e a selecionar posteriormente;
10. Ao ser questionado se confia no autor da pasta e seus arquivos ("Do you trust the authors of the files in this folder"), clique em "Yes..";
11. Na janela "Explorer", onde a pasta escolhida estará sendo exibida, crie um novo arquivo nomeando-o como "teste.py";
12. Digite, como conteúdo desse novo arquivo/script, o seguinte conteúdo:

```
print("Hello World")
```

13. Salve o arquivo/script;
14. Na barra superior direita você verá um ícone no formato de um triângulo deitado, semelhante à imagem abaixo:

15. Clique no ícone. A seguir, deverá ser aberta uma janela, na parte inferior do VS Code, exibindo o terminal e a saída do comando acima, que é a exibição da frase digitada após o comando "print";
16. Caso ocorra algum erro na execução do comando, verifique se digitou o comando no script corretamente, ou seja, configurou apropriadamente o interpretador Python. Nessas situações, a própria mensagem de erro exibida fornece uma pista para a sua solução;

17. Por fim, você poderá criar novas linhas no seu script, inserindo os mesmos comandos que utilizou na microatividade anterior. A diferença é que, usando o VS Code, você poderá declarar todas as suas instruções em sequência (uma em cada linha) e executá-las ao mesmo tempo.

- Resultados esperados ✨

O resultado esperado dessa microatividade é fornecer ao aluno uma visão geral de como criar e executar scripts Python em seu ambiente local de desenvolvimento.

Microatividade 4: Descrever a declaração e utilização de variáveis com Python

- Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

- Procedimentos

1. Abra a IDE VS Code;
2. Na mesma pasta onde criou o script utilizado na microatividade anterior, crie um novo script chamado "variaveis.py";
3. Com o script aberto no VS Code, digite os comandos a seguir:

```
a = 15
```

```
b = 20
```

```
soma = a + b
```

```
print("Resultado da soma: " + str(soma))
```

4. Salve o script e o execute (através do atalho na IDE mostrado na atividade anterior).

Alguns pontos de atenção sobre o código acima:

- Através dos comandos acima você declarou 3 variáveis: a, b e soma. Além disso, atribuiu valores a cada uma delas. Por fim, você utilizou a instrução print para concatenar uma string e uma variável (soma) e exibir a combinação de ambas na tela;
- Repare que a variável soma, na instrução print, é precedida pela palavra str. Isso é necessário para converter a variável soma, que é, implicitamente, do tipo inteiro numa variável do tipo string. Experimente modificar a instrução print para que fique dessa forma:

```
print("Resultado da soma: " + soma)
```

- Perceba que, nesse caso, é exibido um erro dizendo que só é possível concatenar string (str) com strin e não com inteiro (int).
- Estando ainda com o script aberto, insira novas variáveis e realize outras operações matemáticas e também combinações de variáveis, de diferentes tipos, a fim de melhor entender o comportamento de tal recurso na linguagem Python.

- Resultados Esperados ✨

O resultado esperado dessa microatividade é fornecer ao aluno uma visão prática sobre variáveis, e seu uso, na linguagem Python.

Microatividade 5: Descrever a utilização de dados

através de entrada do usuário via teclado com Python

- Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

- Procedimentos

1. Abra a IDE VS Code;
2. Na mesma pasta onde criou os scripts utilizados nas microatividades anteriores, crie um novo script chamado "entrada.py";
3. Com o script aberto no VS Code, digite os comandos a seguir:

```
name = input('Digite seu nome:')
```

```
idade = input('Digite sua idade:')
```

```
print('Olá, ' + name + '. Você tem ' + idade + ' anos.')
```

4. Salve o script e o execute (através do atalho na IDE mostrado na atividade anterior);
5. No prompt do terminal, clique ao lado da frase 'Digite seu nome:' e insira, via teclado, tal informação. Tecle 'enter' ao final e repita esse processo na frase seguinte. Ao final desse processo você deverá ver no terminal o conteúdo da instrução print, onde as variáveis name e idade exibirão os valores inseridos por você.

Alguns pontos de atenção sobre o código acima:

- Nos comandos acima vimos novos exemplos de utilização de variáveis. Nesse caso, tal recurso da linguagem foi usado em conjunto com outro recurso, o de entrada de dados via teclado.
- Como na atividade anterior, treine seus conhecimentos acrescentando novas instruções no script e o executando a seguir.

- Resultados esperados ✨

O resultado esperado dessa microatividade é demonstrar ao aluno como combinar a utilização de variáveis com dados inseridos via interação com o usuário na linguagem Python.

Missão Prática | Desvendando a Caixa Preta

Através dessa atividade o aluno codificará uma calculadora utilizando os recursos da linguagem Python aplicados ao longo das microatividades.

Contextualização

Você foi contratado para desenvolver uma aplicação utilizando a linguagem Python. Tal aplicação será composta por uma calculadora que deverá permitir a execução das quatro operações matemáticas (adição, subtração, multiplicação e divisão) utilizando valores digitados pelos usuários. Os requisitos dessa aplicação são descritos a seguir, no Roteiro de Prática.

Roteiro de prática 📋

- Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

- Procedimentos

1. Abra a IDE VS Code;
2. Na mesma pasta onde criou os scripts utilizados nas microatividades, crie um script chamado "calculadora.py";
3. No script, insira:
 - a. Crie duas variáveis para receber e armazenar os valores "imputados" via prompt pelo usuário;
 - b. Imprima na tela, em sequência, o resultado das quatro operações matemáticas aplicadas sobre as duas variáveis;
 - c. Além dos resultados em si, insira, antes, uma frase informando a qual operação matemática se referem. Ex: "Resultado da adição: ".
4. Salve as alterações no script e o execute via VS Code;
5. Teste o aplicativo interagindo com ele através do prompt, fornecendo os dados necessários para a sua execução.

- Resultados esperados ✨

O resultado esperado dessa atividade é verificar se o aluno possui conhecimentos da linguagem Python que o possibilitem declarar e utilizar variáveis, assim como permitir a interação com usuários através da entrada de dados via teclado.

📌 Referências

Não foram utilizadas referências bibliográficas para a elaboração das atividades.

Entrega da prática

Chegou a hora, gamer!

- 📁 Armazene o projeto em um repositório no GIT.
- 📎 Anexar a documentação do projeto (PDF) no GIT.
- 📢 Compartilhe o link do repositório do GIT com o seu tutor para correção da prática, por meio da **Sala de Aula Virtual**, na aba "**Trabalhos**" do respectivo nível de conhecimento.
- 📅 Ei, não se esqueça de entregar este trabalho na data estipulada no calendário acadêmico!

Made with Microsoft Sway

Create and share interactive reports, presentations,
personal stories, and more.

Get started

