

Missão Prática | Nível 5 | Mundo 1

Material de **orientações** para desenvolvimento da **missão prática** do **5º nível de conhecimento**.

📌 Conforme a metodologia gamificada propõe, a dificuldade das missões aumenta a cada nível, então o Nível 5 é o mais complexo do Mundo atual. Para melhorar a experiência do aluno, orientamos a desenvolver esse nível envolvendo empresas parceiras, colegas ou interações externas, para desenvolver o contato externo e praticar a presencialidade em suas atividades.

RPG0005 - Colocando tudo em ordem e guardando

Objetivos da prática

- Descrever a ordenação de um array utilizando Python;
- Descrever a utilização do algoritmo de ordenação “Bubble Sort” em Python;
- Descrever a utilização do algoritmo de ordenação “Selection Sort” em Python;
- Descrever a leitura de dados a partir de um arquivo externo em Python;
- Descrever a escrita de dados em um arquivo externo em 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

Colocando tudo em ordem e guardando

Microatividade 1: Descrever a ordenação de um array utilizando 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. No menu File, selecione a opção “Open Folder”;
3. Crie uma nova pasta em seu computador para armazenar os códigos desse conjunto de microatividades e a selecione a partir do VS Code;
4. No VS Code, menu Explorer, clique na pasta do projeto e crie um novo arquivo/script chamado “array.sort.py”;
5. No script criado:
 - a. Crie um array de 15 posições com números inteiros e valores aleatórios, não ordenados;
 - b. Realize a ordenação dos dados do array utilizando o método “sort”;
 - c. Imprima, na tela, o conteúdo do array (e confira se os dados foram devidamente ordenados, de forma crescente);
 - d. Utilizando novamente o método “sort”, mas passando os parâmetros “key=None, reverse=True”, ordene, de forma decrescente, os valores constantes no array;
 - e. Imprima, na tela, o array (e confira se os dados estão ordenados conforme planejado);
 - f. Crie um array de strings e repita os passos acima.
 - i. nome;
 - ii. dataNascimento;
 - iii. cpf;
 - iv. rg
6. Salve o arquivo/script;
7. Passos para execução do script:
 - a. Na barra superior direita você verá um ícone no formato de um triângulo deitado, semelhante à imagem abaixo:

- b. Clique no ícone. A seguir, deverá ser aberta uma janela, na parte inferior

do VS Code, exibindo o terminal e a saída dos comandos acima;

c. Caso ocorra algum erro na execução do script, verifique o seu código, corrigindo eventuais erros, e tente executá-lo novamente.

- Resultados esperados 🌟

O resultado esperado dessa microatividade é demonstrar ao aluno como ordenar os valores de um array usando um método próprio da linguagem Python.

Microatividade 2: Descrever a utilização do algoritmo de ordenação “Buble Sort” em Python

- Contextualização

O algoritmo Bubble Sort é um dos algoritmos de ordenação mais simples para se implementar. Além disso, tem performance considerada razoável, devendo ser usado em arrays de tamanho pequeno. Sua lógica consiste em varrer os valores do array comparando os valores adjacentes e trocando-os de lugar a partir da ordenação do menor para o maior valor.

- 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 utilizada na microatividade anterior, crie um novo arquivo/script chamado “bubble.sort.py”;
3. Nesse novo script:
 - a. Crie um método chamado bubbleSort que receberá, como parâmetro, o array a ser ordenado;
 - b. No escopo do método, crie um primeiro laço for que deverá fazer a iteração nos elementos do array:

for i in range(len(array))

- c. Dentro do for acima, crie outro laço for para iterar , de dois a dois, os elementos do array:

`for j in range(0, len(array) - i - 1)`

d. Aninhado ao segundo for, utilize uma condição “if” e compare os valores dos dois elementos adjacentes do array. Caso o elemento “j” seja maior que o elemento “j+1”:

- i. Crie uma variável e atribua a ela o valor de “array[j]”
- ii. Substitua o valor de “array[j]” por “array[j+1]”;
- iii. Substitua o valor de “array[j+1]” pela variável criada acima.

e. Fora do método, declare um array de números com 15 posições;

f. Aplique o método “bubbleSort”, criado acima, no array;

g. Imprima, na tela, o array.

4. Salve o arquivo/script e o execute. Você deverá ver, no prompt, os valores do array ordenados de forma crescente.

- Resultados esperados ✨

O resultado esperado dessa microatividade é verificar os conhecimentos do aluno relacionados a algoritmos de ordenação e permitir ao mesmo aplicar, de forma prática, o algoritmo Bubble Sort.

Microatividade 3: Descrever a utilização do algoritmo de ordenação “Selection Sort” em Python

- Contextualização

O algoritmo Selection Sort também está entre os algoritmos de ordenação de mais fácil implementação. Sua lógica consiste em iterar sobre os valores do array, identificar o menor elemento e o posicionar no lugar do primeiro elemento do array. Após isso, ele faz o mesmo com o segundo menor elemento, trocando-o de lugar com o segundo elemento e assim sucessivamente, até ordenar todo o array.

- 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 utilizada na microatividade anterior, crie um novo arquivo/script chamado "selection.sort.py";
3. Nesse novo script:
 - a. Crie um array e o popule com 15 números inteiros, sem nenhuma ordenação;
 - b. Crie um primeiro laço for que deverá fazer a iteração nos elementos do array:

for i in range(len(array))

- c. Dentro do primeiro laço for, crie uma variável que receba a variável "i" (definida na assinatura do laço for);
- d. Ainda dentro do primeiro laço, crie outro laço para iterar no range entre a posição "i + 1" e o tamanho total do array:

for j in range(i+1, len(Array))

- e. Dentro do for acima, utilize uma condição "if" para verificar se o valor do elemento do array, na posição da variável criada no item "c", é maior que o valor do elemento do array na posição "j" (variável definida na assinatura do segundo laço for);
 - f. Em caso positivo, atribua à variável criada no item "c" o valor de "j";
 - g. Dentro do primeiro laço for, proceda com a troca dos valores do array:
 - i. Array na posição "i" deverá receber o valor do array na posição da variável criada no item "c";
 - ii. Array na posição da variável criada no item "c" deverá receber o valor do array na posição "i";
 - h. Imprima, na tela, o array.
4. Salve o arquivo/script e o execute. Você deverá ver, no prompt, os valores do array ordenados de forma crescente.

- Resultados esperados 🌟

O resultado esperado dessa microatividade é verificar os conhecimentos do aluno relacionados a algoritmos de ordenação e permitir ao mesmo aplicar, de forma prática, o algoritmo Selection Sort.

Microatividade 4: Descrever a leitura de da

a partir de um arquivo externo em 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 utilizada na microatividade anterior, crie um novo arquivo/script chamado “ler.txt.py” e um arquivo com extensão “txt” chamado “loremipsum.txt”. No txt insira as seguintes linhas:

Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi
ut aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia
deserunt mollit anim id est laborum.

3. No script “ler.txt.py”:
 - a. Crie uma variável e atribua a ela, com o método “open”, o conteúdo do arquivo txt criado acima;
 - b. Imprima, na tela, todo o conteúdo da variável (e, consequentemente, do arquivo txt lido);
 - c. Imprima, na tela, apenas a primeira linha do arquivo txt;
 - d. Imprima, na tela, apenas os 3 primeiros caracteres do arquivo txt;
 - e. Agora, utilize a instrução “with” para abrir o arquivo text, armazenar seu conteúdo numa variável e o imprimir, na tela, na sequência.
4. Salve o arquivo/script e o execute. Você deverá ver, no prompt, o conteúdo que mandou imprimir com o código acima.

- Resultados Esperados 🌟

O resultado esperado dessa microatividade é verificar se o aluno possui os conhecimentos básicos de leitura de dados em arquivos utilizando Python.

Microatividade 5: Descrever a escrita de dados em um arquivo externo em 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 utilizada na microatividade anterior, crie um novo arquivo/script chamado “escrever.txt.py”. Nesse script:
 - a. Abra um arquivo de texto, ainda não existente, chamado “texto.txt” e o atribua a uma variável;
 - b. Crie uma lista usando a sintaxe: `texto = list()`;
 - c. Utilizando o método “append”, atribua algumas frases para a lista;
 - d. Escreva, no arquivo aberto, o conteúdo da lista.
3. Salve o arquivo/script e o execute. Você deverá ver, na pasta do projeto, no VS Code, o arquivo criado através do código acima. Abra o arquivo e confira, também, se o seu conteúdo corresponde ao que você codificou.

- Resultados esperados ✨

O resultado esperado dessa microatividade é verificar se o aluno possui conhecimentos sobre a manipulação de dados em arquivos utilizando Python.

Missão Prática | Colocando tudo em ordem guardando

Após aplicar, de forma prática, os conceitos básicos de Programação Orientada a Objetos usando Python, chegou a hora de refatorar um código, inicialmente escrito, em outras microatividades, com uso do paradigma de Programação Estruturada.

Contextualização

Você foi recém-contratado como Analista de Dados Python por uma empresa

Você foi recém-contratado como Analista de Dados Python por uma empresa especialista em mineração de dados. Como primeiro projeto, você recebeu a incumbência de criar um glossário de palavras ordenadas que, posteriormente, se transformará num dicionário de termos jurídicos. Em suma, seu trabalho consistirá na leitura de alguns documentos, no formato txt, na “quebra” do conteúdo desses documentos por palavras e, por fim, na ordenação de todas essas palavras. A seguir você poderá acompanhar os procedimentos para criação do primeiro MVP.

Roteiro de prática

- Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;
- Arquivo em formato “txt” contendo um texto qualquer;

- Procedimentos

1. Abra a IDE VS Code;
2. Na mesma pasta onde criou os scripts utilizados nas microatividades, crie um novo script chamados “kdd.py”;
3. No script:
 - a. Crie uma variável do tipo lista (use a classe list());
 - b. Com a instrução with, leia o conteúdo do arquivo “txt”, percorrendo linha-a-linha;
 - c. Separe cada linha lida em palavras. Para isso, você poderá utilizar o método “split”;
 - d. Atribua, à lista criada, cada uma das palavras lidas;
 - e. A seguir, você deverá ordenar as palavras armazenadas na lista. Para isso, visando comparar a eficiência/performance, utilize os algoritmos de ordenação “Bubble Sort” e “Selection Sort”. Além deles, use também o método nativo de ordenação de listas no Python, o “sort”.
 - f. Imprima, na tela, o resultado da ordenação obtido por cada método acima. Nesse ponto, você deverá também imprimir o tempo de execução de cada método, a fim de comparar os resultados de cada um e poder optar, posteriormente, pelo de melhor performance;
 - g. Para imprimir o tempo de execução, você poderá, por exemplo, usar a biblioteca time, calculando a diferença entre o momento final e o momento de início de execução do método de ordenação;
 - h. Salve as alterações realizadas e execute o script repetidamente, para cada método de ordenação definido;
 - i. Após medir a performance e escolher um dos algoritmos/métodos, remova os demais do script e crie um novo arquivo txt contendo as palavras ordenadas.

- Resultados esperados ✨


O resultado esperado dessa atividade é permitir ao aluno aplicar, de forma prática e conjunta, os conhecimentos abordados nas microatividades através de uma atividade que envolve a manipulação de dados em arquivos e o uso de algoritmos de ordenação.


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!**

