



INSTRUÇÃO PRÁTICA		PP-P002
MÓDULO	PP - PROGRAMAÇÃO EM PYTHON	
OBJETIVO DA ATIVIDADE		TEMPO
<ul style="list-style-type: none"><li>• Dominar as características básicas dos tipos de dados avançados: dicionários e conjuntos;</li><li>• Explorar a implementação e o uso de estruturas de controle de fluxo, estruturas condicionais e de repetição, em python.</li></ul>		4h
DESCRIÇÃO		
<p><b>Exercício 1: Utilizando estruturas de controle de fluxo e listas.</b></p> <p>Para controlar os compromissos do dia a dia podemos utilizar uma lista de tarefas. Com alguns dos recursos básicos da linguagem que já conhecemos podemos implementar o aplicativo ToDoList que permita:</p> <ul style="list-style-type: none"><li>• Listar as tarefas que estão registradas.<ul style="list-style-type: none"><li>○ As tarefas já finalizadas aparecem no início da lista identificadas por um box confirmado ([x]) no final da tarefa;</li><li>○ As tarefas pendentes aparecem logo a seguir identificadas por um box vazio ([ ]) no final da tarefa;</li><li>○ Cada tarefa é precedida por um ID, número sequencial atribuído no momento que ela foi cadastrada;</li><li>○ Exemplo:<ol style="list-style-type: none"><li>1. Preparar a marmita [x]</li><li>2. Arrumar a mochila [ ]</li><li>3. Fechar as janelas [ ]</li></ol></li></ul></li><li>• Registrar uma nova tarefa.<ul style="list-style-type: none"><li>○ Uma descrição da tarefa é solicitada ao usuário (Exemplo: “arrumar o quarto”);</li><li>○ A tarefa é registrada e a ela é atribuído um ID e um box vazio é adicionado no final da string com a descrição da tarefa. (Exemplo: “3.arrumar o quarto[ ]”);</li><li>○ No momento de registrar, deve-se garantir que a string com a descrição da tarefa começa com maiúscula (Exemplo: “3.Arrumar o quarto[ ]”);</li><li>○ Uma mensagem confirmando a execução da tarefa deve ser apresentada. (Exemplo: “Tarefa registrada!!!”)</li></ul></li><li>• Marcar uma tarefa como realizada.<ul style="list-style-type: none"><li>○ O aplicativo solicita o identificador da tarefa e, existindo, ela é movida para o início da lista e o box vazio no final é substituído por um box confirmado;</li><li>○ Caso o identificado não exista ou a tarefa já tenha sido realizada nada será feito.</li><li>○ Uma mensagem confirmando a execução da tarefa deve ser apresentada.</li></ul></li><li>• Editar uma tarefa.</li></ul>		



- O aplicativo solicita o identificador da tarefa e, existindo, é solicitada uma nova descrição da mesma;
- O status do box da tarefa e o identificador da mesma não pode ser alterado na edição;
- Uma mensagem confirmando a execução da tarefa deve ser apresentada.

### Exercício 2: Pesquisa sobre persistência de dados.

- Durante o módulo anterior abordamos o tema de persistência de dados utilizando arquivos. Pesquise sobre arquivos em Python e proponha as modificações necessárias para que o aplicativo do exercício anterior utilize um arquivo para armazenar a lista de tarefas.

### Fechamento da atividade

- Lembre de colocar a implementação na pasta apropriada do repositório.
- Utilizem o fórum do Moodle para tirar suas dúvidas sobre as atividades e interagir entre os membros;
- Garanta que seu repositório do GitHub esteja atualizado com todos os branch e versão final do branch principal.
- A data limite para entrega desta atividade está definida no moodle.

### REFERÊNCIAS

A Whirlwind Tour of Python by Jake VanderPlas (O'Reilly). Copyright 2016 O'Reilly Media, Inc., 978-1-491-96465-1. [link](#)

Python data science handbook : essential tools for working with data. VanderPlas, J. (2016). Sebastopol, CA: O'Reilly Media, Inc. ISBN: 978-1491912058 [link](#)

How to Think Like a Computer Scientist: Learning with Python (2nd. ed.). Jeffrey Elkner, Allen B. Downey, and Chris Meyers. 2016. Samurai Media Limited, London, GBR. [link](#)