

Relatório 02 - Prática: Linguagem de Programação Python (I)

Fernando Luís Poletto

Descrição da atividade

Nesse minicurso sobre a linguagem de programação Python foram abordados alguns assuntos sobre sua sintaxe básica, focando mais em suas funcionalidades. No primeiro vídeo o instrutor fala sobre os tipos de dados, operações, estruturas de controle e pacotes.

Os principais tipos de dados são numéricos como int e float, booleanos: True e False, e strings que são textos. Em Python os dados podem ser organizados e agrupados em diferentes tipos de estruturas que são chamadas de coleções, por exemplo: listas, tuplas, conjuntos e dicionários.

Cada uma delas possuem diferentes características, tuplas são imutáveis enquanto as listas podem ser alteradas, ambas são indexadas o que permite acessar qualquer elemento diretamente, diferente dos conjuntos em que não é possível acessar seus elementos de forma direta além de não aceitarem elementos iguais, porém ele é mutável. O dicionário é como um arquivo JSON, onde cada elemento é composto por uma chave única e um valor.

As operações podem ser classificadas em unárias, binárias e ternárias que são os números de operandos que cada operador pode manipular, onde unárias possuem apenas um operando, binárias e ternárias 2 e 3 respectivamente.

Estruturas de controle servem para controlar o fluxo de execução do algoritmo que são basicamente formadas por estruturas condicionais e loops, sendo elas o “IF e ELSE” as condicionais e “FOR e WHILE” as de loop.

O conceito de pacotes pode ser usado para formar módulos e bibliotecas em Python, onde é possível reaproveitar o código e a lógica de algum algoritmo que já foi criado para ser importado e incorporado ao projeto.

O segundo vídeo tem foco na programação funcional e uma introdução à programação orientada a objetos.

Na programação funcional o foco são as funções, enfatizando o uso de funções puras, que são funções em que ela sempre retorna o mesmo resultado se receber os mesmos argumentos, onde elas deixam o código mais limpo e modular. Também são incentivados o uso de estruturas imutáveis, como as tuplas, a fim de evitar estados mutáveis e possíveis efeitos colaterais.

Nesse paradigma também é encorajado o uso de funções de alta ordem, de um modo simples elas são funções que podem ter como retorno outras funções, ou passar funções como argumentos, com isso é possível uma maior flexibilidade e reaproveitamento de código.

Já a programação orientada a objetos (P.O.O) é um conceito mais antropomorfizado, em que o código é organizado de maneira mais intuitiva e também modular trazendo um lado mais humano para facilitar a lógica e o entendimento do código.

Nesse estilo as principais características são as classes e objetos, onde a classe seria um modelo para instanciar cada objeto seu, esse modelo é composto por atributos, que são as características desses objetos, e os métodos, que seriam suas funcionalidades, por exemplo: em uma possível classe “Carro”, seus atributos podem ser sua cor, modelo, ano de fabricação e outros dados nesse sentido, e seus métodos podem ser acelerar, frear, buzinar e esses tipos de coisas que carros são capazes de fazer.

Alguns conceitos básicos desse paradigma de programação são a Herança, Polimorfismo e Encapsulamento.

Na Herança uma classe herda outra classe a fim de reutilizar seu código e manter uma lógica e organização do código criando uma hierarquia, exemplo uma classe “Gato” e outra “Cachorro” que herdam de uma classe “Animal”, onde ambos, gato e cachorro, tem características e comportamentos em comum que podem ser implementados na classe “Animal” evitando código duplicado, e suas peculiaridades podem ser inseridas em suas próprias classes.

Ainda usando esse exemplo é possível explicar o Polimorfismo em POO, onde torna possível um método com mesmo nome ter comportamentos comuns em diferentes classes, exemplo: na classe Animal tem um método “falar()” que imprime “o animal está falando”, na classe “Gato” também tem esse mesmo método “falar()” porém ele imprime “o gato está miando”, desse modo se um objeto da classe “Gato” usar esse método será impresso que o gato está miando mesmo que essa classe herde da sua classe pai, e que para invocar essa função de sua classe herdada, teria que deixar isso explícito no código com a palavra reservada “super”.

Finalizando esses conceitos básicos também existe o Encapsulamento que serve para limitar o acesso a dados sensíveis das classes, indicando que seus atributos podem ser privados e que para acessá-los podem ser feitos métodos para controlar isso, que são chamados de “Getters” para somente obter seus dados e os “Setters” que podem alterá-los.

Conclusões

Essa atividade permitiu o aprendizado básico da linguagem Python, conhecendo sua sintaxe, estruturas, paradigmas, e como utilizá-los em exemplos práticos. Aplicando os conceitos ensinados nas aulas torna possível desenvolver soluções na área de programação de forma eficiente.

Todo o código fonte que foi feito seguindo o aprendizado dos vídeos está disponível no repositório:

<https://github.com/FernandoLPX/LAMIA/tree/main/Bootcamp/Atividade02>

Referências

Vídeo 1: https://www.youtube.com/watch?v=oUrBHiT-lzo&ab_channel=Cod3rCursos

Vídeo 2: https://www.youtube.com/watch?v=iq7JLIH-sV0&ab_channel=Cod3rCursos