Fatec
Rio Claro

Linguagem de Programação I

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

"Testing leads to failure,

and failure leads to understanding."

Burt Rutan

"Testar leva ao fracasso,

e o fracasso leva à compreensão."

Burt Rutan

Módulos

Módulos

Fatec
Rio Claro

Em Python, **módulos** são arquivos que contêm definições e instruções em código Python, normalmente organizados em funções, classes e variáveis reutilizáveis.

Eles permitem dividir programas grandes em partes menores, facilitando a manutenção, o reaproveitamento de código e a organização do projeto.

Qualquer arquivo .py pode ser considerado um módulo, e pode ser importado em outros scripts por meio da instrução *import*.

Módulos

Fatec
Rio Claro

O uso de módulos também é essencial para aproveitar a vasta biblioteca padrão do Python, que inclui centenas de módulos prontos para tarefas comuns como manipulação de arquivos, datas, expressões regulares, matemática, e muito mais.

Além disso, Python permite importar módulos externos (de terceiros), como *pandas*, *numpy* ou *requests*, usando o gerenciador de pacotes *pip*.

Essa abordagem modular torna Python altamente escalável e produtivo, permitindo que desenvolvedores construam soluções complexas com mais clareza e eficiência.

Módulos

Fatec
Rio Claro

```
# Importa o módulo completo
import math

# Importa função específica
from math import sqrt

# Define um alias
import datetime as dt
```

Datetime

Módulo Datetime

Fatec
Rio Claro

O módulo datetime da biblioteca padrão do Python fornece classes para manipulação de datas e horários de forma simples e eficiente.

Ele permite representar momentos no tempo (***datetime***), apenas a data (***date***), apenas a hora (***time***) ou intervalos de tempo (***timedelta***).

Com ele, é possível obter a data e hora atual do sistema, criar datas específicas, calcular diferenças entre datas, formatar saídas e muito mais.

Módulo Datetime

Fatec
Rio Claro

Isso é especialmente útil em aplicações que precisam registrar eventos, agendar tarefas ou calcular prazos.

O módulo oferece métodos de formatação (strftime) e leitura (strptime) que facilitam a conversão entre objetos de data/hora e strings em diferentes formatos.

Por ser parte da biblioteca padrão, o datetime está sempre disponível em qualquer instalação do Python, sem a necessidade de pacotes externos.

CSV

Módulo CSV

Fatec
Rio Claro

Em Python, a manipulação de arquivos de texto é feita com a função integrada open(), que permite abrir arquivos em diferentes modos, como leitura ('r'), escrita ('w') e adição ('a').

Após abrir um arquivo, é possível ler seu conteúdo com métodos como read(), readline() ou readlines(), ou escrever com write() e writelines().

Sempre que se trabalha com arquivos, é uma boa prática utilizar a estrutura with, que garante o fechamento automático do arquivo após sua utilização, evitando erros e vazamentos de recursos do sistema.

Módulo CSV

Fatec
Rio Claro

O formato CSV (Comma-Separated Values) é amplamente utilizado para armazenar dados tabulares simples, onde cada linha representa um registro e os valores são separados por vírgulas (ou outro delimitador).

Ele é especialmente útil para troca de dados entre sistemas e pode ser facilmente aberto por programas como o Excel ou Google Sheets.

Apesar de parecer simples, arquivos CSV podem conter variações no delimitador, presença de aspas, quebra de linhas dentro de campos e outros detalhes que exigem um tratamento cuidadoso ao serem lidos ou escritos por programas.

Módulo CSV

Fatec
Rio Claro

A biblioteca csv, que faz parte da biblioteca padrão do Python, foi projetada para lidar precisamente com essas particularidades.

Com ela, é possível ler e escrever arquivos CSV de maneira segura e eficiente, utilizando objetos como csv.reader, csv.writer, csv.DictReader e csv.DictWriter.

Em especial, o DictReader permite ler arquivos CSV diretamente como dicionários, onde cada linha é representada como um dict cujas chaves correspondem aos nomes das colunas.

Isso facilita o acesso aos dados por nome, tornando o código mais legível e robusto, especialmente em arquivos com muitas colunas ou estruturas complexas.

Tratamento de Erro

Tratamento de erros

Fatec
Rio Claro

O comando **with** em Python é utilizado para simplificar a manipulação de recursos externos como arquivos, conexões de rede e banco de dados.

Ele garante que, ao final do bloco de código, o recurso seja fechado ou liberado corretamente, mesmo que ocorra um erro durante sua execução. Isso evita problemas como arquivos deixados abertos ou uso excessivo de memória.

Por exemplo, ao abrir um arquivo com

```
with open("dados.txt", "r") as arquivo:
    #codigo
```

não é necessário chamar arquivo.close() manualmente. Isso será feito automaticamente.

Tratamento de erros

Fatec
Rio Claro

O bloco **try-except** permite capturar e tratar erros que possam ocorrer durante a execução do programa.

Isso é fundamental para tornar o código mais robusto e confiável, pois evita que o programa seja interrompido abruptamente em caso de exceções.

A estrutura básica consiste em colocar o código que pode gerar erro dentro do *try*, e o que deve ser feito caso o erro ocorra dentro do *except*.

É possível tratar diferentes tipos de erros de forma específica, como *FileNotFoundError*, *ZeroDivisionError* ou *ValueError*, oferecendo mensagens mais claras ao usuário ou estratégias alternativas de execução.

Tratamento de erros

Fatec
Rio Claro

Erros comuns em Python incluem digitar o nome de uma variável que ainda não foi definida (*NameError*), tentar abrir um arquivo que não existe (*FileNotFoundError*), acessar um índice inválido em uma lista (IndexError) ou dividir por zero (*ZeroDivisionError*).

Saber identificar e tratar esses erros com ***try-except*** é essencial em qualquer aplicação real.

A combinação do ***with*** para gerenciar recursos e do try-except para capturar exceções é uma boa prática que contribui para um código mais limpo, seguro e profissional.

Requests

Módulo requests

Fatec
Rio Claro

O módulo **requests** é uma biblioteca externa do Python amplamente utilizada para realizar requisições HTTP de forma simples e intuitiva.

Com ele, é possível enviar requisições do tipo GET, POST, PUT, DELETE, entre outras, para servidores web e APIs, obtendo respostas que podem ser tratadas diretamente no código Python.

Por exemplo, com apenas uma linha como requests.get("https://api.exemplo.com/dados"), é possível buscar informações de um serviço online.

Módulo requests

Fatec
Rio Claro

O módulo também permite configurar cabeçalhos, autenticação, parâmetros de URL, envio de dados em JSON ou formulários, entre outros recursos essenciais para comunicação com a web.

A resposta obtida de uma requisição, representada por um objeto Response, traz dados importantes como o código de status HTTP, cabeçalhos e o conteúdo retornado (geralmente acessado via response.text ou response.json() para APIs que retornam JSON).

Por não fazer parte da biblioteca padrão do Python, o requests precisa ser instalado separadamente com o comando *pip install requests*, mas é considerado uma ferramenta essencial para quem desenvolve aplicações que interagem com a internet.

Fatec
Rio Claro

Dúvidas

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br