

Módulo 07

Bibliotecas Python (Aula 08)



Questões de Aprendizagem

Questão 1:

Matemática e Estatística

Desenvolva um programa que:

- Gere 50 números aleatórios inteiros entre 1 e 100
- Calcule a média, mediana usando a biblioteca statistics
- Encontre o maior e menor valor
- Calcule a raiz quadrada do maior número

Questão 2:

Manipulação de Arquivos e Diretórios

Crie um script que:

- Liste todos os arquivos do diretório atual
- Crie uma nova pasta chamada "backup"
- Copie todos os arquivos .py para esta pasta (use a biblioteca pathlib, e pesquisa sobre "glob pattern")
- Imprima o caminho absoluto de cada arquivo copiado

Questão 3:

Compressão de Dados

Crie um comparador que:

- Comprima arquivos com diferentes métodos (pesquise sobre os módulos nativos de compressão do python)
- Compare taxas de compressão
- Meça tempo de compressão/descompressão
- Gere relatório comparativo

Módulo 07

Bibliotecas Python (Aula 08)



Questões de Aprendizagem

Questão 4:

Datetime

Crie um programa que:

- Mostre a data e hora atual
- Crie uma data específica (seu aniversário)
- Calcule quantos dias faltam para o Natal deste ano
- Formate a saída no padrão brasileiro (dd/mm/aaaa)

Questão 5:

Requisições HTTP

Desenvolva um programa que faça uma requisição GET para uma API pública da sua escolha (usando a biblioteca requests), converta o resultado de JSON para dicionário/lista e imprima as informações.

Questão 6:

Instalando Pacotes

No exercício anterior você precisou instalar o pacote requests. Mostre onde ele está armazenado na sua máquina, tire um print da pasta mostrando que o requests está lá.

Questão 7:

Medição de Tempo

Implemente duas formas diferentes de gerar uma lista de números de 1 a 1000000. Use o timeit para comparar o desempenho entre:

- List comprehension
- Loop for tradicional com append numa lista

Módulo 07

Bibliotecas Python (Aula 08)



Questões de Aprendizagem

Questão 8

Processamento de logs com expressão regular

Desenvolva um parser de logs que converte entradas de log de servidor web em uma lista de dicionários Python.

Arquivo de Log Exemplo (server_access.log):

```
192.168.1.100 [25/Jan/2024:10:15:32 +0300] "GET /products/electronics
HTTP/1.1" 200 1458 "Mozilla/5.0" 0.234s
10.0.0.45 [25/Jan/2024:10:15:40 +0300] "POST /api/users HTTP/1.1" 201 532
"Python-Requests/2.28.1" 0.891s
172.16.0.89 [25/Jan/2024:10:15:45 +0300] "GET /images/banner.jpg HTTP/1.1"
304 0 "Chrome/120.0.0.0" 0.045s
192.168.1.105 [25/Jan/2024:10:15:50 +0300] "DELETE /api/cart/15 HTTP/1.1"
204 0 "PostmanRuntime/7.32.3" 0.123s
```

Requisitos:

1. Leia linha a linha o arquivo e use o módulo 're' para extrair as seguintes informações de cada linha:
 - IP do cliente
 - Data e hora
 - Método HTTP (GET, POST, etc)
 - URL acessada
 - Código de status HTTP
 - Tamanho da resposta
 - User Agent
 - Tempo de resposta
2. Converta cada linha em um dicionário com estas informações como chaves
3. Crie uma função que retorne uma lista com todos os dicionários
4. Implemente validação básica do formato das linhas

Módulo 07

Bibliotecas Python (Aula 08)



Questões de Aprendizagem

Exemplo de Saída Esperada:

```
[
  {
    'ip': '192.168.1.100',
    'timestamp': '25/Jan/2024:10:15:32 +0300',
    'method': 'GET',
    'path': '/products/electronics',
    'status': 200,
    'size': 1458,
    'user_agent': 'Mozilla/5.0',
    'response_time': 0.234
  },
  # ... demais entradas
]
```

Questão 9:

Testes Unitários

Crie uma classe Calculadora com operações básicas (soma, subtração, multiplicação, divisão) e desenvolva:

- Doctests para cada método
- Testes unitários usando unittest

(Pesquise !)