

Dicionários, JSON e Estruturas de Dados

Objetivos:

- - Compreender dicionários
- - Usar biblioteca JSON
- - Aplicar estruturas de dados para organizar informações.

Dicionários em Python

Um dicionário armazena pares chave:valor únicos.

Ex:

```
aluno = {"nome": "Maria", "idade": 17, "nota":  
8.5}
```

Acessando e Modificando

Acesso: `aluno["nome"]`

Adicionar/Atualizar: `aluno["turma"] = "3B"`

Remoção: `del aluno["idade"]`

Métodos de Dicionários

`aluno.keys()` -> chaves

`aluno.values()` -> valores

`aluno.items()` -> pares chave-valor

Biblioteca JSON

Permite converter dados Python para JSON e vice-versa.

Importação: `import json`

Serialização e Desserialização

Serializar:

```
json.dumps(dado)
```

Desserializar:

```
json.loads(json_str)
```

Trabalhando com Arquivos

Salvar JSON:

```
json.dump(obj, arquivo)
```

Ler JSON:

```
json.load(arquivo)
```


Exemplo com Lista de Dicionários

```
produtos = [{"id":1, "nome":"Notebook",  
"preco":3500.00}, {"id":2, "nome":"Mouse",  
"preco":50.00}]
```

```
json.dumps(produtos, indent=2)
```

Busca em Estrutura

```
def buscar(nome):  
    for p in produtos:  
        if p["nome"]==nome:  
            return p  
    return "Não encontrado"
```

Trabalho com arquivos(leitura)

```
with open("dados.json", "r", encoding="utf-8")  
as arquivo:
```

```
    conteudo = json.load(arquivo)
```

```
    print(conteudo["nome"]) # Acessar valor de  
uma chave
```

Trabalho com arquivos(escrita)

with open("dados.json", "w", encoding="utf-8")
as arquivo:

```
    json.dump(dados, arquivo, indent=4,  
ensure_ascii=False)
```

Exercícios

1. Crie um dicionário de carro
2. Atualize e adicione nova chave
3. Converta para JSON
4. Salve em arquivo
5. Crie função que retorna média das notas