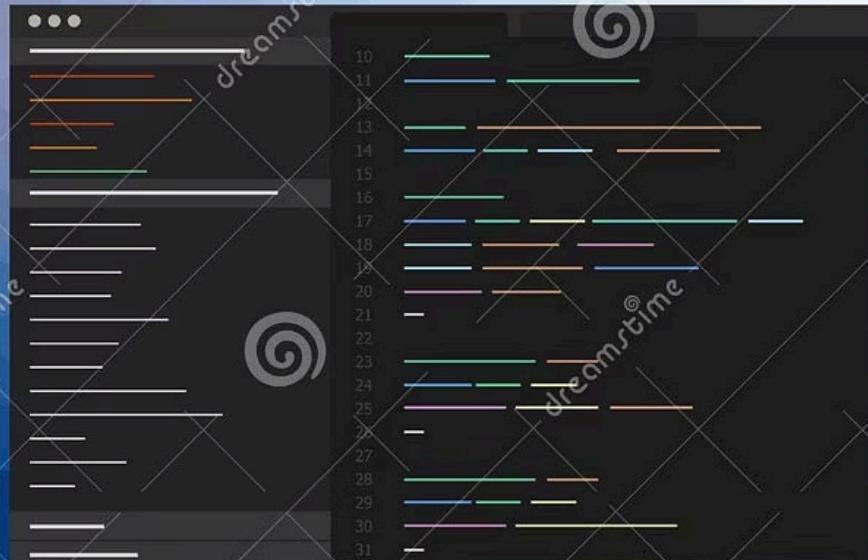


PYTHON

PROGRAMMING LANGUAGE



🎯 Projeto 2: Lista de Tarefas

Aprenda os conceitos fundamentais de Python através da construção de um projeto prático e funcional.

Slide 11: Listas Mutáveis - Append e Pop

Conceito: Modificando listas

.append(): adiciona elemento ao final

.pop(): remove elemento por índice

No seu código:

```
tarefas.append(tarefa) # adiciona  
tarefas.pop(num-1) # remove
```

Slide 12: Validação de Entrada

Conceito: Garantindo dados corretos

.isdigit(): verifica se string contém apenas números

No seu código:

```
if num.isdigit():
    num = int(num)
```



DATA VALIDATION

Slide 13: Funções - Organizando o Código

Conceito: Funções definidas pelo usuário

def: define uma nova função

Vantagens: reutilização, organização, legibilidade

Sintaxe:

```
def nome_funcao():
    # código aqui
```

No seu código:

```
def adicionar():
    tarefa = input("Digite a nova tarefa: ").strip()
    tarefas.append(tarefa)
```

Data Item	Data Type	Data Format	Bytes for Storage	Display			
MemberID	String	XNNNNNNN	7	7	Unique Identifier For Member	M123456	
First Name	String		25	25	First Name of Member	Scott	
Surname	String		25	25	Last Name of Member	Daniels	
D.O.B	Floating Point (Date Format)	DD/MM/YYYY	4	10	Birth Date of Member	02/04/1990	Date < Today - 15 years

Slide 14: Dicionários - Estruturas Chave-Valor

Conceito: Armazenando dados complexos

- **Dicionário:** coleção de pares chave-valor
- **Sintaxe:** meu_dict = {"chave": "valor", "nome": "João"}
- **Acesso:** meu_dict["chave"] → "valor"

No seu código:

```
tarefa = {"texto": texto, "concluida": False, "data": "..."}  
Made with GAMMA
```

Slide 15: Módulos - Importações

Conceito: Usando código externo

import: importa funcionalidades extras

Módulos úteis:



datetime: trabalhar com datas



json: manipular arquivos JSON

No seu código: from datetime import datetime

Slide 16: Tratamento de Erros - Try/Except

Conceito: Lidando com exceções

try/except: captura e trata erros

Sintaxe:

```
try:  
    # código perigoso  
except TipoErro:  
    # tratamento do erro
```

No seu código:

```
try:  
    indice = int(input("Número da tarefa: ")) - 1  
    tarefas[indice]["concluida"] = True  
except (ValueError, IndexError):  
    print("Número inválido.")
```

Slide 17: Trabalhando com Arquivos

Conceito: Persistência de dados



with open(): abre e fecha arquivo automaticamente



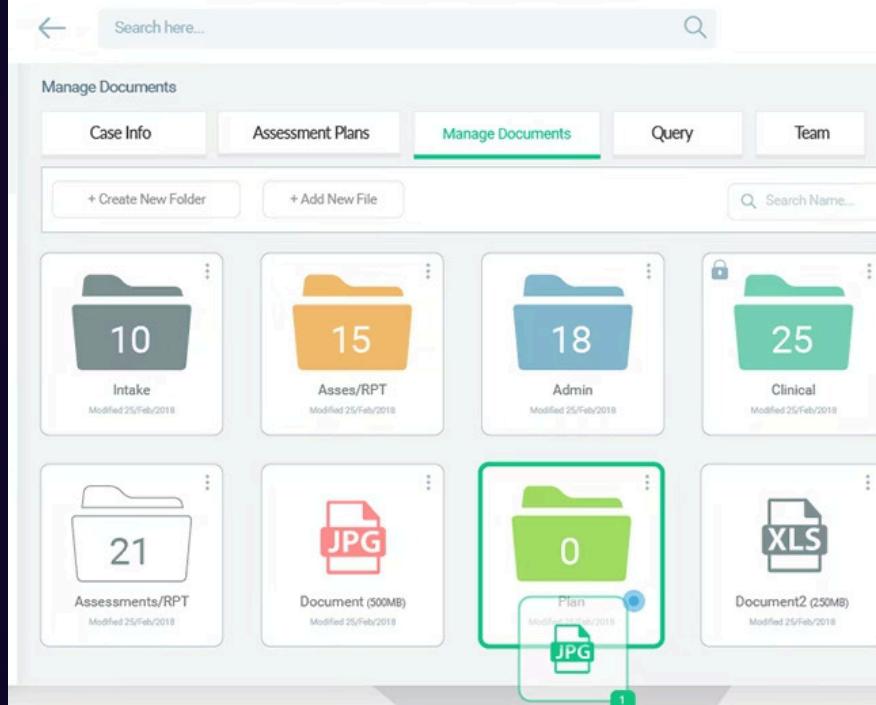
Modos: "r" (leitura), "w" (escrita), "a" (adicionar)

No seu código:

```
with open(ARQUIVO, "r", encoding="utf-8") as f:  
    return json.load(f)
```

File Management System

File upload option through device, drop-box, Google drive.
drag and drop in file and folder



Slide 18: JSON - Armazenando Dados Estruturados

Conceito: Trabalhando com JSON

- **JSON:** formato para troca de dados
- **json.load():** lê JSON do arquivo
- **json.dump():** grava JSON no arquivo

No seu código:

```
def salvar(tarefas):
    with open(ARQUIVO, "w") as f:
        json.dump(tarefas, f)
```

Slide 19: Erros Comuns de Iniciantes:

- Esquecer os : após if, for, def
- Confundir = (atribuição) com == (comparação)
- Índices começando em 1 em vez de 0