def load\_docs\_from\_path(pdf\_path):

```

- Defines a function named `load\_docs\_from\_path`.

- It takes one argument: `pdf\_path`, which is expected to be the \*\*file path\*\* to a PDF document (e.g., `'E:/Chatbot/Data/Dell\_data.pdf'`).

---

### 📄 Load PDF Document

```python

loader = PyPDFLoader(pdf\_path)

documents = loader.load()

```

- `PyPDFLoader(pdf\_path)`: Initializes the loader with your specified PDF file.

- `.load()`: Reads the PDF and returns a list of `Document` objects — typically one per page.

---

### ✂️ Split Into Chunks

```python

text\_splitter = RecursiveCharacterTextSplitter(chunk\_size=2000, chunk\_overlap=100)

final\_documents = text\_splitter.split\_documents(documents)

```

- `RecursiveCharacterTextSplitter`: Smartly splits the text to keep it meaningful (e.g., tries to break at sentence/paragraph level).

- `chunk\_size=2000`: Each chunk will contain \*\*up to 2000 characters\*\*.

- `chunk\_overlap=100`: \*\*100 characters of overlap\*\* between chunks to maintain context across splits.

---

### 🔁 Return Final List of Chunks

```python

return final\_documents

```

- Returns a list of text chunks (still `Document` objects) which are now ready to:

- Be embedded using Hugging Face models.

- Be stored in a vector database like FAISS.

---

### ✅ What This Function Does in Simple Terms

This function:

1. Takes a PDF path.

2. Reads the document.

3. Splits the content into manageable, overlapping pieces.

4. Returns those pieces for further processing (embedding, storage, search, etc.).