

# Model Context Protocol (MCP) – Documentation

## Introduction

The Model Context Protocol (MCP) is a fundamental concept in working with Large Language Models (LLMs). At its core, MCP is about how we provide input (context) to a model so that it can generate high-quality and accurate outputs. This documentation explores the core idea of 'context', its limitations, and the best practices for structuring it.

## What is Model Context?

A Large Language Model (LLM) is essentially a text predictor—it predicts the next word based on the words it has seen so far. The 'context' is the total amount of text that the model receives as input in a single request. Think of it as having a conversation with someone who has a short-term memory. You must tell them everything they need to know upfront in order for them to understand and respond accurately.

## Key Concepts of MCP

### 1 Context Window

LLMs have a limited memory space called a context window. This is measured in tokens (which are roughly equivalent to words or parts of words). If the input exceeds this limit, the model will start to forget the earliest parts of the conversation.

### 2 Context is Everything

The output quality of an LLM depends heavily on the quality, structure, and relevance of the input context. Well-structured instructions, rich examples, and concise problem definitions all help the model understand and respond better.

### 3 The Protocol

The 'protocol' in MCP refers to how we organize the context text. This includes how we write instructions, add examples, and include relevant information for the model to work effectively. A well-structured protocol makes the model more useful, accurate, and consistent.

## Case Study: PDF-based Q&A System Using LLM

Imagine a Django web application that lets users upload PDF files and ask questions based on their content. The backend extracts text from the PDF, breaks it into chunks (due to context window limits), and then forms a prompt to ask a local LLM (like Mistral via Ollama). This prompt must be carefully constructed to provide both relevant context and the user's question.

For example, the model might receive:

Context:

Page 1: 'ISO 21500:2012 provides guidance on project management.'

Page 2: 'It outlines good practices for managing projects across all industries.'

Question: 'What does ISO 21500 recommend about managing projects?'

Answer: (Model will infer based on provided context.)

## 5. Conclusion

The Model Context Protocol is essential for leveraging the power of LLMs. By understanding the context window, structuring input wisely, and maintaining a strong protocol, developers and users can achieve more accurate, useful, and context-aware model outputs.