

# Customize AI behavior in Cursor

Using rules in Cursor you can control the behavior of the underlying model. You can think of it as instructions and/or a system prompt for LLMs.

- [Chapter №1 Rules for AI](#)
  - [№1 Rules for AI](#)

# Chapter №1 Rules for AI

# №1 Rules for AI

Using rules in Cursor, you can control the behavior of the underlying model. You can think of it as instructions and/or a system prompt for LLMs. You can use Cursor by two ways:

- wrong;
- correct.

## Wrong examples of using Cursor:

- Using Cursor as a replacement for Google Search.
- Underspecification of prompts, not knowing how to drive outcomes and using low-level thinking of "implement XYZ, please".
  - Treating Cursor as if it is an IDE, instead of it being an autonomous agent.
  - Blissful unawareness of the concept that you can program LLM outcomes.
  - Unnecessary usage of pleasantries ("please" and "can you") with it as if it were a human. If it fucks up, swear at it - go all caps and call it a clown. It soothes the soul.

**Important Notice!** The `.cursorrules` are deprecated and will be removed in the near future

## Correct examples of using Cursor:

Inside Cursor, we have two main ways to customize the behavior of the AI to suit your needs:

**1. Global Rules.** Rules applied globally to all projects, configured in the Cursor Settings > General > Rules for AI section.

Global rules can be added by modifying the Rules for AI section under Cursor Settings > General > Rules for AI. This is useful if you want to specify rules that should always be included in every project like output language, length of responses etc.

**2. Project Rules.** Rules specific to a project, stored in the `.cursor/rules` directory. They are automatically included when matching files are referenced.

Project rules offer a powerful and flexible system with path specific configurations. Project rules

are stored in the `.cursor/rules` directory and provide granular control over AI behavior in different parts of your project.

### Here's how they work

- **Semantic Descriptions:** Each rule can include a description of when it should be applied
- **File Pattern Matching:** Use glob patterns to specify which `files/` `folders` the rule applies to
- **Automatic Attachment:** Rules can be automatically included when matching files are referenced
- **Reference files:** Use `@file` in your project rules to include them as context when the rule is applied.

## Resource Links

- 1) [You are using Cursor AI incorrectly](#)
- 2) [Optimal structure for `.mdc` rules files](#)
- 3) [Examples of cursor rules](#)