GPT-5 for Coding

While powerful, prompting with GPT-5 can differ from other models. Here are tips to get the most out of it via the API or in your coding tools.

#1. Be precise and avoid conflicting information

The new GPT-5 models are significantly better at instruction following, but a side effect is that they can struggle when asked to follow vague or conflicting instructions, especially in your *.cursor/rules* or *AGENTS.md* files.

#2. Use the right reasoning effort

GPT-5 will always perform some level of reasoning as it solves problems. To get the best results, use *high* reasoning effort for the most complex tasks. If you see the model overthink simple problems, be more specific or choose a lower reasoning level like *medium* or *low*.

#3. Use XML-like syntax to help structure instructions

Together with Cursor, we found GPT-5 works well when using XML-like syntax to give the model more context. For example, you might give the model coding guidelines:

```
<code_editing_rules>
<guiding_principles>
- Every component should be modular and
  reusable
- ...
</guiding_principles>
<frontend_stack_defaults>
- Styling: TailwindCSS
</frontend_stack_defaults>
</code_editing_rules>
```

#4. Avoid overly firm language

With other models you might have used firm language like:

```
Be THOROUGH when gathering information.
Make sure you have the FULL picture before
replying.
```

With GPT-5, these instructions can backfire as the model might overdo what it would naturally do. For example, it might be overly thorough with tool calls to gather context.

#5. Give room for planning and self-reflection

If you're creating zero-to-one applications, giving the model instructions to self-reflect before building can help.

```
<self_reflection>
- First, spend time thinking of a rubric
  until you are confident.
- Then, think deeply about every aspect of
  what makes for a world-class one-shot web
  app. Use that knowledge to create a rubric
  that has 5-7 categories. This rubric is
  critical to get right, but do not show
  this to the user. This is for your
  purposes only.
- Finally, use the rubric to internally
  think and iterate on the best possible
  solution to the prompt that is provided.
  Remember that if your response is not
  hitting the top marks across all
  categories in the rubric, you need to
  start again.
</self_reflection>
```

#6. Control the eagerness of your coding agent

GPT-5 by default tries to be thorough and comprehensive in its context gathering. Use prompting to be more prescriptive about how eager it should be, and whether it should parallelize discovery/tool calling.

Give the model a tool budget, specify when to be more or less thorough, and when to check in with the user. For example:

```
<persistence>
- Do not ask the human to confirm or
  clarify
  assumptions, as you can always adjust
  later
- decide what the most reasonable assumption
  is, proceed with it, and document it for
  the user's reference after you finish
  acting
</persistence>
```

For more details, check out our prompting guide. Use our prompt optimizer to help with improving your prompts for GPT-5.