Using GPT-related tools for Coding: Good Insights/Practices

Before we get into things, this is the biggest game changer that I have personally found. Please use cursor AI code editor OR the bito AI coding assistant in vscode. Both of these are leagues better than co-pilot in my opinion and it's like having a more adept version of chatgpt directly in your editor. I prefer cursor, but the benefit of both of these tools is that they can directly understand your codebase and in cursor you can do @<filename> and you could add multiple files directly into context. You can also do something very similar for specific functions also.

Now here are some of the key things i've learned from using chatgpt/gpt related tools in every aspect of my coding process since chatgpt's release:

1. **Segmentation of Goals + "Think step-by-step":**

- Break down your coding project into manageable tasks. This approach not only optimizes token usage but also more importantly, enhances the model's accuracy by preventing information overload [so goddamn helpful].
- Had the following tip lower in the list, but I am going to squeeze this at the top here also because it is so important :) - For more complex tasks or even just for tasks in general, prompting chat gpt/gpt related tools to "think step-by-step" and then "execute the change/show me the code" etc is so helpful. I know this has been talked about online before but it actually does help. There has been some research that suggests that the usage of tokens (in this case planning) could potentially be a key part in GPT's "thinking process".

2. **Pseudocode as a Planning Tool:**

- You can sometimes pseudocode to outline your code structure. This can help ChatGPT understand your project's framework and maintain context. Rather than dumping entire files to gpt (which IS necessary sometimes), you can sometimes have gpt make pseudocode for a file or multiple files (separately) and then provide that as a contextual preface before beginning new threads. You still ofcourse need to include entire functions line by line and other logic related directly to what you are working on and what you are asking it to solve.

3. **UI Design and Customization:**

- When you encounter an appealing UI design on a website, consider using the inspect element feature to copy its code. ChatGPT can assist in stripping down unnecessary elements or customizing the code to fit your project's requirements.

4. **Leveraging Vision Capabilities:**

- Don't forget the potential of integrating screenshot-to-code functionalities, utilizing OpenAI's vision capability. This approach can be particularly useful for converting visual design (both drawn on paper and from other websites) into workable code snippets. Prompt engineering is important here, make sure gpt pays attention to certain specific aspects of the layout/other important things + explain functionality if needed. You will need to learn this with trial/error.
- GitHub “screenshot to code” repo is also sick - there is a site for it too.
- This can be hit or miss, but sometimes it’s so great.

5. **Focused Contextualization:**

- When working with extensive codebases or specific files, directing ChatGPT to focus on particular segments can enhance its understanding and output relevance. Mentioning specific functions or routes can help the model provide more contextually accurate suggestions or code modifications.
- I know that I mentioned that you can directly reference certain files with both of the tools that I recommended, but sometimes it's good to literally focus on certain code snippets from your codebase. This can be good for more difficult problems or if your files have too many lines of code/end up with slight information overload

6. **Iterative Development and Big Picture Consideration:**

- Approach coding tasks iteratively, especially when dealing with complex problems. However, always keep the overarching project goals in mind to ensure cohesive development. For example, when implementing features, consider future scalability and integration from the outset.
    - I know this is just general programming advice, but when you are adding certain features with chat GPT and other AI tools, you should keep this in mind because you know the full context of your project but chat GPT doesn't always (you will need to help it like mentioned earlier)

7. **Initial Context Setup:**

- At the start of a ChatGPT session, provide a brief description of your web app or project, including an overview of the codebase. This initial context setting can significantly improve the relevance and accuracy of subsequent interactions.
- I know I already kind of mentioned this, but chat GPT give me an outline of my notes and kind of included this again and it's so important that fuck it, I'll leave it
- Of course this is a little bit different if you are using one of the tools that I recommended, but even with those schools, having a nice little paragraph that describes your project is great when you are chatting with them and starting new threads.

Closing notes:

- I know I might have repeated a couple things, but I had a very extensive document and had chat GPT help me organize/summarize things and then did a lot of editing/rewriting on top of it
- Discover your own techniques and always try to learn what works for you and make your own notes
- Familiarize yourself with one of those two tools that I mentioned before and really learn how to use them well. Will step up your programming game drastically. At least it did for me. Also you can still use chat GPT directly in the browser sometimes, you just have to figure out for yourself when you want to do this
- DM me on Reddit if you have other questions