**GitHub Copilot and Copilot Chat Cheat Sheet**

**Autocomplete Suggestions**

*GitHub Copilot Hotkeys (Windows)*

* Accept an inline suggestion: `Tab`
* Dismiss an inline suggestion: `Esc`
* Show next inline suggestion: `Alt + ]`
* Show previous inline suggestion: `Alt + [`
* Trigger inline suggestion: `Alt + \`
* Open GitHub Copilot (additional suggestions in separate pane): `Alt + Enter`

*GitHub Copilot Hotkeys (Mac)*

* Accept an inline suggestion: `Tab`
* Dismiss an inline suggestion: `Esc`
* Show next inline suggestion: `Option (⌥)` or `Alt + ]`
* Show previous inline suggestion: `Option (⌥)` or `Alt + [`
* Trigger inline suggestion: `Option (⌥) + \`
* Open GitHub Copilot (additional suggestions in separate pane): `Option (⌥)` or `Alt + Return`

**GitHub Copilot Tips**

*Effective Prompts:*

* **Be Descriptive**: Use clear and detailed comments to describe the desired functionality.

Example: `// Generate a function to sort an array of integers`

* **Provide Examples**: Include input and output examples in comments to guide Copilot.

Example: `// Example input: [3, 1, 4], Example output: [1, 3, 4]`

* **Iterative Refinement**: Start with a basic prompt and refine it based on the suggestions provided by Copilot.

*Common Prompts:*

* **Basic Function Generation**:

Comment: `// Generate a function to reverse a string`

* **Complex Algorithm**:

Comment: `// Generate a func to find the longest common subsequence between two strings`

* **Data Processing**:

Comment: `// Generate a function to filter an array of objects based on a property value`

**GitHub Copilot Chat Tips**

*Engaging with Copilot Chat:*

* **Interactive Debugging**: Ask Copilot Chat specific questions about errors or issues to get detailed explanations and solutions.

Example: `Why am I getting a null pointer exception here?`

* **Contextual Help**: Use Copilot Chat to get context-aware assistance.

Example: `Can you explain what this piece of code is doing?`

* **Code Review Assistance**: Ask Copilot Chat to review code and identify potential issues.

Example: `Are there any potential issues with this function?`

*Common Queries:*

* Understanding Code: `What does this code do?`
* Improving Code: `How can I improve this function?`
* Fixing Errors: `How do I fix this error: [error message]?`
* Best Practices: `What are the best practices for implementing [feature]?`