



GitHub Expert Services



Introduction to Prompt Engineering for GitHub Copilot

Agenda



What is Prompt Engineering?



Importance of Prompt Engineering



Key Strategies for Effective Prompts



Steps in Prompt Engineering



Examples of Effective Prompts



Common Challenges



Best Practices

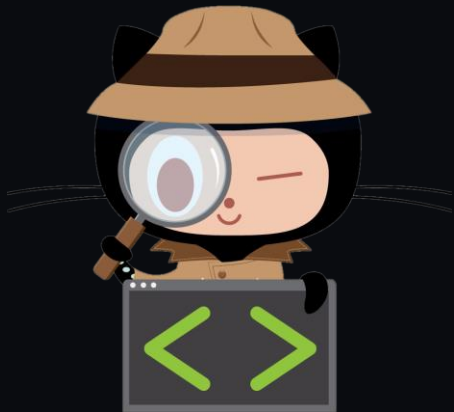


Q&A



Prompt Engineering

What is Prompt Engineering?



Prompt engineering is the process of designing and creating high-quality prompts that can be used to generate accurate and useful code suggestions with Copilot.

- Copilot

Why is Prompt Engineering Important?



- Maximises utility of Copilot
- minimises repeated iterations
- Determines code suggestion quality
- Skill prompts guide Copilot to understand context & nuances
- Refined interactions reduces misunderstandings

Key Strategies for Effective Prompts



- **Clarity and Specificity:** Clear and specific prompts reduce ambiguity and improve the accuracy of the AI's response.
- **Contextual Information:** Providing sufficient context helps the AI generate more relevant and precise outputs.
- **Iterative Refinement:** Testing and refining prompts through multiple iterations to achieve optimal results.

Best Practices for Effective Prompts

- **Start General, Then Get Specific:** Begin with a broad description and then list specific requirements.
- **Give Examples:** Use examples to help Copilot understand what you want.
- **Break Complex Tasks into Simpler Tasks:** Divide large tasks into smaller, manageable tasks.
- **Avoid Ambiguity:** Be specific and clear to avoid misunderstandings.
- **Indicate Relevant Code:** Specify any relevant code or libraries to be used.

Techniques



Zero-Shot Prompting
No Example



One-Shot Prompting
One Example



Few-Shot Prompting
Handful of Examples

Techniques

Zero-Shot Programming with Copilot



abelberhane

Write me unit tests for all of the operations in my calculator.js file.



GitHub Copilot

Sure! Here's an example of a test suite that covers all of the operations in a `calculator.js` file:

```
const { calculate } = require('./calculator');

describe('calculator', () => {
  describe('addition', () => {
    it('adds 1 + 2 to equal 3', () => {
      expect(calculate(1, 2, '+')).toBe(3);
    });

    it('adds -5 + 7 to equal 2', () => {
      expect(calculate(-5, 7, '+')).toBe(2);
    });

    it('adds 0.1 + 0.2 to equal 0.3', () => {
      expect(calculate(0.1, 0.2, '+')).toBeCloseTo(0.3);
    });
  });

  describe('subtraction', () => {
    it('subtracts 5 - 3 to equal 2', () => {
      expect(calculate(5, 3, '-')).toBe(2);
    });
  });
});
```



Techniques

One-shot Programming with Copilot



abelberhane

Write me unit tests for all of the operations in my calculator.js file.

Use the following example: `test('adds 1 + 2 to equal 3', () => { expect(calculate(1, 2, '+')).toBe(3); });`



GitHub Copilot

Here are some examples of unit tests for all of the operations in a `calculator.js` file:

```
const { calculate } = require('./calculator');

test('adds 1 + 2 to equal 3', () => {
  const result = calculate(1, 2, '+');
  expect(result).toBe(3);
});

test('subtracts 5 - 3 to equal 2', () => {
  const result = calculate(5, 3, '-');
  expect(result).toBe(2);
});

test('multiplies 2 * 4 to equal 8', () => {
  const result = calculate(2, 4, '*');
  expect(result).toBe(8);
});

test('divides 10 / 5 to equal 2', () => {
  const result = calculate(10, 5, '/');
  expect(result).toBe(2);
});

test('throws an error for an invalid operation', () => {
  expect(() => calculate(1, 2, '%')).toThrow('Invalid operation');
});

test('throws an error for division by zero', () => {
  expect(() => calculate(1, 0, '/')).toThrow('Division by zero');
});
```




Techniques

Cue with Copilot

/explain #file:Program.cs The purpose of this file is

[Program.cs](#) [Program.cs:1-30](#)

 GitHub Copilot

> Used 1 reference

The purpose of the [Program.cs](#) file is to set up and run a .NET application that ingests data from GitHub using HTTP clients. Here's a detailed explanation:

Using Directives

These bring in necessary namespaces for HTTP handling, dependency injection, and hosting:

```
using System.Net.Http.Headers;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Microsoft.CopilotDashboard.DataIngestion.Services;
```

Host Configuration

A [HostBuilder](#) is used to create and configure a host:

```
var host = new HostBuilder()
    .ConfigureFunctionsWebApplication()
    .ConfigureServices((ctx, services) =>
    {
        services.Configure<GithubMetricsApiOptions>(ctx.Configuration.GetSection<GithubMetricsApiOptions>());
        services.AddHttpClient<GithubCopilotMetricsClient>(ConfigureClient);
        services.AddHttpClient<GithubCopilotUsageClient>(ConfigureClient);
        services.AddHttpClient<GithubCopilotApiService>(ConfigureClient);
    })
    .Build();
```



Techniques

Supporting Content with Copilot

shin-akuma

create a powershell method that inserts a json file to cosmosdb collection.

test.ps1:230-258

GitHub Copilot

> Used 1 reference

Here is a PowerShell method that inserts a JSON file into a CosmosDB collection:

```
function Insert-JsonToCosmosDb (
    param (
        [string]$cosmosDbEndpoint,
        [string]$cosmosDbKey,
        [string]$databaseName,
        [string]$collectionName,
        [string]$jsonFilePath
    )
    # Install the Az.CosmosDB
    if (-not (Get-Module -List
        Install-Module -Name Az.CosmosDB -CurrentUser
    ))
    # Import the Az.CosmosDB module
    Import-Module Az.CosmosDB
    # Read the JSON file
    $jsonContent = Get-Content $jsonFilePath -Raw
    # Convert JSON content to a PowerShell object
    $jsonObject = $jsonContent | ConvertFrom-Json

```

shin-akuma

@github create a powershell method that inserts a json file to cosmosdb collection. Use this repo as reference <https://github.com/PlagueHO/CosmosDB>

get-changed-modules.ps1:24-47

GitHub

> Used 7 references

Repo not indexed for semantic code search

Do you wish to index the repo for semantic code search?

Based on the retrieved information, here's a PowerShell method that inserts a JSON file to a CosmosDB collection using the CosmosDB PowerShell module:

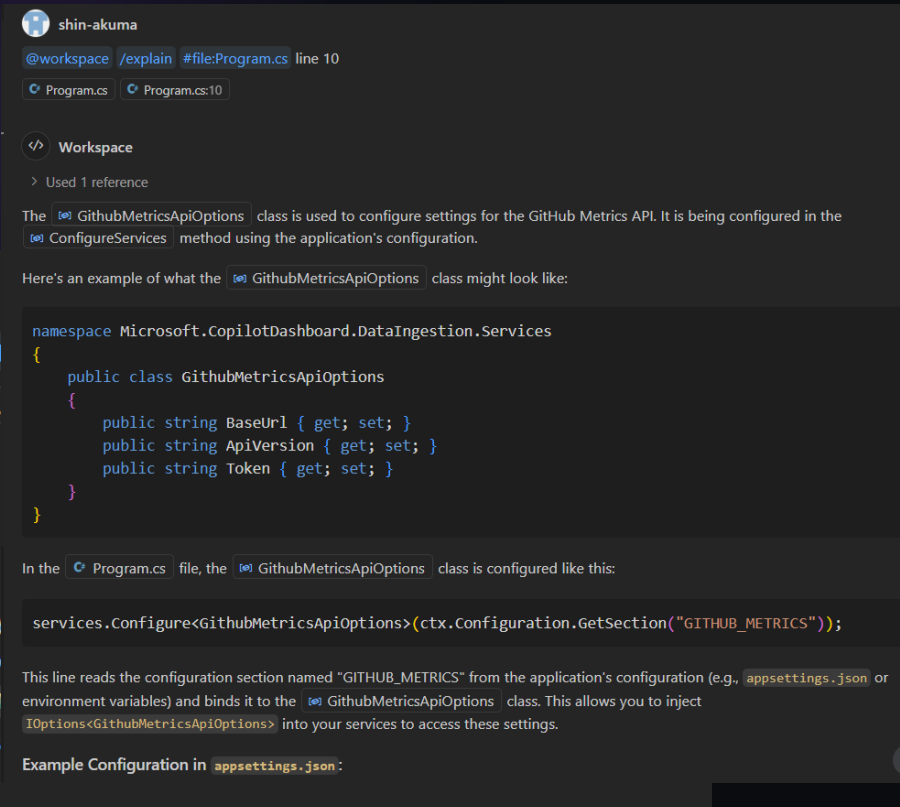
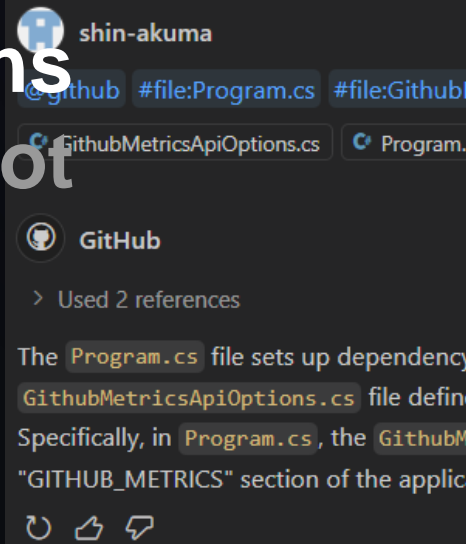
```
function Insert-JsonToCosmosDb (
    param (
        [string]$cosmosDbEndpoint,
        [string]$cosmosDbKey,
        [string]$databaseName,
        [string]$collectionName,
        [string]$jsonFilePath
    )
    # Install the Az.CosmosDB
    if (-not (Get-Module -List
        Install-Module -Name Az.CosmosDB -CurrentUser
    ))
    # Import the Az.CosmosDB module
    Import-Module Az.CosmosDB
    # Read the JSON file
    $jsonContent = Get-Content $jsonFilePath -Raw
    # Convert JSON content to a PowerShell object
    $jsonObject = $jsonContent | ConvertFrom-Json

```



Techniques

Start with clear instructions with Copilot



Techniques

Break the task down with Copilot

Write a function to generate a 10 by 10 grid of letters.

Write a function to find all words in a grid of letters, given a list of valid words.

Write a function that uses the previous functions to generate a 10 by 10 grid of letters that contains at least 10 words.

Update the previous function to print the grid of letters and 10 random words from the grid.



Role Prompting

Use copilot to assume a role during a task

Focus to improves results

Improves results by narrowing the AI's focus

Enhanced Results



abelberhane

I want you to act as a front end developer coach. I will provide some questions and the task that I have to complete and would like for you to walk me through the process before our sprint due date. This could involve offering advice on various topics, such as utilizing the react framework or handling API calls. My first request is to help setup my development environment.



GitHub Copilot

Sure, I can help you with that. What programming language and framework are you planning to use for your front-end development? Also, what operating system are you using?



Techniques

Getting Specific with Copilot

Write a function that tells me if a number is prime

The function should take an integer and return true if the integer is prime

The function should error if the input is not a positive integer



Q & A

Upcoming Sessions

- 1 GitHub Copilot for Developers
- 2 GitHub Copilot for DevOps
- 3 Q&A Session



Thank you!