Microsoft

AI-050: Develop Generative AI Solutions with Azure OpenAI Service

© Copyright Microsoft Corporation. All rights reserved.

Generate code with Azure OpenAI Service

Agenda

- Review how to construct code from natural language
- Complete code and assist in development
- Improve code and fix bugs

© Copyright Microsoft Corporation. All rights reserved.

Learning Objectives

After completing this module, you will be able to:

1. Use natural language prompts to write code
2. Build unit tests and understand complex code with AI models
3. Improve performance and fix bugs for existing code

Use natural language prompts to generate code

- Use prompt engineering to describe what you want clearly
- Break down complex tasks into smaller pieces

Prompt:

`write a function for binary search in python`

```
def binary_search(array, target):
    low = 0
    high = len(array) - 1

    while low <= high:
        mid = (low + high) // 2
        if array[mid] == target:
            return mid
        elif array[mid] < target:
            low = mid + 1
        else:
            high = mid - 1

    return -1
```

© Copyright Microsoft Corporation. All rights reserved.

Change programming language and understand unknown code

Change programming language

```
Convert the following code to C#:
def print_squares(n):
    for i in range(1, n+1):
        print(i**2)
```

```
public void PrintSquares(int n)
{
    for (int i = 1; i <= n; i++)
    {
        Console.WriteLine(i * i);
    }
}
```

Understand unknown code

```
Could you explain what this code is doing?:
fn qzplv(n: i32, m: i32) -> i32 {
    if n == 0 {
        return m + 1;
    } else if m == 0 {
        return qzplv(n - 1, 1);
    } else {
        return qzplv(n - 1, qzplv(n, m - 1));
    }
}
```

```
Sure, I'd be happy to explain what this code is
doing!

This code is implementing the Ackermann
function,...
```

© Copyright Microsoft Corporation. All rights reserved.

Complete partial code

- Writing your prompt with the most amount of information available will be the most accurate
- If enough code is given to determine the programming language, the model will use that language

Prompt:

```
complete the following function
"""
# calculate the average of the numbers
# in an array, but only if they're even
def
```

```python
# calculate the average of the even
# numbers in an array
def calculate_even_average(numbers):
    total = 0
    count = 0
    for number in numbers:
        if number % 2 == 0:
            total += number
            count += 1
    if count == 0:
        return 0  # avoid division by zero
    else:
        return total / count
```

© Copyright Microsoft Corporation. All rights reserved.

Write unit tests

Prompt:

```
write three unit tests for this function
"""
def binary_search(arr, target):
    left, right = 0, len(arr) - 1
    while left <= right:
        mid = (left + right) // 2
        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            left = mid + 1
        else:
            right = mid - 1
    return -1
```

```
def test_binary_search():
    # Test case 1: Target element is in the middle of the
    # array
    arr1 = [1, 2, 3, 4, 5, 6]
    target1 = 4
    assert binary_search(arr1, target1) == 3

    # Test case 2: Target element is at the beginning of
    # the array
    arr2 = [1, 2, 3, 4, 5, 6]
    target2 = 1
    assert binary_search(arr2, target2) == 0

    # Test case 3: Target element is not in the array
    arr3 = [1, 2, 3, 4, 5, 6]
    target3 = 7
    assert binary_search(arr3, target3) == -1
```

© Copyright Microsoft Corporation. All rights reserved.

Fix bugs in your code

- Help identify and fix bugs quickly
- Provide an explanation of the issues

Prompt:

```
This function isn’t working for me. Fix
the bugs in this function
"""
def calculate_average(numbers):
    total = 0
    for i in range(len(numbers)):
        number = numbers[i]
        total += number
    average = total
    rerun averave
```

```
def calculate_average(numbers):
    total = 0
    for i in range(len(numbers)):
        number = numbers[i]
        total += number
    average = total / len(numbers)
    return average
```

There were a few issues with the original code:

1. The variable "average" was assigned the value of "total" instead of the actual average.
2. There was a typo in the last line - "rerun" should be "return".
3. The average was not actually calculated - the corrected code divides the total by the length of the "numbers" list to get the average.

© Copyright Microsoft Corporation. All rights reserved.

Improve performance

- Identify where improvements can be made
- Models able to optimize and refactor

Prompt:

```
Can this function be more efficient?
"""
def sum_of_n(n):
    result = 0
    for i in range(1, n+1):
        result += i
    return result
```

We can improve this function it by using a mathematical formula instead.

Here's the more efficient version:

```
def sum_of_n(n):
    return n * (n + 1) // 2
```

© Copyright Microsoft Corporation. All rights reserved.

Exercise: Generate and improve code with Azure OpenAI Service

Use the hosted lab environment if provided, or view the lab instructions at the link below:

https://aka.ms/mslearn-openai-code-gen

© Copyright Microsoft Corporation. All rights reserved.

Knowledge check

1 **What are some benefits of using Azure OpenAI to generate code?**

☑ Increase in efficiency and productivity

☐ Increase in bugs and readability

☐ Increase in time spent coding

2 **What is the purpose of providing more context to the Azure OpenAI model when completing partial code?**

☐ Providing more context makes the model less accurate

☐ Providing more context doesn’t affect the accuracy of the model

☑ The more context provided to the model, the more accurate the response likely is

3 **What is an example of how Azure OpenAI models can change coding language?**

☐ Azure OpenAI models can only translate code from one language to another if the code is written in C#

☑ If you have code in one language, but need it in another, Azure OpenAI can translate that for you in several languages

☐ Azure OpenAI models can only translate code from Python to C# or Java

Learning Recap

In this module, we:

Generated code from natural language prompts

Explored using AI models to fix bugs, understand code, and complete partial code

Discovered techniques to improve the development process, including writing unit tests and refactoring functions

© Copyright Microsoft Corporation. All rights reserved.

Resources

Generate code with Azure OpenAI Service

https://aka.ms/mslearn-code-openai

© Copyright Microsoft Corporation. All rights reserved.

Microsoft

Thank you.

© Copyright Microsoft Corporation. All rights reserved.