



# USING AI AS YOUR CO-DEVELOPER

---

**JOSEPH WRIGHT**

*Wright4i.com*



# OBJECTIVES

1. Understanding AI
2. Talking to AI
3. Using AI in Practice



*KNOWLEDGE IS POWER*

# WHAT IS AI REALLY?

## Facts

- AI is not **magic**
- AI won't **replace you**
- AI cannot **create new** concepts
- AI is a **tool** that needs **oversight**
- AI is **only as good** as the **data**
- AI is evolving

## Key Concepts of AI

### Narrow AI

#### Weak AI

Speech Recognition

Image Classification

Language Translation

### General AI

#### AGI

Wide range of tasks / roles

Understands

Innovates

Adapts

### Machine Learning

#### ML

Algorithm-driven

Learns from data collected

Improves over time / generations

### Deep Learning

#### DL

Subset of ML

Neural Networks

Complex Patterns

### Large Language Model

#### LLM

Natural language processing

Sequences of Text

Massive Datasets

Various Tasks





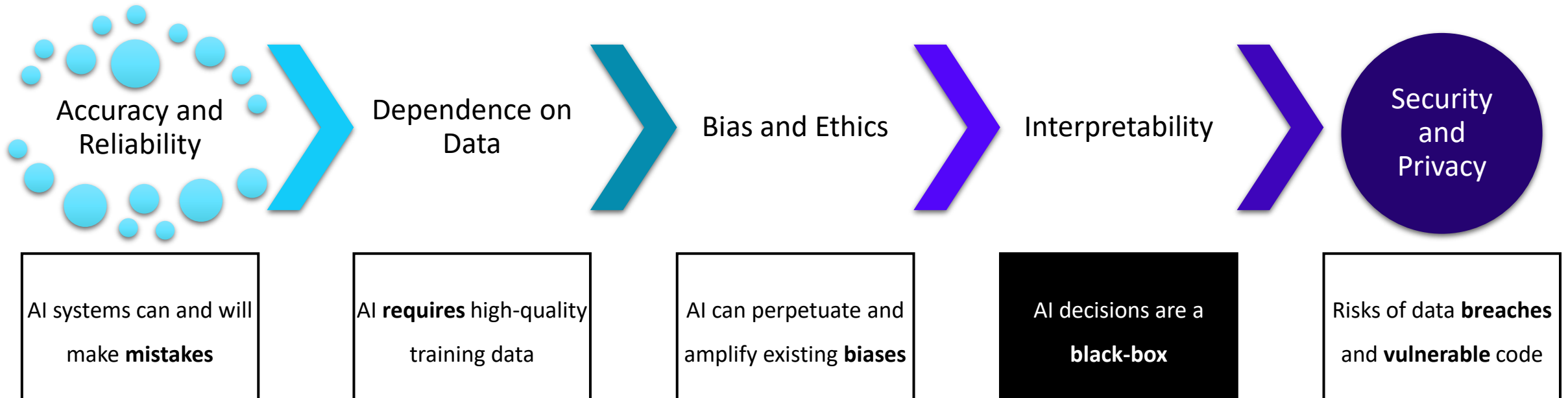
## A CASE STUDY: AMAZON FRESH

### **Amazon struggles with its checkout-less grocery stores with “Just Walk Out” technology.**

The technology allowed customers to skip checkout altogether...

Despite advanced AI and Machine Learning technology, human intervention remains critical. More than **1,000 people** in India were employed to monitor and label the “*fully automated*” system while Amazon continues to improve.

# LIMITATIONS OF AI



# EVALUATING THE OUTPUT

- Code generated from an AI cannot be trusted and it does not absolve you from **personal responsibility**.
- All code generated must be evaluated. It needs to be code up to your coding **standards**.
- Use AI to save you from **tedium**. Repetitive tasks, research intensive time-sinks, and to gap-fill holes in your knowledge.
- Learn from the code being provided. Try to understand it. Asks questions and read the explanations on how it works. **Don't run code you do not understand.**

## *Example*

ChatGPT 4o Prompt:

can I get a sub-procedure that centers text by padding it on the left and right with spaces. I'm going to pass it in a string called sTextToCenter and the total length I want returned as iReturnLength. The actual return length can be a 256 varchar.



Here's an RPG subroutine to center a given text string by padding it on both the left and right sides with spaces. The procedure takes the text to center (^ sTextToCenter`) and the total length of the resulting string (^ iReturnLength`), and returns the centered string as a 256 character varchar.

## THE FIRST TAKE

rpg

```
**FREE
dcl-pr CenterText varchar(256) extpgm('CenterText');
  sTextToCenter varchar(256) const;
  iReturnLength int(10) const;
end-pr;

dcl-proc CenterText export;
  dcl-pi *n varchar(256);
    sTextToCenter varchar(256) const;
    iReturnLength int(10) const;
  end-pi;

  dcl-s sPaddedText varchar(256);
  dcl-s iTextLength int(10);
  dcl-s iTotalPadding int(10);
  dcl-s iLeftPadding int(10);
  dcl-s iRightPadding int(10);
```

```
// Calculate the length of the input text
iTextLength = %len(%trim(sTextToCenter));


// Calculate total padding needed
iTotalPadding = iReturnLength - iTextLength;

// If the input text is longer than the return length, truncate it
if iTotalPadding < 0;
  return %subst(sTextToCenter: 1: iReturnLength);
endif;

// Calculate left and right padding
iLeftPadding = %int(iTotalPadding / 2);
iRightPadding = iTotalPadding - iLeftPadding;

// Construct the padded text
sPaddedText = %trimr(%char(%xlate(' ' : %xlate(' ' : sTextToCenter))) + %char(%xlate(' ' : sTextToCenter)));

// Return the padded text
return %trimr(%char(%xlate(' ' : sPaddedText))) + %char(%xlate(' ' : sPaddedText)));
end-proc;
```





Can you do that using embedded SQL to do the left and right padding?



Certainly! Below is the RPG subroutine using embedded SQL to achieve the left and right padding for centering the text.

```
rpg

**FREE
dcl-pr CenterText varchar(256) extpgm('CenterText');
  sTextToCenter varchar(256) const;
  iReturnLength int(10) const;
end-pr;

dcl-proc CenterText export;
  dcl-pi *n varchar(256);
    sTextToCenter varchar(256) const;
    iReturnLength int(10) const;
  end-pi;

  dcl-s sCenteredText varchar(256);
  dcl-s iTextLength int(10);
  dcl-s iTotalPadding int(10);
  dcl-s iLeftPadding int(10);
  dcl-s iRightPadding int(10);
```

BETTER?

```
// Calculate total padding needed
iTotalPadding = iReturnLength - iTextLength;

// If the input text is longer than the return length, truncate it
if iTotalPadding < 0;
  return %subst(sTextToCenter: 1: iReturnLength);
endif;

// Calculate left and right padding
iLeftPadding = %int(iTotalPadding / 2);
iRightPadding = iTotalPadding - iLeftPadding;

// Use SQL to pad the text
exec sql
  set :sCenteredText = trim(:sTextToCenter)
                        || repeat(' ', :iRightPadding)
                        || repeat(' ', :iLeftPadding);

// Return the padded text
return %trim(sCenteredText);
end-proc;
```





“What do you think **RPG** is?”

# DOES AI KNOW RPG?

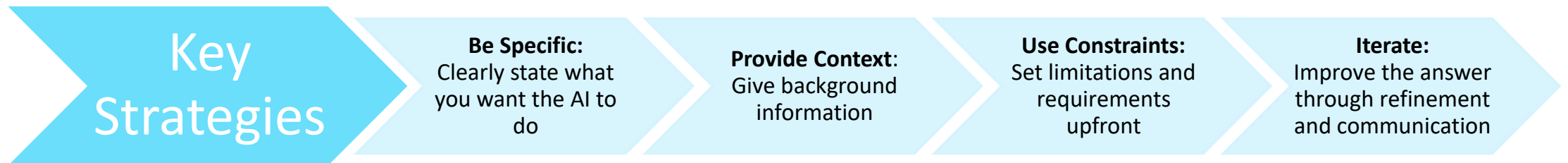
- **Of course!** RPG has been around for a very long time and eventually that thing called the **internet** was invented and we even posted a few articles online about RPG too!
- **Not all AI Models are the same.** The data they were trained on and the number of parameters (which can be thought of as the "complexity" or "capacity" of the model) vary widely.
- The largest AI models like **OpenAI's ChatGPT**, **Google's Bard/Gemini/Gemma**, and **Microsoft's Copilot** all have a decent grasp on RPG the language, once you **coax** it out of them.

# PROMPT ENGINEERING

Prompt engineering is the process of designing and refining your requests to get the most useful and accurate responses from an AI.

## Importance

- Much like using a Search Engine how you ask a question will give you very different results.
- A well-written prompt can make a huge difference in the type of code you receive from an AI, especially with RPG.





# YOUR IDEAL COWORKER

## Prompt

*"You are an expert in IBM's RPG programming language with a deep understanding of fully-free RPG code and embedded SQL. Your goal is to assist in writing, reviewing, and refining RPG code with best practices. When providing code examples, ensure they are in fully-free format and include embedded SQL where applicable. Please ask for any additional context if needed to provide accurate responses."*

## Breakdown

- **Expertise**: Specifies the AI's role and area of expertise.
- **Goals**: Clearly states what the AI should help with.
- **Format and Requirements**: Indicates the preferred code format and use of embedded SQL.
- **Interactive Element**: Encourages the AI to ask for more context if needed.



# PRACTICAL EXAMPLES

REAL-WORLD USE CASES



# WHAT DOES THIS CODE DO?

## Prompt

What does this code do?

```
**free
ctl-opt dftactgrp(*no);
CallPython('/path/to/hello.py', 'arg1 arg2 arg3');

*inlr = *on;
dcl-proc CallPython;
  dcl-pi ind;
    script varchar(256) const;
    parms varchar(256) const;
  end-pi;

  dcl-pr system extproc('system');
    *n pointer value options(*string);
  end-pr;

  dcl-s command varchar(512);
  dcl-s logId int(10);
  dcl-s resultCode int(10);

exec sql
  SELECT ID
  INTO :logId
  FROM FINAL TABLE (
    INSERT INTO PYTHON_LOGS (SCRIPT_PATH, PARAMETERS)
    VALUES (:script, :parms)
  );
```

Continued...

## Response



The provided RPG code demonstrates the calling of a Python script from an RPG program and logging the execution details to a database table. Here's a detailed breakdown of what the code does:

### Overview

The main objective of this program is to call a Python script (`hello.py`) with specified arguments (`arg1 arg2 arg3`) and log the execution details (script path, parameters, and result code) in a database table named `PYTHON_LOGS`.

### Code Breakdown

rpg

Copy code

```
**free
ctl-opt dftactgrp(*no);
CallPython('/path/to/hello.py', 'arg1 arg2 arg3');

*inlr = *on;
```

- `ctl-opt dftactgrp(*no);`: This control option specifies that the program does not run in the default activation group, which allows for better modularity and resource management.
- `CallPython('/path/to/hello.py', 'arg1 arg2 arg3');`: This line calls the `CallPython` procedure with the specified script path and arguments.
- `*inlr = *on;`: This sets the last record indicator to `*on`, indicating the end of the program.

Continued...

# HOW DO I DO X?

## Overload a Prototype?

How do I overload a prototype?



To overload a prototype in ILE RPG, you use the `OVERLOAD` keyword, which defines a list of other prototypes that can be called by using the name of the prototype with the `OVERLOAD` keyword. When you call the prototype with the `OVERLOAD` keyword, the compiler determines which candidate prototype to call based on the parameters specified.

Here is an example to demonstrate how to overload prototypes in ILE RPG:

rpg Copy code

```
**FREE
// Define individual prototypes
dcl-pr format_date varchar(100);
      dateParm date(*ISO) const;
end-pr;

dcl-pr format_time varchar(100);
      timeParm time(*ISO) const;
end-pr;

dcl-pr format_message varchar(100);
      msgid char(7) const;
      replacement_text varchar(100) const options(*nopass);
end-pr;

// Define the overloaded prototype
dcl-pr format varchar(100) overload(format_time: format_date: format_message);
end-pr;

// Variables to store results
dcl-s result varchar(100);

// Example calls to the overloaded prototype
result = format(%date());           // Calls format_date
result = format(%time());           // Calls format_time
result = format('MSG0100': 'Filename'); // Calls format_message

*inlr = *on;
return;
```

Continued...

## Parse JSON in DB2?

how do I parse json in my db2 sql?



To parse JSON in DB2 SQL, you can utilize the built-in functions available in DB2 for i. Here are some key functions and examples that demonstrate how to work with JSON data:

### Key Functions:

1. **JSON\_QUERY**: Extracts an SQL/JSON value from a JSON string.
2. **JSON\_TABLE**: Converts a JSON document into a relational table.
3. **JSON\_VALUE**: Extracts a scalar value from a JSON document.
4. **JSON\_ARRAYAGG**: Aggregates a set of values into a JSON array.
5. **JSON\_OBJECTAGG**: Aggregates a set of key-value pairs into a JSON object.

### Examples:

#### 1. Extract a JSON object

sql Copy code

```
VALUES JSON_QUERY('{ "id": "701", "name": { "first": "John", "last": "Doe" } }', '$.name');
```

This query returns:

json Copy code

```
{ "first": "John", "last": "Doe" }
```

#### 2. Convert a JSON document to a relational table

sql Copy code

```
SELECT U.id, U.first_name, U.last_name, U.phone_type, U.phone_number
FROM EMPLOYEE_TABLE E,
     JSON_TABLE(E.jsondoc, 'lax
COLUMNS(
```

Continued...

# FORMAT MY CODE, PLEASE.

## Prompt

Can you format my code?

```
select
column_name,
data_type,
length,
numeric_scale,
column_text
from
qsys2.syscolumns
where
system_table_schema = 'QGPL'
and system_table_name = 'SAMPLE';
```



Here is your formatted SQL code:

sql

Copy code

```
SELECT
    column_name,
    data_type,
    length,
    numeric_scale,
    column_text
FROM
    qsys2.syscolumns
WHERE
    system_table_schema = 'QGPL'
    AND system_table_name = 'SAMPLE';
```

## Iterate

Can you join in the table description and give the fields friendlier names for a report?



Sure, here's an updated version of your query that includes joining the table description and renaming the fields for a more readable report:

sql

Copy code

```
SELECT
    cols.column_name AS "Column Name",
    cols.data_type AS "Data Type",
    cols.length AS "Length",
    cols.numeric_scale AS "Numeric Scale",
    cols.column_text AS "Column Text",
    tbl.table_text AS "Table Description"
FROM
    qsys2.syscolumns cols
JOIN
    qsys2.systables tbl
ON
    cols.system_table_schema = tbl.system_table_schema
    AND cols.system_table_name = tbl.system_table_name
WHERE
    cols.system_table_schema = 'QGPL'
    AND cols.system_table_name = 'SAMPLE';
```

# OCR

A picture embedded in an email:

Task Name	July		August			
	21	28	4	11	18	25
Task Name	Assignme	Assignme	Assignme	Assignme	Assignme	Assignme
<b>Total</b>	<b>34</b>	<b>40</b>	<b>13</b>	<b>3</b>	<b>4</b>	<b>3</b>
Conduct / Attend Project Status Meetings	2	2	2	2	2	
Deployment to Production				1		
Develop new API JSON map to handle customer specific codes	1	10				
Develop new API push to Target and associated trigger points	1	10				
Develop new Auxiliary tables to warehouse customer specific hold codes	8					
Develop new entry point for customer specific hold codes within DECON	15					
Post Deployment Support				1	3	
Prepare Closure Documentation						3
Test API push to customer with new JSON map		4	4			
Test new API JSON map and associated trigger points		8	8			
Test new DECON enhancements for customer specific exception processes/Database	8	8				
	<b>34</b>	<b>40</b>	<b>13</b>	<b>3</b>	<b>4</b>	<b>3</b>

Can you turn this into an Excel file?

Response

Task_Assignment.xlsx - Saved to this PC							
File Home Insert Page Layout Formulas Data Review View Automate Help							
H15 : X Y fX							
	A	B	C	D	E	F	G
1	Task Name	July 21	July 28	August 4	August 11	August 18	August 25
2	Total	34	40	13	3	4	3
3	Conduct / Attend Project Status Meetings	2	2	2	2	2	
4	Deployment to Production				1		
5	Develop new API JSON map to handle customer specific codes	1	10				
6	Develop new API push to Target and associated trigger points	1	10				
7	Develop new Auxiliary tables to warehouse customer specific hold codes	8					
8	Develop new entry point for customer hold codes within DECON	15					
9	Post Deployment Support				1	3	
10	Prepare Closure Documentation						3
11	Test API push to customer with new JSON map		4	3			
12	Test new API JSON map and associated trigger points		8	8			
13	Test new DECON enhancements for customer exception processes	8	8				
14							



# PROJECT REQUIREMENTS

## Requirements Document for Shopify API Project

### Project Overview

The objective of this project is to develop a system that integrates with the Shopify API to manage order fulfillment efficiently. The system will download open orders from the Shopify store into a new database table named SHOPIFY\_ORDERS, develop an RF Scanner application for processing orders, and update the order status on Shopify by marking them as fulfilled and assigning the tracking number.

### Functional Requirements

#### Data Integration

- Connect to the Shopify API to retrieve open orders.
- Store the retrieved orders into a new table named SHOPIFY\_ORDERS in the database.
- Ensure that the data is accurately mapped and stored in the database.
- Schedule regular intervals to download new open orders and update the SHOPIFY\_ORDERS table.

#### RF Scanner Application

- Develop an RF Scanner application that allows the user to input the Order ID and UPS tracking number.
- Validate the input to ensure the Order ID exists in the SHOPIFY\_ORDERS table and the UPS tracking number follows the correct format.
- Once the Order ID and UPS tracking number are inputted, the application should update the respective order in the SHOPIFY\_ORDERS table with the tracking number and mark it as fulfilled.

#### Shopify Order Update

- Connect to the Shopify API to update the order status.
- Mark the order as fulfilled.
- Assign the provided UPS tracking number to the order.
- Ensure the update is successful and handle any errors or exceptions.

## Prompt



Shopify\_API\_Project\_Requirements.docx  
Document



I've been given this requirements document. I want to program in fully-free RPG, embedded SQL, and Python for this project. Can you give me a step by step breakdown on what I need to do and the best language to use for each to fulfill the requirements?

# REFERENCE MATERIAL

## Reference



### Project Overview

- **Objective:** Integrate with the Shopify API to manage order fulfillment efficiently.
- **Key Functionalities:**
  1. Data Integration with Shopify API.
  2. RF Scanner Application.
  3. Shopify Order Update.

### Recommended Languages and Key Functions

- **Python:**
  - **Data Integration:**
    - Connect to the Shopify API to retrieve open orders.
    - Schedule regular intervals to download new open orders.
    - Handle API responses and errors.
  - **Shopify Order Update:**
    - Update order status on Shopify.
    - Assign UPS tracking numbers to orders.
- **RPG with Embedded SQL:**
  - **Data Integration:**
    - Store retrieved orders into the `SHOPIFY\_ORDERS` table.
  - **RF Scanner Application:**
    - Input and validate Order ID and UPS tracking number.
    - Update orders in the `SHOPIFY\_ORDERS` table with tracking numbers and fulfillment status.

## Build

Can you give me the API endpoints I'd need to hit for this project and a code example?



### Shopify API Endpoints

1. **Retrieve Open Orders:**
  - **Endpoint:** `GET /admin/api/2021-04/orders.json`
  - **Parameters:** `status=open`
  - **Purpose:** Retrieve all open orders from the Shopify store.
2. **Update Order Status:**
  - **Endpoint:** `PUT /admin/api/2021-04/orders/{order\_id}.json`
  - **Purpose:** Update the order status to fulfilled and assign a UPS tracking number.

### Code Example in Python

#### Retrieve Open Orders

```
python
import requests
import json

# Shopify API credentials
SHOPIFY_API_KEY = 'your_api_key'
SHOPIFY_PASSWORD = 'your_password'
SHOPIFY_STORE_NAME = 'your_store_name'

# Endpoint for retrieving open orders
url = f'https://{SHOPIFY_API_KEY}:{SHOPIFY_PASSWORD}@{SHOPIFY_STORE_NAME}.myshopify.com/admin/orders.json?status=open'
params = {
    'status': 'open'
}

# Request to retrieve open orders
```

# DOCUMENT THIS CODE

## Prompt

Can you document this code?

```
ctl-opt debug option(*NODEBUGIO : *SRCSTMT : *NOUNREF :
*NOSHOWCPY)
    dftactgrp(*NO) actgrp(*caller) ;

dcl-s input varchar(30) inz;
dcl-s total_length packed(2);
dcl-s position packed(2) inz(1);
dcl-s word_start packed(2) inz(0);
dcl-s word_end packed(2) inz(0);
dcl-s unique_letters packed(2) inz(0);
dcl-s word_length packed(2) inz(0);
dcl-s word like(input) inz;
dcl-s dupe_letters like(input) inz;
dcl-s counter packed(2) inz(1);
dcl-s result like(input) inz;

input = 'Automotive Repair Shop #5 Utah';

total_length = %len(input);

dow position <= total_length;

    if
%check('abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
XYZ':
        %subst(input:position:1)) <> 0 or
        position = total_length;

        if word_start > 0;
            if position = total_length;
                word_end = position;
```

## Response

Here is the documented version of your RPG code:

```
rpg
Copy code

**FREE
// Title: Unique Letter Counter in Words
// Program Name: WordUniqueLetterCounter.rpgle
// Date: 2024-07-22
// Author: Your Name
// Description: This program processes a given input string, counts the unique letters in it, and returns the result.

ctl-opt debug option(*NODEBUGIO : *SRCSTMT : *NOUNREF : *NOSHOWCPY)
    dftactgrp(*NO) actgrp(*caller);

// Declare variables
dcl-s input varchar(30) inz;
dcl-s totalLength packed(2);
dcl-s position packed(2) inz(1);
dcl-s wordStart packed(2) inz(0);
dcl-s wordEnd packed(2) inz(0);
dcl-s uniqueLetters packed(2) inz(0);
dcl-s wordLength packed(2) inz(0);
dcl-s word like(input) inz;
dcl-s dupeLetters like(input) inz;
dcl-s counter packed(2) inz(1);
dcl-s result like(input) inz;

// Initialize input string
input = 'Automotive Repair Shop #5 Utah';

// Calculate total length of the input string
totalLength = %len(input);

dow position <= totalLength;
    // Check if the current character is not a letter or end of the input string is reached
    if %check('abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ': %subst(input:position:1)) <> 0 or
        // If a word start is marked
        if wordStart > 0;
            // Mark the end of the word
            if position = totalLength;
                wordEnd = position;
            else;
```

# FINAL TIPS & TAKEAWAYS

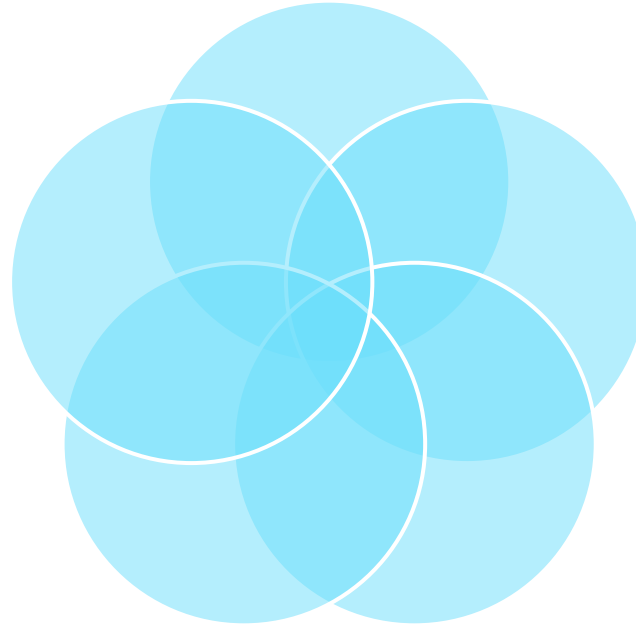
AI is a powerful  
tool, but it has its  
limits.

Stay updated!  
AI is moving  
faster than  
ever.

Take the time  
to craft  
effective  
prompts.

Collaborate  
with the AI -  
Iterate & Build.

Review the  
code,  
understand it,  
learn from it.



The background of the slide is a dark, textured surface with a subtle, abstract pattern of light gray lines and dots, resembling a molecular structure or a network diagram. The text is centered and rendered in a clean, white, sans-serif font.

# THANK YOU!

JOSEPH WRIGHT

*Wright4i.com*