

Experiment-7

Execute Python Code Cleaning

Date: _____

AIM

Execute python code cleaning.

PROCEDURE

Step-1: Formatting and Style.

Step-2: Naming Conventions

Step-3: Refactoring

Step-4: Optimization

Step-5: Error Handling

Step-6: Documentation

Step-7: Testing

Step-8: Use Linters and Formatters

Step-9: Code Reviews

Step-10: Continuous Integration (CI)

open browser > search openAI > click on try chatgpt >
Login using your credentials.

Prompt 1:

Generate a Non-Pythonic code that prints numbers from 1 to 5
output:-



Here's a non-Pythonic approach to print numbers from 1 to 5. This code avoids using Pythonic constructs such as list comprehensions, built-in functions, and idiomatic looping:

python

Copy code

```
# Non-Pythonic code to print numbers from 1 to 5
```

```
i = 1
while i <= 5:
    if i == 1:
        print(1)
    elif i == 2:
        print(2)
    elif i == 3:
        print(3)
    elif i == 4:
        print(4)
    elif i == 5:
        print(5)
    i = i + 1
```

Characteristics of Non-Pythonic Code:

- **Repetitive Conditions:** Multiple `if` or `elif` statements are used to check each possible value of `i`, which is verbose and repetitive.
- **Manual Loop Control:** The loop increments the variable manually and uses explicit condition checks.
- **Lack of Idiomatic Constructs:** Avoids using Pythonic features like list comprehensions, `range()`, or built-in functions for simplicity and clarity.

Prompt 2:

Generate a Pythonic code for the above code which includes:

- 1) variable tricks
- 2) list manipulation (initialization, slicing)
- 3) dealing with functions
- 4) explicit code

Output:-

Here's a Pythonic version of the code that prints numbers from 1 to 5, incorporating all the specified elements:

python

Copy code

```
# Pythonic code to print numbers from 1 to 5

# Variable trick: Define start and end values
start, end = 1, 5

# Function to print a list of numbers
def print_numbers(start, end):
    # List manipulation: Initialize list and slice
    numbers = list(range(start, end + 1)) # Create a list of numbers from start to end
    sliced_numbers = numbers[:] # Slice the entire list (demonstrates list slicing)
    for num in sliced_numbers: # Iterate through the sliced list
        print(num)

# Explicitly call the function with start and end
print_numbers(start, end)
```

Breakdown:

1. **Variable tricks:** `start, end = 1, 5` assigns the beginning and ending numbers in one line.
2. **List manipulation:** `numbers = list(range(start, end + 1))` creates a list of numbers from 1 to 5. The slicing operation `sliced_numbers = numbers[:]` demonstrates list slicing, even though it's not strictly necessary here.
3. **Dealing with functions:** The logic is contained in the `print_numbers()` function, making it reusable and clear.
4. **Explicit code:** The function is explicitly called with `print_numbers(start, end)`, ensuring the code's purpose and execution are clear.

This version is both concise and adheres to Pythonic principles.

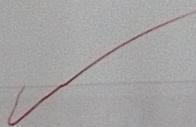
Execution of both Non-Pythonic and Pythonic code produces the same output.

Non-Pythonic code and its execution

```
main.py
1 # Non-Pythonic code to print numbers from 1 to 5
2
3 i = 1
4 while i <= 5:
5     if i == 1:
6         print(1)
7     elif i == 2:
8         print(2)
9     elif i == 3:
10        print(3)
11    elif i == 4:
12        print(4)
13    elif i == 5:
14        print(5)
15    i = i + 1
16
```

Output

```
1
2
3
4
5
--- Code Execution Successful ---
```



OUTPUTPythonic code and its execution

The screenshot shows a Jupyter Notebook interface with a code cell titled "main.py". The code prints numbers from 1 to 5 using a range and a for loop. It also defines a function "print_numbers" that prints the numbers in the list. The output cell shows the numbers 1 through 5, followed by a success message. A red checkmark is drawn over the handwritten note below the screenshot.

```
main.py
1 # Pythonic code to print numbers from 1 to 5
2
3 # Variable trick: Define start and end using a tuple
4 start, end = 1, 5
5
6 # Function to print numbers
7 def print_numbers(start, end):
8     # List manipulation: Initialize a list with a range and slice it
9     numbers = list(range(start, end + 1)) # List initialization
10    for num in numbers[:]: # Slicing the list (though not needed, included for demonstration)
11        print(num)
12
13 # Explicitly call the function with start and end
14 print_numbers(start, end)
15
```

1
2
3
4
5
--- Code Execution Successful ---
6
7
8
9
10
11
12
13
14
15

How Pythonic code is executed?

Execution of main.py in Jupyter Notebook

VIVA QUESTIONS

1. What are PEP 8 guidelines, and why are they important in Python programming?

Ans. PEP 8 is the Python Enhancement Proposal that provides coding conventions for writing readable Python code. It is important because it ensures consistency, improves code readability, and makes it easier for developers to collaborate.

2. Can you explain the concept of code refactoring? Provide an example of how you might refactor a piece of Python code to improve its readability or performance?

Ans. Code refactoring involves restructuring existing code without changing its external behaviour. The goal is to improve readability, reduce complexity and enhance maintainability. Ex:- Replacing redundant loops with list comprehensions in Python to make the code shorter and faster.

3. What tools do you use for linting and formatting Python code, and how do they help in code cleaning?

Ans. Common tools include `pylint`, `flake8` and `black`. Linting helps catch syntax errors, code style issues and potential bugs. Formatting tools like `black` ensure code follows consistent formatting rules, making it clean and readable.

4. How do you handle error and exception management in Python to ensure robust code?

Ans. Python uses `try`, `except`, and `finally` blocks for handling exceptions.

This ensures that the program doesn't crash unexpectedly and provides meaningful error messages. It also allows for graceful recovery from errors.

5. Explain the difference between mutable and immutable objects in Python and how this knowledge impacts the way you write clean code?

Ans. Mutable objects (e.g., lists, dictionaries) can be modified after creation, while immutable objects (e.g., tuples, strings) cannot be changed. Knowing this helps avoid unintended side effects when passing objects to functions, making the code more predictable and reliable.

Experiment-8

Enact Cross-Lingual Conversations

Date: _____

AIM

Enact Cross-Lingual Conversations.

PROCEDURE

Step-1: Identify Languages Involved

Step-2: Set Up Translation Tools

Step-3: Understand the Context

Step-4: Start the Conversation

Step-5: Use ChatGPT for Translation Assistance

SOURCE CODE

open browser > search openAI > click on try chatgpt > Login using your credentials.

Prompt to generate "Translation Code" :

Generate a Python script that uses the `googletrans` library to translate a series of conversation snippets into English. The script should include the following:

1. Import the `Translator` class from the `googletrans` library.
2. Define a function called `translate_text` that takes a text string and a destination language code as parameters. This function should return the translated text.
3. Create a list of conversation snippets, each containing a text string and its corresponding language code.
4. Loop through the list and print each original conversation snippet along with its translation in English if it's not already in English.

Procedure :

1. Search for "Google Colab" on a search engine.
2. Click on "Colab" from the search results.
3. When a pop-up appears, click on "New Notebook".
4. Give the notebook a title.
5. Run the command "~~pip install googletrans==4.0.0-rc1~~" in the cell.
6. After it runs successfully, click on the +code to create a new cell, paste the generated code, and execute it.
7. It will provide the original text and its translated version (in English) as output.
8. Finally, save the result.

40 | week8.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```

Q ✓ pip install googletrans==4.0.0-rc1
(x)   Collecting googletrans==4.0.0-rc1
      Downloading googletrans-4.0.0rc1.tar.gz (20 kB)
      Preparing metadata (setup.py) ... done
C   Collecting httpx==0.13.3 (from googletrans==4.0.0-rc1)
      Downloading httpx-0.13.3-py3-none-any.whl.metadata (25 kB)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from httpx==0.13.3->googletrans==4.0.0-rc1) (2024.8.30)
Collecting hstspreload (from httpx==0.13.3->googletrans==4.0.0-rc1)
      Downloading hstspreload-2024.9.1-py3-none-any.whl.metadata (2.1 kB)
Requirement already satisfied: sniffio in /usr/local/lib/python3.10/dist-packages (from httpx==0.13.3->googletrans==4.0.0-rc1) (1.3.1)
Collecting chardet==3.* (from httpx==0.13.3->googletrans==4.0.0-rc1)
      Downloading chardet-3.0.4-py2.py3-none-any.whl.metadata (3.2 kB)
Collecting idna==2.* (from httpx==0.13.3->googletrans==4.0.0-rc1)
      Downloading idna-2.18-py2.py3-none-any.whl.metadata (9.1 kB)
Collecting rfc3986<2,>=1.3 (from httpx==0.13.3->googletrans==4.0.0-rc1)
      Downloading rfc3986-1.5.0-py2.py3-none-any.whl.metadata (6.5 kB)
Collecting httpcore==0.9.* (from httpx==0.13.3->googletrans==4.0.0-rc1)
      Downloading httpcore-0.9.1-py3-none-any.whl.metadata (4.6 kB)
Collecting h11<0.18,>=0.8 (from httpcore==0.9.*->httpx==0.13.3->googletrans==4.0.0-rc1)
      Downloading h11-0.9.0-py2.py3-none-any.whl.metadata (8.1 kB)
Collecting h2==3.* (from httpcore==0.9.*->httpx==0.13.3->googletrans==4.0.0-rc1)
      Downloading h2-3.2.0-py2.py3-none-any.whl.metadata (32 kB)
Collecting hyperframe<6,>=5.2.0 (from h2==3.*->httpcore==0.9.*->httpx==0.13.3->googletrans==4.0.0-rc1)
      Downloading hyperframe-5.2.0-py2.py3-none-any.whl.metadata (7.2 kB)
Collecting hpack<4,>=3.0 (from h2==3.*->httpcore==0.9.*->httpx==0.13.3->googletrans==4.0.0-rc1)
      Downloading hpack-3.0.0-py2.py3-none-any.whl.metadata (7.0 kB)
Downloading httpx-0.13.3-py3-none-any.whl (55 kB)
      55.1/55.1 kB 2.5 MB/s eta 0:00:00
Downloading chardet-3.0.4-py2.py3-none-any.whl (133 kB)
      133.4/133.4 kB 5.2 MB/s eta 0:00:00
Downloading httpcore-0.9.1-py3-none-any.whl (42 kB)
      42.6/42.6 kB 1.5 MB/s eta 0:00:00
Downloading idna-2.18-py2.py3-none-any.whl (58 kB)
      58.8/58.8 kB 2.7 MB/s eta 0:00:00
Downloading h2-3.2.0-py2.py3-none-any.whl (65 kB)
      65.0/65.0 kB 3.7 MB/s eta 0:00:00
Downloading rfc3986-1.5.0-py2.py3-none-any.whl (31 kB)
Downloading hstspreload-2024.9.1-py3-none-any.whl (1.2 MB)
      1.2/1.2 kB 15.3 MB/s eta 0:00:00
Downloading h11-0.9.0-py2.py3-none-any.whl (53 kB)
      53.6/53.6 kB 2.4 MB/s eta 0:00:00
Downloading hpack-3.0.0-py2.py3-none-any.whl (38 kB)

```

week8.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
Q ① from googletrans import Translator

(x) # Initialize the translator
translation = Translator()

② def translate(text, dest_language):
    """Translate text to the specified destination language."""
    translated = translation.translate(text, dest=dest_language)
    return translated.text

# Conversation setup
conversations = [
    {"text": "Hi there! How's your day going?", "lang": "en"},
    {"text": "¡Hola! Mi día ha sido muy ocupado, pero está bien. ¿Qué has estado haciendo?", "lang": "es"},
    {"text": "J'ai travaillé sur un projet important au travail. Et toi, qu'est-ce que tu as fait de plus intéressant aujourd'hui?", "lang": "fr"},
    {"text": "Ja, ich habe ein wenig Zeit gefunden, um ein Buch zu lesen. Was machst du normalerweise, um dich zu entspannen?", "lang": "de"}]

# Print the conversation
for convo in conversations:
    text = convo["text"]
    lang = convo["lang"]
    print(f"Original ({lang}): {text}")
    if lang != "en":
        translated_text = translate_text(text, "en")
        print(f"Translated to English: {translated_text}")
    print()
```

→ Original (en): Hi there! How's your day going?

Original (es): ¡Hola! Mi día ha sido muy ocupado, pero está bien. ¿Qué has estado haciendo?

Translated to English: Hello! My day has been very busy, but it's fine.What have you been doing?

⟨⟩ Original (fr): J'ai travaillé sur un projet important au travail. Et toi, qu'est-ce que tu as fait de plus intéressant aujourd'hui?

Translated to English: I worked on an important project at work.And you, what did you do more interesting today?

☰ Original (de): Ja, ich habe ein wenig Zeit gefunden, um ein Buch zu lesen. Was machst du normalerweise, um dich zu entspannen?

Translated to English: Yes, I found a little time to read a book.What do you usually do to relax?

✓ 1s completed at 11:53 AM

VIVA QUESTIONS

- How do you ensure accuracy in translation when conducting a cross-lingual conversation?

Ans. Accuracy can be ensured by understanding the context, using reliable translation tools, double-checking ambiguous terms and confirming with native speakers when necessary.

- What are the major challenges of cross-lingual communication and how do you address them?

Ans. Challenges include language barriers, cultural differences and misunderstandings. These can be addressed through active listening, simplified language, seeking clarification and being aware of cultural sensitivities.

- What role does non-verbal communication play in cross-lingual interactions?

Ans. Non-verbal cues such as gestures, facial expressions and body language can help convey meaning when verbal communication is limited or unclear. They often go beyond language barriers.

- How do you handle misunderstandings or miscommunications in a cross-lingual setting?

Ans. Misunderstandings can be managed by staying patient, rephrasing or repeating the message, seeking clarification, and using visual aids to enhance comprehension.

- What ethical considerations do you take into account when translating or interpreting in cross-lingual conversations?

Ans. Ethical considerations include maintaining confidentiality, avoiding personal biases, ensuring cultural respect, and providing accurate and faithful translations without altering the meaning.

Experiment-9

Sketch Scientific Image

Date: _____

AIM

Sketch scientific image.

PROCEDURE

Creating a scientific image using ChatGPT involves several steps, particularly when using tools like DALL-E for image generation.

Step-1: Define the Concept.

Step-2: Detail the Description.

Step-3: Use the Image Generation Tool

Step-4: Generate the Image.

Step-5: Review and Refine

SOURCE CODE

open browser > search openAI > click on try chatgpt >
Login using your credentials .

Procedure :

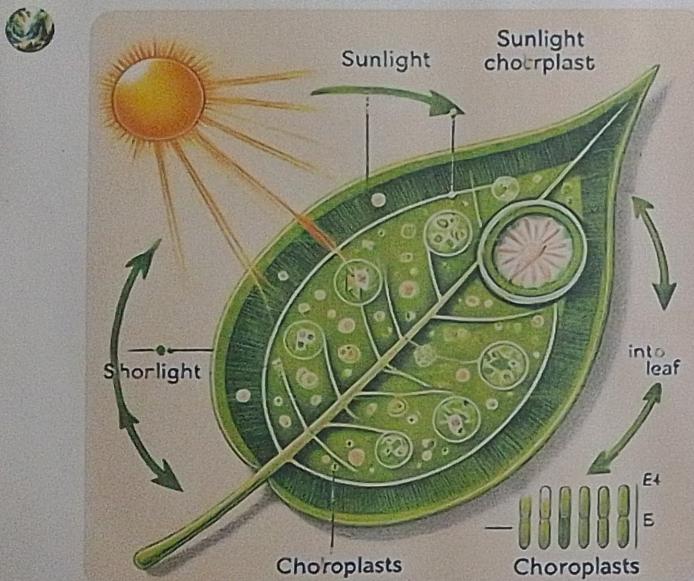
Click on the user icon, then click on "My GPTs", select any image generator tool (DALL-E, image generator etc.), and give a prompt that generates "Scientific Sketches."

USING IMAGE GENERATOR TOOL :-

Prompts to generate scientific sketch of photosynthesis :-

Prompt 1: A diagram illustrating the process of photosynthesis in a plant leaf. Show the chloroplasts within the cells and sunlight striking the leaf. Arrows should depict the flow of energy, specifically sunlight, entering the leaf.

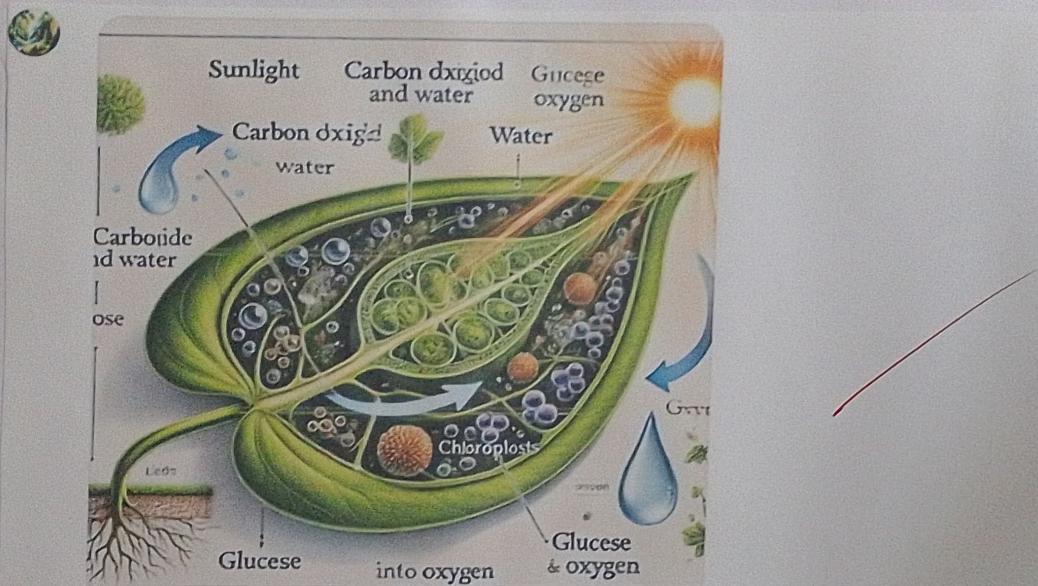
Output (Response):



Here is the diagram illustrating the process of photosynthesis in a plant leaf, showing chloroplasts within the cells and sunlight striking the leaf with arrows depicting the flow of energy. Let me know if you'd like any adjustments!

Prompt 2: In the same diagram, illustrate the conversion of carbon dioxide and water into glucose and oxygen. Show arrows representing the flow of carbon dioxide entering the leaf and oxygen exiting, along with glucose being produced inside the leaf.

Output(response):

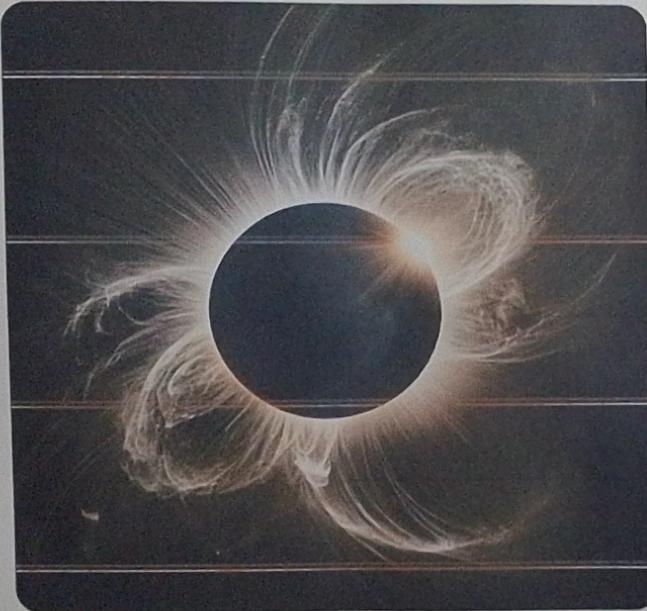


Here is the updated diagram showing the full process of photosynthesis, including the conversion of carbon dioxide and water into glucose and oxygen. Arrows represent the flow of carbon dioxide entering the leaf, oxygen exiting, and glucose being produced inside the leaf. Let me know if you need further adjustments!

Prompts to generate scientific sketch of Solar Eclipse:-

Prompt 1: Create a sketch depicting a solar eclipse from space. Illustrate the Moon passing in front of the Sun, highlighting the darkened Moon against the bright backdrop of the Sun. Show the Sun's corona visibly radiating around the edges of the Moon.

Output (Response):



Here is the sketch of a solar eclipse as seen from space, showing the dark silhouette of the Moon blocking the Sun, with the Sun's corona radiating around the edges. Let me know if you'd like any adjustments!

Prompt 2: In the same diagram, draw a scene of the shadow cast by the Moon as it moves across the surface of the Earth during a solar eclipse. Include details of the shadow's path and how it interacts with the Earth's features below, emphasizing the contrast between light and dark.

output(response) :



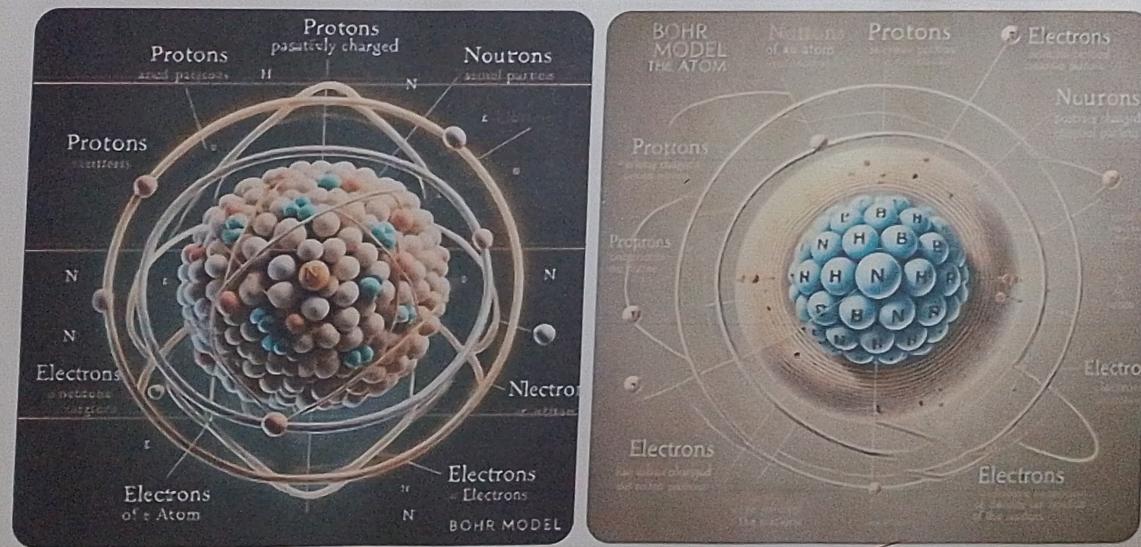
Here's the updated scene, now showing both the solar eclipse in space and the Moon's shadow cast on Earth. The shadow creates a striking contrast with the sunlit areas below, emphasizing the eclipse's path across the planet. Let me know if you'd like further changes!

USING DALL-E TOOL :-

Prompts to generate scientific sketch of Atom Model :-

Prompt 1: Create a diagram of an atom with a nucleus at the center, containing protons and neutrons. The protons and neutrons should be clearly labeled and positioned together inside the nucleus.

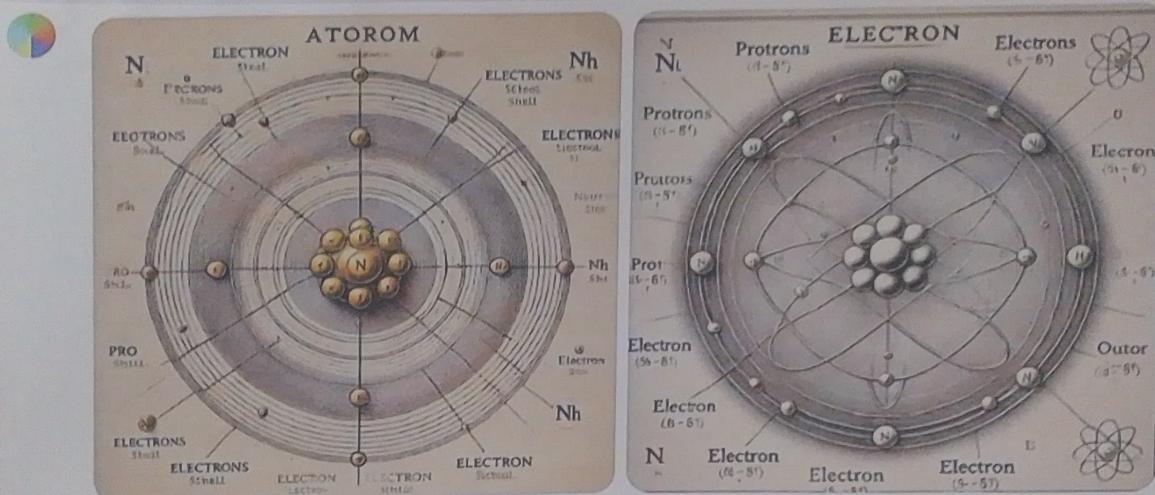
output(response) :



Here are two diagrams of an atom with the nucleus at the center, showing protons and neutrons clearly labeled. Let me know if you'd like any adjustments!

Prompt 2: The diagram should also include electrons orbiting the nucleus in defined shells, representing different energy levels. The electrons should be labeled, with lines indicating their paths around the nucleus. Each shell should be depicted as distinct from one another to show the difference in energy levels.

output(response):



Here are the updated diagrams showing the atom with electrons orbiting the nucleus in defined shells representing different energy levels. Each shell is distinct and labeled, with clear electron paths. Let me know if you need any further adjustments!

VIVA QUESTIONS

1. Can you explain the scientific concept depicted in this image? For ex:- Photosynthesis

Ans. Photosynthesis is the process by which plants convert sunlight, water and carbon dioxide into glucose(energy) and oxygen. This occurs in the chloroplasts, where sunlight powers the conversion of carbon dioxide and water into glucose, releasing oxygen as a by-product.

2. What tools or software did you use to create this image, and why?

Ans. Tools like Open AI's DALL-E are used. These AI-based tools are chosen because they can generate images from text descriptions, making them useful for visualizing abstract or complex ideas.

3. How can scientific images like the one you created enhance understanding of complex concepts?

Ans. Scientific images simplify complex data by presenting it visually, making patterns, relationships and structures easier to comprehend, which aids in better understanding for both experts and the general public.

4. Discuss the importance of color choice in scientific imaging?

Ans. Colors can highlight key areas, differentiate between variables, and improve readability. Proper use of color enhances clarity, emphasizes important details and ensures that the image is accessible to those with color vision deficiencies.

5. What ethical considerations did you take into account while creating and presenting this image?

Ans. Ethical considerations include ensuring accuracy, avoiding manipulation or exaggeration of data, giving credit to original sources and making sure the image does not mislead or misrepresent scientific facts.