**LIST**

**Create a List of Well Names**

# Task: Create a list of 5 well names:

# Example: ['HT1', 'HT2', 'HT3', 'RC1', 'RC2']

**Append a New Well**

# wells = ['HT1', 'HT2']

# Task: Add 'HT3' to the list using append()

**Insert a Well at Index 1**

# wells = ['HT1', 'HT3']

# Task: Insert 'HT2' at position 1

**Remove a Well by Name**

# wells = ['HT1', 'HT2', 'HT3']

# Task: Remove 'HT2'

**Access the Last Well**

# wells = ['HT1', 'HT2', 'HT3']

# Task: Print the last well using negative indexing

**Reverse the List**

# wells = ['HT1', 'HT2', 'HT3']

# Task: Reverse the list order

**Calculate Average Porosity**

# porosities = [0.12, 0.15, 0.18, 0.11]

# Task: Calculate and print the average

**Find Max Net Pay**

# net\_pays = [12.5, 8.4, 15.6, 10.2]

# Task: Print the maximum net pay using max()

**Loop Through Zones**

# zones = ['Lower Miocene', 'Upper Miocene', 'Pliocene']

# Task: Print each zone with its index

**Combine 2 Lists of Logs**

# logs1 = ['GR', 'RT']

# logs2 = ['SP', 'RHOB']

# Task: Combine into one list using +

**Count How Many Times 'GR' Appears**

# logs = ['GR', 'RT', 'SP', 'GR', 'NPHI']

# Task: Count the number of 'GR'

**Filter Logs Starting with 'R'**

# logs = ['GR', 'RT', 'SP', 'RHOB', 'NPHI']

# Task: Create a new list with logs starting with 'R'

**Convert All Well Names to Uppercase**

# wells = ['ht1', 'rc1', 'ht2']

# Task: Use list comprehension to convert all to uppercase

**Generate Depth Values**

# Task: Create a list of depths from 1000 to 2000 m with a step of 100

# Output: [1000, 1100, 1200, ..., 2000]

**DICTIONARY**

**Create a Well Info Dictionary**

# Task: Create a dictionary with the following keys and values:

# 'name': 'HT1'

# 'depth': 3550

# 'status': 'producing'

**Access a Value by Key**

**well = {'name': 'HT1', 'depth': 3550, 'status': 'producing'}**

# Task: Print the status of the well

# Task: Update 'depth' to 3600 in the above dictionary

# Task: Add new key-value pair to the dictionary

# Task: Remove the 'status' key from the dictionary

**Loop Through a Dictionary**

# production = {'HT1': 12500, 'HT2': 13200, 'RC1': 11900}

# Task: Print each well name and its production value

**Find Well with Highest Production**

# Task: Find the well name with the maximum production

# Hint: use max() with key=...

**Create a Dictionary from Two Lists**

# wells = ['HT1', 'HT2', 'RC1']

# statuses = ['producing', 'shut-in', 'producing']

# Task: Create a dictionary where each well maps to its status

**Check If Key Exists**

well = {'name': 'HT1', 'depth': 3550}

# Task: Check if 'status' exists in the dictionary

**Nested Dictionary for Well Logs**

logs = {

'HT1': {'GR': 85, 'RT': 12.4},

'HT2': {'GR': 79, 'RT': 14.2}

}

# Task: Print RT value of HT2

**Count Number of Wells by Status**

statuses = {

'HT1': 'producing',

'HT2': 'shut-in',

'RC1': 'producing',

'RC2': 'shut-in'

}

# Task: Count how many wells are 'producing' and 'shut-in'

**Merge Two Dictionaries**

depth = {'HT1': 3550, 'HT2': 3600}

status = {'HT1': 'producing', 'HT2': 'shut-in'}

# Task: Merge them into one dictionary per well:

# Output: {'HT1': {'depth': 3550, 'status': 'producing'}, ...}