



ShadowFox

LEARN • CREATE • LEAD

Python Development Internship Task List





Internship Pre-requisites before starting your tasks:

- 1. LinkedIn Profile Update:** Ensure that your LinkedIn profile is updated to reflect your technical skills, and update your experience section to include "**ShadowFox Python Development Intern.**"
- 2. LinkedIn Post:** It's not mandatory to post the offer letter on LinkedIn, but if you wish to receive **extra swags** at the end of the internship, you can post your offer letter and **tag us** to receive assured swags.
- 3. GitHub Repository:** You are required to create a separate GitHub repository named "**ShadowFox**" for all tasks. You can use any IDE to write your code and upload it to the respective repository.
- 4. Completion of Tasks:** Complete the required tasks as specified in this Task List.
- 5. Proof of Work:** At ShadowFox, we value **Proof of Work (POW)**. You are required to post a video explanation of your respective tasks in LinkedIn. Screenshots must be submitted during your task submission.

After completing all the above steps, proceed with your task completion. Kindly note that all the details and screenshots you submit will be thoroughly verified.



Task Level (Beginner): Do 5 out of 9

1. Variables

1. Create a variable named pi and store the value 22/7 in it. Now check the data type of this variable.
2. Create a variable called for and assign it a value 4. See what happens and find out the reason behind the behavior that you see.
3. Store the principal amount, rate of interest, and time in different variables and then calculate the Simple Interest for 3 years. Formula: Simple Interest = $P \times R \times T / 100$

2. Numbers

1. Write a function that takes two arguments, 145 and 'o', and uses the `format` function to return a formatted string. Print the result. Try to identify the representation used.
2. In a village, there is a circular pond with a radius of 84 meters. Calculate the area of the pond using the formula: Circle Area = πr^2 . (Use the value 3.14 for π) Bonus Question: If there is exactly 1.4 liters of water in a square meter, what is the total amount of water in the pond? Print the answer without any decimal point in it. Hint: Circle Area = πr^2 Water in the pond = Pond Area Water per Square Meter
3. If you cross a 490meterlong street in 7 minutes, calculate your speed in meters per second. Print the answer without any decimal point in it. Hint: Speed = Distance / Time



3. List

1. You have a list of superheroes representing the Justice League. `justice_league = ["Superman", "Batman", "Wonder Woman", "Flash", "Aquaman", "Green Lantern"]`

Perform the following tasks:

1. Calculate the number of members in the Justice League.
2. Batman recruited Batgirl and Nightwing as new members. Add them to your list.
3. Wonder Woman is now the leader of the Justice League. Move her to the beginning of the list.
4. Aquaman and Flash are having conflicts, and you need to separate them. Choose either "Green Lantern" or "Superman" and move them in between Aquaman and Flash.
5. The Justice League faced a crisis, and Superman decided to assemble a new team. Replace the existing list with the following new members: "Cyborg", "Shazam", "Hawkgirl", "Martian Manhunter", "Green Arrow".
6. Sort the Justice League alphabetically. The hero at the 0th index will become the new leader.

(BONUS: Can you predict who the new leader will be?)

Your task is to write Python code to perform these operations on the "justice_league" list. Display the list at each step to observe the changes.



4. If Condition

1. Write a program to determine the BMI Category based on user input.

Ask the user to:

Enter height in meters

Enter weight in kilograms

Calculate BMI using the formula: $BMI = \text{weight} / (\text{height})^2$

Use the following categories:

If BMI is 30 or greater, print "Obesity"

If BMI is between 25 and 29, print "Overweight"

If BMI is between 18.5 and 25, print "Normal"

If BMI is less than 18.5, print "Underweight"

Example:

Enter height in meters: 1.75

Enter weight in kilograms: 70

Output: "Normal"

2. Write a program to determine which country a city belongs to. Given list of cities per country:

Australia = ["Sydney", "Melbourne", "Brisbane", "Perth"]

UAE = ["Dubai", "Abu Dhabi", "Sharjah", "Ajman"]

India = ["Mumbai", "Bangalore", "Chennai", "Delhi"]

Ask the user to enter a city name and print the corresponding country.

Example:

Enter a city name: "Abu Dhabi"

Output: "Abu Dhabi is in UAE"



4. If Condition (Continued...)

3. Write a program to check if two cities belong to the same country. Ask the user to enter two cities and print whether they belong to the same country or not.

Example:

Enter the first city: "Mumbai"

Enter the second city: "Chennai"

Output: "Both cities are in India"

Example:

Enter the first city: "Sydney"

Enter the second city: "Dubai"

Output: "They don't belong to the same country"



5. For Loop:

1. Using a for loop, simulate rolling a sixsided die multiple times (at least 20 times).

Count and print the following statistics:

How many times you rolled a 6

How many times you rolled a 1

How many times you rolled two 6s in a row

2. Imagine you are doing a workout routine, and you have to complete 100 jumping jacks.

Write a program that:

Asks you to perform 10 jumping jacks at a time.

After each set, it asks, "Are you tired?"

If you reply "yes" or "y," it should ask if you want to skip the remaining sets.

If you reply "yes" or "y," it should break and print, "You completed a total of jumping jacks."

For example, if you did only 30 jumping jacks and answered "yes," the program will break and print, "You completed a total of 30 jumping jacks."

If you reply "no" or "n," it should continue and display how many jumping jacks are remaining. After that, ask you again, "Are you tired?"

For example, if you answered "no," it should display that 70 jumping jacks are remaining and ask you again, "Are you tired?"

If you reply "no" or "n," it should continue and display how many jumping jacks are remaining. After that, ask you again, "Are you tired?"

For example, if you answered "no," it should display that 70 jumping jacks are remaining and ask you again, "Are you tired?"

If you complete all 100 jumping jacks, it should print, "Congratulations! You completed the workout," and stop the program



6. Dictionary

1. Create a list of your friends' names. The list should have at least 5 names. Create a list of tuples. Each tuple should contain a friend's name and the length of the name.

For example, if someone's name is Aditya, the tuple would be: ('Aditya', 6)

2. You and your partner are planning a trip, and you want to track expenses. Create two dictionaries, one for your expenses and one for your partner's expenses. Each dictionary should contain at least 5 expense categories and their corresponding amounts.

For example:

Your expenses

your_expenses = {

 "Hotel": 1200,

 "Food": 800,

 "Transportation": 500,

 "Attractions": 300,

 "Miscellaneous": 200

}

Your partner's expenses

partner_expenses = {

 "Hotel": 1000,

 "Food": 900,

 "Transportation": 600,

 "Attractions": 400,

 "Miscellaneous": 150

}

Calculate the total expenses for each of you and print the results.

Determine who spent more money overall and print the result.

Find out the expense category where there is a significant difference in spending between you and your partner.

Print the category and the difference.



7.File handling:

1. student_marks.csv contains the marks and other details for some students.
Write a python program to:

1. Open the file in read mode
2. Create a dictionary from the given data
3. Add a new field to the dictionary total_marks and store the total marks of the students.
4. Add a new field to the dictionary Average and store the average marks of the students.
5. Create a new file and write this information to the new file (<https://www.kaggle.com/arunkumar413/studentmarks>)

2. Create a new CSV file with the newly created total marks and average marks.

8. Classes and Objects:

1. Avengers is a Marvel's American Superheroes team, Now you want to showcase your programming skills by representing the Avengers team using classes. Create a class called Avenger and create these six superheroes using this class.

2. super_heroes = ["Captain America", "Iron Man", "Black Widow", "Hulk", "Thor", "Hawkeye"]

3. Your Avenger class should have these properties:

1. Name
2. Age
3. Gender
4. Super Power
5. Weapon

4. Captain America has Super strength, Iron Man has Technology, Black Widow is superhuman, Hulk has Unlimited Strength, Thor has super Energy and Hawkeye has fighting skills as superpowers.

5. Weapons: Shield, Armor, Batons, No Weapon for hulk, Mjölnir, Bow, and Arrows

6. Create methods to get the information about each superhero

7. Create a method is_leader() which will tell if the superhero is a leader or not.



9. Inheritance

1. Create inheritance using MobilePhone as base class and Apple & Samsung as child class
 1. The base class should have properties:
 1. ScreenType = Touch Screen
 2. NetworkType = 4G/5G
 3. DualSim = True or False
 4. FrontCamera = (5MP/8MP/12MP/16MP)
 5. rearCamera = (8MP/12MP/16MP/32MP/48MP)
 6. RAM = (2GB/3GB/4GB)
 7. Storage = (16GB/32GB/64GB)
 2. Create basic mobile phone functionalities in the classes like: make_call, receive_call, take_a_picture, etc.
 3. Use super() constructor for calling parent class's constructor
 4. Make some objects of Apple class with different properties
 5. Make some objects of Samsung class with different properties



ShadowFox

LEARN • CREATE • LEAD

Task 2(Intermediate Level): Do both the tasks

1. Web Scraper: Extract data from websites using libraries like Beautiful Soup or Scrapy.
2. Hangman: Implement the wordguessing game with visual progress and hints.



Task 3(Advanced level): Do any 1 out of 3

1. Cricket Fielding Analysis Data Collection Objective:

As a budding sports analyst with an interest in cricket, your task is to conduct a detailed fielding performance analysis for three players of your choice from any innings of a T20 match. This analysis will help to gauge individual fielding contributions and their impact on the team's defensive play. A detailed sample data and sample performance matrix is attached along the mail of task list.

Dataset Features:

- Match No.: Identifier for the match.
- Innings: Which innings the data is being recorded for.
- Team: The team in the field.
- Player Name: The fielder involved in the action.
- Ballcount: Sequence number of the ball in the over.
- Position: Fielding position of the player at the time of the ball.
- Short Description: Brief description of the fielding event.
- Pick: Categorize the pick-up as clean pick, good throw, fumble, bad throw, catch, or drop catch.
- Throw: Classify the throw as run out, missed stumping, missed run out, or stumping.
- Runs: Enter the number of runs saved (+) or conceded (-) through the fielding effort.
- Overcount: The over number in which the event occurred.
- Venue: Location of the match.



Task 3(Advanced Level): Continued...

Performance Metrics Formula:

To assess the fielding performance, use the following formula:

$$PS = (CP \times WCP) + (GT \times WGT) + (C \times WC) + (DC \times WDC) + (ST \times WST) + (RO \times WRO) + (MRO \times WMRO) + (DH \times WDH) + RS$$

Where:

PS: Performance Score

CP: Clean Picks

GT: Good Throws

C: Catches

DC: Dropped Catches

ST: Stumpings

RO: Run Outs

MRO: Missed Run Outs

DH: Direct Hits

RS: Runs Saved (positive for runs saved, negative for runs conceded)

Task Instructions:

1. Data Collection: For each ball bowled in the match, record the fielding effort according to the dataset features outlined above. Pay close attention to the effectiveness of fielding actions and their outcomes.

2. Analysis Preparation: Your collected data will be used for advanced fielding analysis, identifying key areas of improvement and fielding strengths within the team.

Deliverable: A well-organized spreadsheet or database containing the complete fielding data for the match.

This task requires meticulous attention to detail and an understanding of cricket fielding dynamics. Your analysis will contribute to strategic fielding placements and improvements in team performance.

View Sample Dataset with Calculation [here](#)

Resources: <https://roadmap.sh/python>



Task 3(Advanced level):

2. Present your findings on the final project, where you are tasked with creating a Jupyter notebook from scratch and conducting a data analysis on a dataset of your choice. This comprehensive process involves selecting a dataset that piques your interest, exploring its contents within a Jupyter notebook, and identifying research questions that the data might help answer.

Guidelines:

1. Begin by finding a dataset that piques your interest. You can choose from a list of places with valuable datasets provided in our reading, or feel free to select data related to your hobbies or work if it is publicly available.
2. Explore the dataset in a Jupyter notebook, gaining a deep understanding of its contents. This exploration phase will help you identify the types of questions that can be addressed using the available data. Remember that data cleaning may be necessary during this step.
3. Based on your exploration, narrow down and define a research question. Consider what specific information or insights you want to derive from the dataset. Your research question should be tailored to the characteristics of the data you have chosen.
4. Utilize various visualization techniques to analyze the dataset and find a solution for your research problem. Visualization is a powerful tool for uncovering patterns, trends, and relationships within the data.
5. Throughout this process, pay close attention to the project description and grading rubric to ensure that your work aligns with the project's requirements and expectations.
6. By the end of this week, you should have not only selected a dataset and formulated a clear research question but also utilized visualizations to derive meaningful insights and solutions to your research problem.



Task 3(Advanced Level): Continued...

Example: In the intricate realm of sales data analytics, the dataset under consideration is a compilation of transactional intricacies encompassing transaction ID, date, gross sales, net sales, profit/loss, and additional factors such as cost of goods sold (COGS), manufacturing costs, and freight costs. The analytical odyssey commences with a meticulous exploration, encompassing exploratory data analysis (EDA) techniques to unravel underlying patterns, detect outliers, and address data anomalies, with a particular focus on intricate financial metrics like COGS.

Fiscal years, representing the financial reporting period distinct from the conventional calendar year, add an additional layer of complexity to the analysis.

Understanding the cyclical nature of fiscal periods is crucial for discerning seasonality effects, identifying peak sales periods, and aligning data insights with broader financial reporting structures.

The formulation of a research question in this sophisticated landscape requires consideration of multifaceted variables. For instance, one might delve into understanding the correlation between manufacturing costs and net sales or investigate the impact of freight costs on overall profitability. This step involves not only identifying questions relevant to the sales domain but also establishing nuanced connections between financial variables to unearth strategic insights.



Task 3(Advanced Level): Continued...

Visualization techniques, implemented through Python libraries like Matplotlib and Seaborn, extend beyond conventional line plots. They may include sophisticated financial visualizations such as cost breakdowns, profit margin trends, and comparative analyses between fiscal years. Time series analyses, integrating fiscal timelines, offer a comprehensive perspective on the evolution of financial metrics over distinct reporting periods.

The ultimate presentation in the Jupyter notebook encapsulates not only the temporal trends in gross sales, net sales, and profit/loss but also incorporates detailed breakdowns of COGS, manufacturing costs, and freight costs. This holistic approach to sales data analysis, integrating both financial and temporal dimensions, demonstrates proficiency not only in conventional data analytics but also in navigating the intricacies of fiscal reporting and complex financial metrics within the sales domain.



Task 3(Advanced Level):

3.Embark on an AI driven journey in the realm of natural language processing (NLP) and machine learning (ML) by deploying a Language Model (LM) of your choice. In this project, you are tasked with delving into the intricacies of LM technology, where the selection of the LM is entirely at your discretion. The comprehensive process involves not only implementing the chosen LM but also conducting an in-depth analysis of its performance and capabilities.

Guidelines:

1. LM Selection: Choose an LM that aligns with your interests, whether it be cutting-edge models like GPT3, BERT, or even a specialized domain-specific LM. The selection should be driven by your curiosity and the specific context in which the LM will be applied.

2. Implementation in a Jupyter Notebook: Create a Jupyter notebook from scratch to implement and showcase the chosen LM. Provide a step-by-step demonstration of the implementation process, highlighting key features, parameters, and any unique aspects of the selected LM.

3. Exploration and Analysis: Conduct a thorough exploration of the LM's capabilities within the Jupyter notebook. Analyze its performance on sample text inputs, showcase its ability to understand context, and evaluate its language generation capabilities. This phase may involve experimenting with different input scenarios and documenting the LM's responses.

4. Research Questions and Objectives: Based on your exploration, define research questions that delve into the strengths and limitations of the chosen LM. Consider aspects such as contextual understanding, creativity in generating text, and adaptability to diverse domains. Tailor your research questions to extract meaningful insights from the LM's behavior.



Task 3(Advanced Level): Continued...

5. Visualization of Results: Utilize visualization techniques to present the results of your LM analysis. This could involve graphical representations of the LM's responses, comparisons with baseline models, or even visualizing the attention mechanisms within the LM architecture. Visualization aids in conveying complex information in an accessible manner.

6. Project Alignment and Evaluation: Align your project with the overarching goals of advancing understanding in the field of NLP and ML. Ensure that your work aligns with best practices, ethical considerations, and the evolving landscape of LM technology. Regularly refer to the project description and grading rubric to meet the specified requirements and expectations.

7. Conclusion and Insights: Summarize your findings, draw insightful conclusions from the LM analysis, and discuss potential applications or areas for improvement. Reflect on the broader implications of your work within the context of the rapidly evolving field of AI and LM technologies. By the conclusion of this project, you should have not only implemented an LM of your choice but also conducted a robust analysis, showcasing your ability to navigate and leverage advanced AI models for language processing tasks.