

PartOne:

A:

- Create Person Object Contains the following Properties: ID:1, Name : "Empty" ,
 - Create Employee object which contains Salary property with set and get using defineProperty and set this value with bounds 20%
- and make it's prototype Person Object
- Create HREmployee object which contains location Property and make it's Prototype Employee object:
- Test prototype chain in console for HREmployee and Employee objects
 - Try to access person ID and Person Name using HREmployee object -
 - Define Name And ID Properties with values For HREmployee Object then test if it accessible with person object
 - Define Age Property with Person Object and test if it accessible with HREmployee Object

--After Try previous ,try create the previous objects again but using defineProperties to create each object property

PartTwo:

Write a Constructor function for Employee (name,age,department,salary) contains setters , getters for salary (salary is private) and toString function to display Employee data on console

- Consider You have Array Of Employees

--use this Array To:

Q1: Create a function that returns another function that Take Emp and Return it's Name e

Q2: Create a counter function that increases every time it's called.

Q3: Create a function that tracks how many times a button is clicked each Time Clicked To change Body Background.

Q4: Create a closure that adds a fixed number to any number.

Q5: Create a closure that keeps track of how many employees have been added.

Q6: Create a closure that Takea Bonus percentage and applies it To Emp Salary.

Q7: Create a closure that remembers a department name and returns a Greeting.

Q8: Use map to get an array of employee names.

Q9: Use filter to get only employees who earn more than 4500.

Q10: Use reduce to calculate the total Salaries.

Q11: Create a pure function that increases an employee salary by 10%.

Q12: Add a new employee to EmpArray immutably(without changing the original use map).

Q13: Write a higher-order function applyBonus(fn).

Q14: Filter employees by department using a reusable curried function.

Q15: Use map to update salaries (+5%) without modifying the original.