

Functions p.I

Basics

Exercises

Level - Easy.....	2
Exercise 1-1.....	2
Exercise 1-2.....	2
Exercise 1-3.....	2
Exercise 1-4.....	2
Exercise 1-5.....	2
Level - Moderate.....	3
Exercise 2-1.....	3
Exercise 2-2.....	3
Exercise 2-3.....	3
Exercise 2-4.....	3
Exercise 2-5.....	3
Exercise 2-6.....	3
Exercise 2-7.....	3

Level - Easy

Exercise 1-1

Write a function called `greet` that takes a name as its parameter and returns (not print !) a greeting message.

Exercise 1-2

Write a function called `remainder` that takes two numbers as parameters and returns (not print !) the remainder of their division.

Exercise 1-3

Write a function called `rectangle_area` that takes the width and height of a rectangle and returns (not print !) its area.

Exercise 1-4

Write a function called `power` that takes two parameters: a number and its exponent. If no exponent is provided, assume it's 2 (default parameter). The function should return (not print !) the number raised to the exponent.

Exercise 1-5

Write a function called `reverse_string` that takes a string as its parameter and prints the string in reverse order using slicing.

Level - Moderate

Exercise 2-1

Write a function called `average` that takes a list of numbers as its parameter and returns the average of those numbers. If the list is empty, return `None`.

Exercise 2-2

Write a function called `factorial` that takes a non-negative integer as its parameter and returns its factorial. Don't use recursion.

Exercise 2-3

Write a function called `max_in_list` that takes a list of numbers and returns the maximum value in the list. Do not use any built-in functions like `max()`. Assume the list is not empty.

Exercise 2-4

Write a function called `is_prime` that takes an integer as its parameter and returns `True` if the number is prime and `False` otherwise.

Exercise 2-5

1. Write a function called `safe_divide` that takes two numbers, `a` and `b`, and returns their division result. If `b` is 0, the function should return `None`.
2. Write a function called `compute_details` that takes two numbers, `a` and `b`. The function should return their sum, difference, product, and quotient (as a tuple). Use the `safe_divide` function to compute their quotient.

Exercise 2-6

A perfect number is a positive integer that is equal to the sum of its proper divisors, excluding itself. For example, 28 is a perfect number because $28 = 1 + 2 + 4 + 7 + 14$. Write a function called `is_perfect` that determines whether a given number is perfect.

Exercise 2-7

Implement a function called `calculate` that can handle different operations based on the number of parameters:

- If it gets 1 parameter, return its square.
- If it gets 2 parameters, return their sum.
- If it gets 3 parameters, return the sum of the two first divided by the last one.