- **Abstractions with Higher Order Functions**
    - Functions allow us to build abstractions by assigning names to common patterns and then to work in terms of the abstraction directly. Eg Square, Sqrt etc that we saw earlier.
    - Functions that manipulate functions - can accept functions as argument or return function as value - are called **Higher Order Funcitons**
    - Higher order functions serve as a powerful abstraction mechnism
- Functions as arguments
    - sum_of_integers, sum_cubes, pi_sum
    - pi_sum = 1/1*3 + 1/5*7 + 1/9*11 + …..
    - tail call optimisation
    - Function expressions (Arrow functions)
    - Ex: The reduce function returns the result of applying a given function to an iterable object, accumulating the result in each iteration. So the idea is (like the name says) to transform the iterable object in a single object (or a "smaller "iterable),
- Functions as return values
    - Closures
- Pure functions
- Higher order functions means:
    - They may be referred to using names
    - They may be passed as arguments to functions
    - They may be returned as results of functions
    - They may be included in the data structures