Ruby - Enumerable - reduce

A common scenario that arises when using a collection of any sort, is to get perform a single type of operation with all the elements and collect the result.

For example, a `sum(array)` function might wish to add all the elements passed as the array and return the result.

A generalized abstraction of same functionality is provided in Ruby in the name of `reduce` (`inject` is an alias). That is, these methods iterate over a collection and accumulate the value of an operation on elements in a base value using an operator and return that base value in the end.

Let's take an example for better understanding.

```
>>> (5..10).inject(1) {|product, n| product * n }
=> 151200
```

In above example, we have the following elements: a base value **1**, an enumerable **(5..10)**, and a block with expressions instructing how to add the calculated value to base value (i.e., multiply the array element to *product*, where product is initialized with base value)

So the execution follows something like this:

```
# loop 1
n = 1
product = 1
return value = 1*1

# loop 2
n = 2
product = 1
return value = 1*2

n = 3
product = 2
return value = 2*3

..
```

As you can notice, the base value is continually updated as the expression loops through the element of container, thus returning the final value of base value as result.

Other examples,

```
>>> (5..10).reduce(1, :*)   # :* is shorthand for multiplication
=> 151200
```

Consider and arithmetico-geometric sequence where the n^{th} term of the sequence is denoted by $t_n = n^2 + 1, n > 0$. In this challenge, your task is to complete the `sum` method which takes an integer `n` and returns the **sum to the n terms of the series**.