

Rewriting Expressions

In this exercise we ask you to rewrite some expressions or to understand the difference forms that a solution can take. Some of the expressions are explained in <https://medium.com/concerning-pharo/elegant-pharo-code-bb590f0856d0>

Exercise: Examine the block expressions

```
[ | sum |  
  sum := 0.  
  #(21 23 53 66 87) do: [:item | sum := sum + item].  
  sum ]
```

- What is the final result of sum?
- Rewrite this piece of code to use explicit array indexing (with the message at:) to access the array elements. Test your version.
- Rewrite this code using inject:into: (Look at the implementation of inject:into: and its users in the system to understand how to use it).

Exercise: Comparing expressions

You can express in different way the same computation. Have a look at each of them and check the messages that you do not know. Look for their implementation.

```
[ | array |  
  array := #(2 4 4 4 5 5 7 9).  
  ((array - array average) squared sum / (array size - 1)) sqrt ]  
[ [ :input | ((input - input average) squared sum / (input size - 1))  
  sqrt ]  
:]
```

```
i value: #(2 4 4 4 5 5 7 9)
[ #(2 4 4 4 5 5 7 9) in: [ :input |
  ((input - input average) squared sum / (input size - 1)) sqrt ]
[ #(2 4 4 4 5 5 7 9) stddev
```