

Réécriture d'expressions

Dans cet exercice, nous vous demandons de réécrire des expressions ou de comprendre les différentes formes que la solution peut prendre. Quelques-uns de ces expressions sont expliquées dans : <https://medium.com/concerning-pharo/elegant-pharo-code-bb590f0856d0>

Exercice : Examinons les expressions de block

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

Exercice :

- Quel est le résultat final ?

Solution. The sum of all the numbers of the array: 250

Exercice :

- Réécrire ce morceau de code pour utiliser une indexation explicite du tableau pour accéder aux éléments du tableau (message at:). Tester votre version.

Solution.

```
| array sum |  
array := #(21 23 53 66 87).  
sum := 0.
```

```
[ 1 to: array size do: [ :i | sum := sum + array at: i ].
  sum
```

Exercice :

- Réécrire ce morceau de code en utilisant le message `inject:into:` (Regarder l'implantation d'`inject:into:` et ces utilisateurs dans le système pour comprendre comment l'utiliser).

Solution.

```
[#(21 23 53 66 87) inject: 0 into: [:item :sum | sum + item]
```

Exercice : Comparer les expressions

Nous pouvons exprimer de différentes manière le même calcul. Regarder chacune des expression et vérifier les messages que vous ne connaissez pas. Regardez leur implantation.

```
[ | 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 ]
  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
```